



a subsidiary of **FORTINET**

*The recognized leader in proven and affordable load
balancing and application delivery solutions*

Equalizer Installation & Administration Guide

Version 8.6
April 2013

Copyright © 2013 Fortinet, Inc. All rights reserved. Fortinet®, FortiGate®, and FortiGuard® are registered trademarks of Fortinet, Inc., and other Fortinet names herein may also be trademarks of Fortinet. All other product or company names may be trademarks of their respective owners. Performance metrics contained herein were attained in internal lab tests under ideal conditions, and performance may vary. Network variables, different network environments and other conditions may affect performance results. Nothing herein represents any binding commitment by Fortinet, and Fortinet disclaims all warranties, whether express or implied, except to the extent Fortinet enters a binding written contract, signed by Fortinet's General Counsel, with a purchaser that expressly warrants that the identified product will perform according to the performance metrics herein. For absolute clarity, any such warranty will be limited to performance in the same ideal conditions as in Fortinet's internal lab tests. Fortinet disclaims in full any guarantees. Fortinet reserves the right to change, modify, transfer, or otherwise revise this publication without notice, and the most current version of the publication shall be applicable.

Coyote Point Systems
A subsidiary of Fortinet, Inc.
56 Main Street
Millerton, NY 12546

This document was revised for Equalizer Software Version 8.6.0i



Contents 3

Preface 15

In This Guide	15
Typographical Conventions	16
Where to Go for More Help	17

Equalizer Overview 19

Introducing Equalizer	20
Intelligent Load Balancing	20
Load Balancing Configuration	21
Real-Time Server Status Information	21
Network Address Translation and Spoofing	22
Maintaining Persistent Sessions and Connections	23
Cookie-Based Persistence (Layer 7)	23
IP-Address Based Persistence (Layer 4).....	24
Is Connection Persistence Always Needed With Session Persistence?	24
Layer 7 Load Balancing and Server Selection	24
Geographic Load Balancing	25
Geographic Load Balancing Routing	26
Distributing the Geographic Load	26
Adding Equalizer to Your Network	29
Equalizer E250GX Network Configuration	29
Using Equalizer E250GX in a Single Network Environment.....	30
Using Equalizer E250GX in a Dual Network Environment	31
Equalizer E350GX, E450GX, E650GX Network Configuration	32
Using Equalizer E350/450/650GX in a Single VLAN Environment.....	33
Using Equalizer E350/450/650GX in a Dual VLAN Environment	34
Using Equalizer E350/450/650GX in a Complex VLAN Environment	35
Link Aggregation	36
Using a Second Equalizer as a Backup Unit	36
Where Do I Go Next?	37

Installing and Configuring Equalizer Hardware 39

Before You Turn Equalizer On for the First Time 40

Stepping Through the Hardware Installation 40

Setting Up a Terminal or Terminal Emulator 41

 Serial Connection 41

Performing Basic Equalizer Configuration 41

 Starting to Configure Equalizer 42

 Configuring External and Internal Interfaces on E250GX 42

 Configuring the Default VLAN on E350/450/650GX 43

 Setting the Time Zone 44

 Setting the Date and Time 44

 Adding Administrative Interface Logins 44

 Changing Equalizer’s Console Password 45

 Upgrading Equalizer Software 45

 Shutting Down Equalizer 46

 Adding Alternate DNS Servers 46

Managing Remote Access to the Equalizer 47

 Managing the Remote Access Account 47

 Using the Remote Access Account 47

Configuring a Second Equalizer As a Backup (Failover) 48

Configuring DNS and Firewalls for Envoy 48

 Configuring the Authoritative Name Server to Query Envoy 48

 Using Geographic Load Balancing with Firewalls 49

Testing Basic Connectivity 49

Using the Administration Interface 51

Logging In and Navigating the Administrative Interface 52

 Logging In 52

 Navigating Through the Interface 53

Managing Access to Equalizer 55

 Viewing and Changing GUI and SSH Access 55

Updating the Administration Interface Certificate 56

Managing Multiple Interface Users 56

 Objects and Permissions 57

 Viewing or Modifying Login Permissions 59

 Adding a Login 60

 Deleting a Login 61

Entering Names for Equalizer Objects 61

Equalizer Network Configuration 63

VLAN Basics	64
Configuring VLANs on Equalizer	65
Initial VLAN Configuration Using the VLAN Wizard	66
Adding a VLAN in Standalone Mode	67
Modifying a VLAN in Standalone Mode	69
Deleting a VLAN in Standalone Mode	71
Adding a VLAN with Failover Enabled	71
Modifying a VLAN with Failover Enabled	72
Modifying a VLAN Name or Heartbeat Setting	72
Modifying VLAN Settings Other Than the Name or Heartbeat Setting	72
Deleting a VLAN with Failover Enabled	74
Managing Interface Ports	75
interface Administration Interface	76
Viewing Link Status	76
Viewing Current Port Settings.....	76
Editing Port Settings	77
Committing and Applying interface Port Configuration Changes.....	78
Switch Interface Usage Scenarios	79
Resetting the Front-Panel Interface Ports	79
Interface Notes for Pre-GX Equalizer Hardware	80
Configuring Static Routes	80
Adding a Static Route	80
Modifying a Static Route	81
Deleting a Static Route	81
Configuring Servers on Your Network	82
Configuring Routing on Servers	82
Server Configuration Constraints	82
Configuring Equalizer Operation 85	
Licensing Equalizer	86
Requesting a License Online	86
Requesting a License Offline	88
Modifying Global Parameters	89
Global Probe Parameters	90
Global Networking Parameters	91
Setting Up a Failover Configuration	95
General Failover Operation	95
Failover Issues with Spanning Tree	95
Failover Determination	95
Failover Between Different Hardware & Software Releases	96

Using Release 8.6 with Release 8.5.1, and GX or 'si' Hardware	96
Using a GX and an 'si' in Failover with Version 8.6	97
Required VLAN Configuration for Failover in Release 8.6	97
Failover and VLAN Configuration in Version 8.6.0f and Later Versions	97
Failover and VLAN Configuration in Versions Prior to Version 8.6.0f	98
Setting Up or Modifying Failover Using the Failover Wizard	99
Enabling Failover Using the Failover Tabs	100
Manually Enabling Failover	101
Modifying the Failover Configuration	105
Disabling the Failover Configuration	105
Re-enabling Failover After Disabling	106
Clearing the Failover Configuration	107
Changing from Multi-VLAN to Single-VLAN Configuration	107
Managing System Time and NTP	109
NTP and Plotting	109
Selecting an NTP Server	110
General System Maintenance	112
Creating a Backup Archive	112
Restoring a Backup Archive	113
Restoring a Backup Archive on an Equalizer in Standalone Mode	113
Restoring a Backup Archive on an Equalizer in a Failover Pair	114
Shutting Down Equalizer	116
Rebooting Equalizer	116
Creating a System Information Archive	116
Upgrading Equalizer Software	117
Administering Virtual Clusters 119	
Working with Virtual Clusters	121
Adding a Layer 7 Virtual Cluster	122
Modifying a Layer 7 Virtual Cluster	122
Layer 7 Required Tab	124
Layer 7 Probes Tab	125
Layer 7 Persistence Tab	126
LB Policy Tab	127
Layer 7 Networking Tab	128
Layer 7 Security > Certificates Tab (HTTPS only)	129
Layer 7 Security > SSL Tab (HTTPS only)	130
Adding a Layer 4 Virtual Cluster	131
Modifying a Layer 4 Virtual Cluster	132
Layer 4 Required Tab	132
Layer 4 Probes Tab	134
Layer 4 Persistence Tab	135
LB Policy Tab	135

Deleting a Virtual Cluster	136
Copying an Existing Virtual Cluster	136
Configuring a Cluster's Load-Balancing Options	137
Equalizer's Load Balancing Policies	137
Equalizer's Load Balancing Response Settings	138
Aggressive Load Balancing	138
Dynamic Weight Oscillations	138
Configuring a Cluster to Use Server Agents	138
Enabling Persistent Server Connections	139
Enabling Sticky Connections	139
Enabling Cookies for Persistent Connections.....	140
Enabling the Once Only and Persist Options	140
Enabling Both the Once Only and Always Options.....	143
Enabling Once Only and No Header Rewrite for HTTPS	143
Enabling Once Only and Compression	144
Using Active Content Verification (ACV)	144
Using ACV	144
Enabling ACV	145
HTTPS Header Insertion	146
Specifying a Custom Header for HTTP/HTTPS Clusters	146
Performance Considerations for HTTPS Clusters	147
HTTPS Performance and Xcel SSL Acceleration.....	147
Providing FTP Services on a Virtual Cluster	148
FTP Cluster Configuration	148
Managing Servers	150
The Server Table	150
Server Software Configuration	151
Adding a Server to a Cluster	152
Modifying a Server	154
Configuring Outbound NAT	156
Enabling Outbound NAT.....	156
Configuring Outbound NAT for a Server.....	156
Using Outbound NAT on a Server IP in Multiple Clusters	157
Adjusting a Server's Initial Weight	157
Setting initial Weights for Homogenous Clusters.....	158
Setting initial Weights for Mixed Clusters	158
Setting Maximum Connections per Server	158
Maximum Connections Limits, Responders, and Hot Spares	159
Interaction of Server Options and Connection Processing	160
Shutting Down a Server Gracefully	160
Removing a Layer 7 Server from Service.....	160
Removing a Layer 4 Server from Service.....	161
Deleting a Server	161

Automatic Cluster Responders	162
Managing Responders	162
Adding a Responder	162
Modifying a Responder	164
Plotting Responder Statistics	164
Using Regular Expressions in Redirect Responders	164
Example 1 -- HTTPS Redirect.....	165
Example 2 -- Multi-Hostname Redirect	166
Example 3 -- Directory Redirect	167
Using Responders in Match Rules	168
Creating a Match Rule for a “Sorry Page”	168
Creating a Match Rule to Redirect All Traffic for a Specific URL	169
More Responder Examples	170
Responders and Hot Spares	170
Configuring Smart Events	171
Smart Events Components	171
Smart Event Trigger Expressions	171
Smart Event Action Functions and Variables	173
Smart Event Operators	175
Smart Event Configuration Parameters	175
Setting Event Timing Parameters	176
Using IPMI to Power Servers On/Off	177
Complex Smart Event Expressions	177
Managing Smart Events	178
Adding a Smart Event	178
Editing a Smart Event	178
Deleting a Smart Event	179
Displaying Smart Event Statistics	179
Using the Smart Event Expression Editor	179
Smart Event Examples	180
Logging a Message When Server Count is Low	180
Unquiescing a Server When Server Count is Low	181
Using IPMI to Conserve Server Resources	183
Configuring Direct Server Return (DSR)	188
Configuring Servers for Direct Server Return	190
Configuring Windows Server 2003 and IIS for DSR	191
Configuring a Linux System running Apache for DSR	192
Configuring a Loopback Interface on Other Systems for DSR.....	192
Loopback Interface Responds to ARP Requests.....	192
Weak and Strong Host Models and DSR.....	193
Testing Virtual Cluster Configuration	193
Testing Your Basic Configuration	194

Monitoring Equalizer Operation 195

Displaying Equalizer System Information	196
Displaying General Cluster Status	197
Displaying the System Event Log	198
Displaying the Virtual Cluster Summary	199
Displaying Global Connection Statistics	201
Displaying Cluster Statistics	203
Displaying Server Statistics	203
Displaying Envoy Statistics	203
Displaying Site Statistics	204
Plotting Global Performance History	205
Plotting Cluster Performance History	205
Plotting Server Performance History	206
Plotting Match Rule Performance History	208
Plotting Responder Performance History	208
Plotting GeoCluster Performance History	209
Plotting Site Performance History	209
Exporting Usage Statistics	210
Configuring Custom Event Handling	213
Forwarding Equalizer Log Information	213
Specifying a Command to Run on an Event	213
Configuring Email Notification	214
Disabling Email Notification	215
Browsing Equalizer Configurations using SNMP	216
Enabling the SNMP Agent	217
Setting Up an SNMP Management Station	218
MIB Description	218
Siblings	219
Configuration and Status	219
Clusters.....	219
Servers	219
Events.....	219

Using Match Rules 221

Why Match Rules?	222
Match Rules Overview	222
Match Rule Processing	223

Match Rule Order	224
Match Rules, the Once Only Flag, and Cookies	225
General Match Expressions and Match Bodies	226
Match Expressions	226
Match Bodies	228
Match Rule Definitions	228
Managing Match Rules	229
The Match Rules Table	230
The Default Match Rule	230
Creating a New Match Rule	231
Modifying a Match Rule	235
Removing a Match Rule	235
Match Functions	235
Match Function Notes	239
Match Rule Behavior When Server Status is not 'Up'	239
Considering Case in String Comparisons	240
Regular Expressions	240
Supported Headers	240
HTTPS Protocol Matching.....	241
Supported Characters in URIs	241
Logical Operators and Constructs in the GUI	241
Using Responders in Match Rules	242
Example Match Rules	242
Parsing the URI Using Match Rules	243
Changing Persistence Settings Using Match Rules	244
Changing the Spoof (SNAT) Setting Using Match Rules	246
Selective SNAT Example	246
Server Selection Based on Content Type Using Match Rules	249
Using the Custom Load Balancing Policy with Match Rules	251
Administering GeoClusters 253	
Overview of Geographic Load Balancing with Envoy	254
Overview of Configuration Process	254
Overview of Envoy Site Selection	254
Licensing and Configuring Envoy	258
Enabling Envoy	258
Configuring the Authoritative Name Server to Query Envoy	258
Using Envoy with Firewalled Networks	260
Using Envoy in a Failover Configuration	260
Using Envoy with NAT Devices	260

Upgrading a Version 7 GeoCluster to Version 8	261
Working with GeoClusters	262
Adding a GeoCluster	262
Viewing and Modifying GeoCluster Parameters	263
Deleting a GeoCluster	265
Displaying Envoy Statistics	266
Plotting GeoCluster History	266
Working with Sites	266
Adding a Site to a GeoCluster	266
Displaying and Modifying Site Information	268
Deleting a Site from a GeoCluster	270
Displaying Site Statistics	270
Plotting Site History	270
Envoy Configuration Worksheet	271
Server Agent Probes 273	
Using Server Agents	273
Enabling Agents	273
Server Agents and Load Balancing Policies	274
Server Agents and Server 'Down' Conditions	274
Sample Server Agent in Perl	274
Timeout Configuration 277	
Connection Timeouts	278
HTTP and HTTPS Connection Timeouts	278
The Once Only Option and HTTP / HTTPS Timeouts	281
Layer 4 Connection Timeouts	281
Application Server Timeouts	282
Connection Timeout Kernel Variables	282
Server Health Check Probes and Timeouts	283
ICMP Probes	283
High Level TCP and ACV Probes	283
TCP Probe Aggregation.....	286
Server Agent Probes	287
Agent Probe Process.....	287
Enabling and Disabling Server Agents	287
Using Reserved IP Addresses 289	
Reserved IP Addresses and Outbound NAT	290
Outbound NAT and Failover	290

Regular Expression Format 291

Regular Expressions in Match Rules and Responders 291

- Terms 291
- Learning About Atoms 292
- Creating a Bracket Expression 292
- Escape Sequences 293
- Matching Expressions 293

Using Certificates in HTTPS Clusters 295

Using Certificates in HTTPS Clusters 296

- About Server Certificates 296
- About Client Certificates 297
- General Certificate Guidelines 297
- Software vs. Hardware Encryption/Decryption 298
- Using Certificates in a Failover Configuration 298

Enabling HTTPS with a Server Certificate 298

Enabling HTTPS with Server and Client Certificates 299

Generating a CSR and Getting It Signed by a CA 300

- Generating a CSR using OpenSSL 300

Generating a Self-Signed Certificate 301

Preparing a Signed CA Certificate for Installation 301

Installing Certificates for an HTTPS Cluster 302

Using IIS with Equalizer 304

- Generating a CSR and Installing a Certificate on Windows Using IIS 304

Converting a Certificate from PEM to PKCS12 Format 305

Private Keys for Cluster Certificates 306

- Private Key Storage 306
 - Clearing Secure Key Storage on Xcel I..... 306
- Private Key Length 307

Configuring Cipher Suites 307

- Default Cipher Suites 307
- Updating the Cipher Suites Field 308
- No Xcel (Software) and Xcel II Cipher Suites 308
- Xcel I Cipher Suites 309
- HTTPS Performance and Xcel SSL Acceleration 309
- Choosing the Cipher for an HTTPS Client Connection 309

Equalizer VLB 311

Equalizer VLB Basic	312
Using VLB Basic	312
Equalizer VLB Advanced	313
Using VLB Advanced	313
Installation and Licensing	314
Enabling Equalizer VLB	314
Enabling VLB Agents on a Cluster	315
Disabling VLB Agents for a Cluster	316
Disabling Equalizer VLB for all Clusters	317
Associating a Server with a Virtual Machine	317
Smart Control Event Examples Using VLB	318
Configuring Multiple Hot Spares (VLB Only)	318
Rebooting an Unresponsive Virtual Machine (VLB only)	320
VLB Logging	323
VLB Plotting	323
Additional Operational Notes	323
Troubleshooting 325	
Equalizer Doesn't Boot for First Time	325
Terminal or terminal emulator not connected to Equalizer	325
Clients Time Out Trying to Contact a Virtual Cluster	326
Equalizer is not gatewaying reply packets from the server.....	326
Test client is on the same network as the servers.....	326
No active servers in the virtual cluster	326
Equalizer is not active	326
Primary and Backup Equalizer Are in a Conflict Over Primary.....	326
Backup Equalizer Continues to Boot	326
Primary and Backup Equalizer Are in a Conflict over Primary.....	326
Can't View Equalizer Administration Pages	326
Equalizer is not active	326
Equalizer Administration Interface Unresponsive	327
Equalizer Administration Page Takes a Long Time to Display	327
DNS server configured on Equalizer is not responding	327
Equalizer Doesn't Respond to Pings to the Admin Address	327
Equalizer is not powered on	327
Equalizer isn't connected to your network	327
Administration address not configured on the external interface.....	327
Browser Hangs When Trying to Connect Via FTP to an FTP Cluster	327

FTP server returns its private IP address in response to a "PASV" command 327

Return Packets from the Server Aren't Routing Correctly 328

 IP spoofing is enabled..... 328

Web Server Cannot Tell Whether Incoming Requests Originate Externally or Internally .. 328

 IP Spoofing is not enabled 328

Why aren't my clusters working if the server status is "up"? 328

Context Help Does Not Appear 328

Restoring IP Access to the Administrative Interface 328

Restoring Login Access to the Administrative Interface 329

Log Contains 'interrupted system call' Messages 329

Log Contains SSL Errors with "wrong version number" 330

GUI Always Reports All Configuration Errors 330

Updating the Configuration File Sequence Number 330

License and Warranty 333

Additional Requirements and Specifications 335

Short-Circuit Protection 335

Power Supply Cord 335

Installation into an Equipment Rack 335

Chassis Warning—Rack-Mounting and Servicing 336

Battery 336

Specifications 336

 Power Requirements 336

 Power Consumption 337

 110V Test Results..... 337

 220V Test Results..... 338

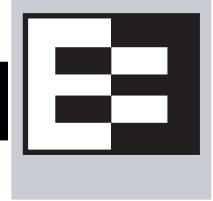
 Operating Environment 338

 Physical Dimensions 338

 Regulatory Certification 338

Glossary 339

Index 349



This version of the *Equalizer Installation and Administration Guide* tells you how to install, configure, and maintain Equalizer™ load balancers running Release 8.6 of the Equalizer software.

In This Guide

This guide contains the following chapters and appendices:

- Chapter 1, “*Equalizer 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 and Configuring Equalizer Hardware*”, provides comprehensive instructions for installing Equalizer hardware and setting up Equalizer to work with your networks and servers.
- Chapter 3, “*Using the Administration Interface*”, discusses how to use Equalizer’s HTML-based administration interface, including adding administrative logins with distinct permissions.
- Chapter 4, “*Equalizer Network Configuration*”, covers VLAN and switch port configuration, as well as how to define static routes on Equalizer.
- Chapter 5, “*Configuring Equalizer Operation*”, tells you how to configure system and global resources through the Equalizer Administration Interface, including setting up a failover configuration.
- Chapter 6, “*Administering Virtual Clusters*”, tells you how to add and remove virtual clusters and servers, changing load balancing options, and shutting down servers.
- Chapter 7, “*Monitoring Equalizer Operation*”, describes how to view information, statistics, and graphical displays about Equalizer’s operation.
- Chapter 8, “*Using Match Rules*”, shows you to create match rules that distribute requests based on a request’s attributes.
- Chapter 9, “*Administering GeoClusters*”, shows you how to use the optional Envoy product to add and remove geographic clusters and sites and change geographic load balancing and targeting options.
- Appendix A, *Server Agent Probes*, describes how to develop custom server agents.
- Appendix B, *Timeout Configuration*, provides detailed descriptions of all the timeout parameters used by Equalizer and the effect that each has on client/server connections and server probes.
- Appendix C, *Using Reserved IP Addresses*, describes how to configure Equalizer to distribute requests to servers assigned IP addresses on reserved, non-routable networks.
- Appendix D, *Regular Expression Format*, discusses Equalizer’s regular expressions, components, formats, and usage.
- Appendix E, *Using Certificates in HTTPS Clusters*, shows you how to obtain and install certificates for HTTPS clusters.

- Appendix F, *Equalizer VLB*, describes the optional Equalizer VLB product, which supports integration of Equalizer with VMware Infrastructure and ESX Server virtual machine configurations.
- Appendix G, *Troubleshooting*, helps you to diagnose problems with Equalizer.
- Appendix H, *License and Warranty*, contains the complete License and Warranty information.
- Appendix I, *Additional Requirements and Specifications*, 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 and file names.

Boldface text highlights command names in instructions and text entered by the user. **Boldface** text highlights graphical administrative interface screen elements: labels, buttons, tabs, icons, etc.

`Courier` text is used to denote computer output: messages, commands, file names, directory names, keywords, and syntax exactly as displayed by the system.

Sequences such as “**Equalizer > Status > Event Log**” are used to indicate the Administrative Interface click-path the user should follow to display the information or form relevant to the task at hand. In this example, the user would click on the Equalizer system name displayed on the left side of the Administrative Interface, then click on the **Status** tab on the right side of the screen, and then click on the **Event Log** sub-tab. Similarly, “**Cluster > Probes**” starts by selecting a cluster name in the left frame tree, and “**Server > Reporting**” starts with a server name.

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 safety information or information critical to Equalizer operation.

Where to Go for More Help

This *Equalizer Installation and Administration Guide* is part of the product documentation delivered with Equalizer's browser-based Administrative Interface. You can display the appropriate manual section for any interface screen by selecting **Help > Context** help from the menu at the top of the interface. The Help menu also contains links to the Release Notes for the currently running software version, and other documentation.

Hardcopy documentation provided with every Equalizer includes the *Getting Started Guide* and the *basic Configuration Guide*. These two documents are designed to help you get Equalizer out of the box and working with your first virtual clusters. The *Basic Configuration Guide* also contains a **Resource CD** with copies of all product documentation, including support documents that help you configure Equalizer for a variety of environments. The latest Resource CD content is available on the web at:

<http://docs.coyotepoint.com>

Customer Support contact information is available from <http://www.coyotepoint.com/support.php>. Register today to get access to the **Coyote Point Support Portal** (<http://support.coyotepoint.com>). Registration provides you with a login so you can access these benefits:

- **Support FAQs:** 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.
- **Links to additional resources,** and more.



Introducing Equalizer	20
Intelligent Load Balancing	20
Load Balancing Configuration	21
Real-Time Server Status Information	21
Network Address Translation and Spoofing	22
Maintaining Persistent Sessions and Connections	23
Cookie-Based Persistence (Layer 7).....	23
IP-Address Based Persistence (Layer 4)	24
Is Connection Persistence Always Needed With Session Persistence?	24
Layer 7 Load Balancing and Server Selection	24
Geographic Load Balancing	25
Geographic Load Balancing Routing.....	26
Distributing the Geographic Load.....	26
Adding Equalizer to Your Network	29
Equalizer E250GX Network Configuration	29
Using Equalizer E250GX in a Single Network Environment.....	30
Using Equalizer E250GX in a Dual Network Environment	31
Equalizer E350GX, E450GX, E650GX Network Configuration	32
Using Equalizer E350/450/650GX in a Single VLAN Environment	33
Using Equalizer E350/450/650GX in a Dual VLAN Environment.....	34
Using Equalizer E350/450/650GX in a Complex VLAN Environment	35
Link Aggregation	36
Using a Second Equalizer as a Backup Unit	36
Where Do I Go Next?	37

Introducing Equalizer

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

- Intelligent load balancing based on multiple, user-configurable criteria
- Non-stop availability with no single point of failure, through the use of redundant servers in a cluster and the optional addition of a failover (or backup) Equalizer
- Layer 7 content-sensitive routing
- Connection persistence using cookies or IP addresses
- Real-time server and cluster performance monitoring
- Server and cluster administration from a single interface
- SSL acceleration (on Equalizer models with Xcel SSL Hardware Acceleration)
- Data compression (on Equalizer models with Express Hardware GZIP Compression)
- Geographic load balancing (with the optional Envoy software add-on)

This chapter is an introduction to Equalizer's basic load balancing and application acceleration capabilities, for those who have some networking experience but may not have previously used an appliance like Equalizer.

Intelligent Load Balancing

Equalizer functions as a *gateway* to one or more sets of *servers* organized into *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 with the appropriate server information and forwards the modified packet to the selected server. Depending on the cluster options chosen, Equalizer may also modify the headers in server responses on the way back to the client.

Equalizer support clusters that route requests based on either *Layer 4* (TCP or UDP) or *Layer 7* (HTTP or HTTPS) protocols. Layer 4 is also referred to as the *Transport Layer*, while Layer 7 is referred to as the *Application Layer*. These terms come from the OSI and TCP/IP Reference Models, abstract models for network protocol design.

In general, Layer 4 clusters are intended for configurations where routing by the destination IP address of the request is sufficient and no examination of the request headers is required. Layer 7 clusters are intended for configurations where routing decisions need to be made based on the content of the request headers. Equalizer evaluates and can modify the content of request headers as it routes packets to servers; in some cases, it can also modify headers in server responses on their way back to the client.

The table below summarizes the basic capabilities of the cluster types supported by Equalizer.

Feature	Cluster Type			
	L4 UDP	L4 TCP	L7 HTTP	L7 HTTPS
Load balancing policies	round robin, static weight, adaptive, fastest response, least connections, server agent, custom			
Server failure detection (probes)	ICMP, TCP, Server Agent	ICMP, TCP, ACV, Server Agent		
Persistence	Based on IP		Using Cookies	
Server selection by request content (i.e., Match Rules)	No; load is balanced according to current load balancing policy.		Yes; load is balanced according to decisions made by examining request content.	
Load balanced protocols	Ideal for stateless UDP-based protocols, such as DNS and RADIUS; WAP gateways; NFS server clusters that provide a single-system image.	Ideal for stateful TCP-based protocols, such as HTTP, HTTPS, SMTP, FTP, LDAP/LDAPS ^a and others.	HTTP	HTTPS
NAT and spoofing	Yes			

a. Note that some LDAP/LDAPS implementations are UDP-based.

Regardless of cluster type, Equalizer uses intelligent *load balancing algorithms* to determine the best server to receive a request. These algorithms take into account the configuration options set for the cluster and servers, real-time server status information, and information from the request itself. For Layer 7 clusters, user-defined match rules can also be used to determine the route a packet should take.

Load Balancing Configuration

When you configure a 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**, **server agent**, or **custom**.

When you configure the servers in a virtual cluster, you assign an *initial weight* between 0 and 200 for each server. When you select one of the adaptive load-balancing algorithms (i.e., any algorithm other than round robin), Equalizer uses the servers' initial 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, and Equalizer does not route requests to servers in this state.

Real-Time Server Status Information

Equalizer gathers real-time information about a server's status using ICMP Probes, TCP Probes, Active Content Verification (ACV), and Server Agents. ICMP and TCP Probes are the default probing methods.

ICMP Probes uses the Internet Control Message Protocol to send an "Echo request" to the server, and then wait for the server to respond with an ICMP "Echo reply" message (like the Unix **ping** command). ICMP is a Layer 3 protocol. ICMP probes can be disabled via a global flag.

TCP Probes establish (and tear down) a TCP connection between Equalizer and the server, in a typical Layer 4 exchange of TCP SYN, ACK, and FIN packets. If the connection cannot be completed, Equalizer considers the server down and stops routing requests to it. TCP probes cannot be disabled.

Equalizer's *Active Content Verification (ACV)* provides an optional method for checking the validity of a server's response using Layer 7 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 ACV is not supported for Layer 4 UDP clusters.) For more information, see "Using Active Content Verification (ACV)" on page 144.

Server Agent Probes are an optional feature that enable Equalizer to communicate with a user-written program (the *agent*) running on the server. A server agent is written to open a server port and, when Equalizer connects to the port, the server agent responds with an indication of the current server load and performance. This enables Equalizer to adjust the dynamic weights of the server according to detailed performance measurements performed by the agent, based on any metrics available on the server. If the server is overloaded and you have enabled **server agent** load balancing, Equalizer reduces the server's dynamic weight so that the server receives fewer requests. The interface between Equalizer and server agents is simple and well-defined. Agents can be written in any language supported on the server (e.g., perl, C, shell script, javascript, etc.). For more information see "Server Agent Probes" on page 273.

For those who have one or more VMware ESX Servers, *Equalizer VLB* can be configured to use VMware's status reporting to determine server status, and can also be configured to automatically manage VMware servers based on status information obtained from VMware. For more information, see Appendix F, "Equalizer VLB".

Network Address Translation and Spoofing

The servers load balanced by Equalizer provide applications or services on specific IP addresses and ports, and are organized into virtual clusters, each with its own IP address. Clients send requests to the cluster IP addresses on Equalizer (instead of sending them to the IP addresses of the servers).

Central to the operation of any load balancer is the *Network Address Translation (NAT)* subsystem. On Equalizer, NAT is used in the following ways:

1. When Equalizer receives a client packet, it *always* translates the destination IP (the cluster IP) to the IP address of one of the servers in the cluster. The server IP used is determined by the cluster's load balancing settings.
2. Depending on the setting of the cluster **spoof** option, Equalizer may also perform *Source NAT*, or *SNAT*.

When the **spoof** option is *enabled*, then SNAT is *disabled*: the NAT subsystem leaves the client IP address as the source IP address in the packet it forwards to the server. For this reason, the servers in a cluster with **spoof** enabled are usually configured to use Equalizer's IP as their default gateway, to ensure that all responses go through Equalizer (otherwise, the server would attempt to respond directly to the client IP).

When the **spoof** option is *disabled*, then SNAT is *enabled*. Equalizer translates the source IP (the client IP) to one of Equalizer's IP addresses before forwarding packets to a server. The servers will send responses back to Equalizer's IP (so it is usually not necessary to set Equalizer as the default gateway on the servers when **spoof** is disabled).

Match rules can be used to selectively apply the **spoof** option to client requests. This is sometimes called *selective SNAT*. See the section "Changing the Spoof (SNAT) Setting Using Match Rules" on page 246.

3. When a server sends a response to a client request through Equalizer, the NAT subsystem *always* translates the source IP in the response packets (that is, the server IP) to the cluster IP to which the client originally sent the

request. This is necessary since the client sent its original request to the cluster IP and will not recognize the server's IP address as a response to its request -- instead, it will drop the packet.

4. NAT can also be enabled for packets that *originate* on the servers behind Equalizer and are destined for subnets other than the subnet on which the servers reside -- on Equalizer, this is called *outbound NAT*. This is usually required in dual network mode when reserved IP addresses (e.g., 10.x.x.x, 192.168.x.x) are being used on the internal interface, so that the recipients do not see reserved IP addresses in packets originating from the servers. When the global **outbound NAT** option is enabled, Equalizer translates the source IP in packets from the servers that are not part of a client connection to the Equalizer's Default VLAN IP address (the external interface IP address on the E250GX and legacy 'si' systems), or to the address specified in the server's **Outbound NAT** tab. Enabling **outbound NAT**, as a result, has a performance cost since Equalizer is examining every outbound packet.

Note – When Equalizer is in single network mode, outbound NAT should be *disabled*. Since Equalizer resides on a single subnet, outbound NAT is not needed, and may cause unexpected behavior. See “Adding Equalizer to Your Network” on page 29 for a description of Equalizer's network modes.

Note that when Equalizer receives a packet that is not destined for a virtual cluster IP address, a failover IP address, a client IP address on an open connection, or one of its own IP addresses, Equalizer passes the packet through to the destination network unaltered.

For more information:

- about setting NAT and spoofing options, see “Working with Virtual Clusters” on page 121.
- about using reserved, non-routing IP addresses with Equalizer, see Appendix C, “Using Reserved IP Addresses” on page 289.

Maintaining Persistent Sessions and Connections

The *persistence of session data* is important when a client and server need to refer to data previously generated again and again as they interact over more than one transaction, possibly more than one connection. Whenever a client places an item in a shopping cart, for example, session data (the item in the cart, customer information, etc.) is created that potentially needs to persist across many individual TCP connections before the data is no longer needed and the session is complete.

It's important to note that *session persistence* is managed by the server application, not Equalizer. Equalizer provides *server persistence* so that a *persistent connection* between a particular client and a particular server can be maintained; this supports a client-server session where session data is being maintained on the server for the life of the connection. In other words, whether you need to enable persistence on Equalizer depends on the application you are load balancing.

Equalizers have no knowledge of the fact that the user has placed something in a shopping cart, logged into a web application, requested a file from shared storage, or made a "post" in a front end presentation server that has been written to a database. Basically, a "state" has been created in the load balanced application of which Equalizer is not aware. What Equalizer *does* know is that a specific client has been load balanced to a specific server in one of its virtual clusters. With this knowledge, Equalizer can track that information and send that client back to the same server they were connected the first time.

Equalizer provides server or connection persistence using cookies in Layer 7 HTTP and HTTPS clusters, and using the client IP address in Layer 4 TCP and UDP clusters. The following sections explain connection persistence provided by Equalizer, and its relationship to session persistence.

Cookie-Based Persistence (Layer 7)

Equalizer can use cookie-based persistent connections for Layer 7 HTTP and HTTPS clusters. 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 (Layer 4)

For Layer 4 TCP and UDP clusters, Equalizer supports IP address based persistent connections. With the *sticky connection* feature enabled, Equalizer identifies clients by their IP addresses when they connect to a cluster. Equalizer then 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 it to an available server based on the current load balancing policy.

To correctly handle sticky connections from ISPs that use multiple proxy servers to direct user connections, Equalizer supports *sticky network aggregation*, which uses only the network portion of a client's IP address to maintain a persistent 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 *intercluster 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 current load balancing policy.

Is Connection Persistence Always Needed With Session Persistence?

Session persistence is a function of the application and the state created when a user logs into a web site. If the session persistence is maintained in the front end server, then Equalizer cookie persistence should be enabled. The client must maintain the connection to the same front end server in order for the login to remain valid. For example, Windows Terminal Services maintains a session directory "database" when a user logs into a session. If that state or database is in the front end server, or even in a back end server that only associates the client connection to that front end server, then the client must "persist" to the front end server to which it is originally connected.

In other configurations, the session "state" is kept in shared storage in a backend server or database that is accessible to all the front end servers. If this is the case, then connection persistence may not be needed; if the user is balanced among servers, then the session can still be maintained across the front end server group via access to the shared storage.

It's therefore important to understand how the load balanced application provides session persistence when managing persistent connections on Equalizer.

Layer 7 Load Balancing and Server Selection

Equalizer's support for Layer 7 content-sensitive load balancing 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*. A match rule might, for example, route requests based on whether the request is for a text file or a graphics file:

- load balance all requests for text files (html, etc.) across servers A and B
- load balance all requests for graphics files across servers C, D, and E
- load balance all other requests across all of the servers

Match Rules are constructed using match functions that make decisions based on the following:

- HTTP protocol version; for HTTPS connections, the SSL protocol level the client uses to connect.
- Client IP address
- Request method (GET, POST, etc.)
- All elements of the request URI (host name, path, filename, query, etc.)
- Pattern matches against request headers

Match functions can be combined using logical constructs (AND, OR, NOT, etc.) to create extremely flexible cluster configurations. Please see “Using Match Rules” on page 221 for an overview of Match Rules, a complete list of match functions, and usage examples.

Geographic Load Balancing

The optional Envoy add-on supports , 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 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.

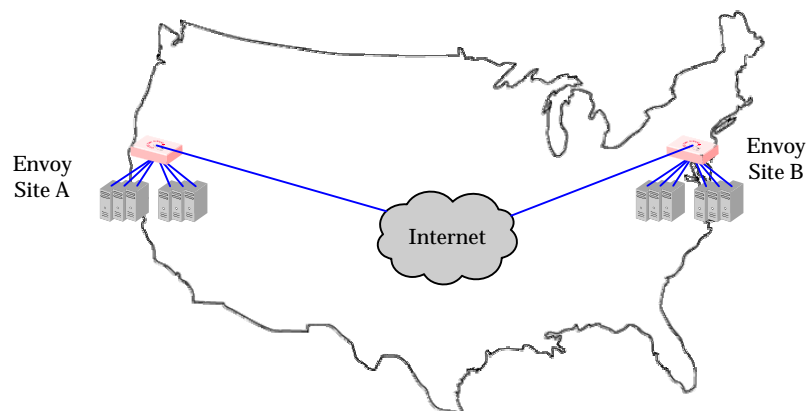


Figure 1 Geographic cluster with two sites

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) protocol¹ 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 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.

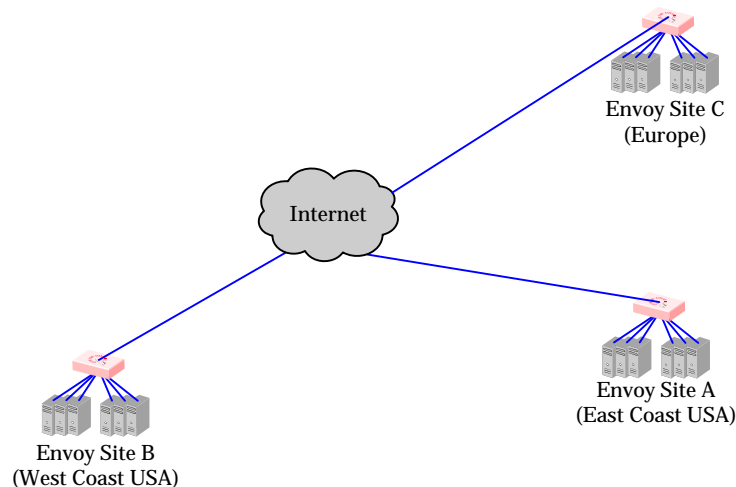


Figure 2 Three-site geographic cluster configuration

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

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

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

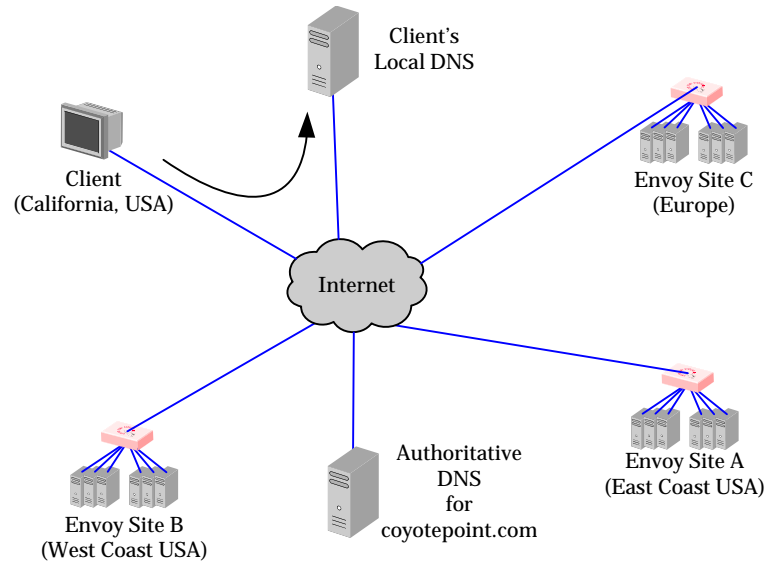


Figure 3 Client queries its local DNS for coyotepoint.com

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

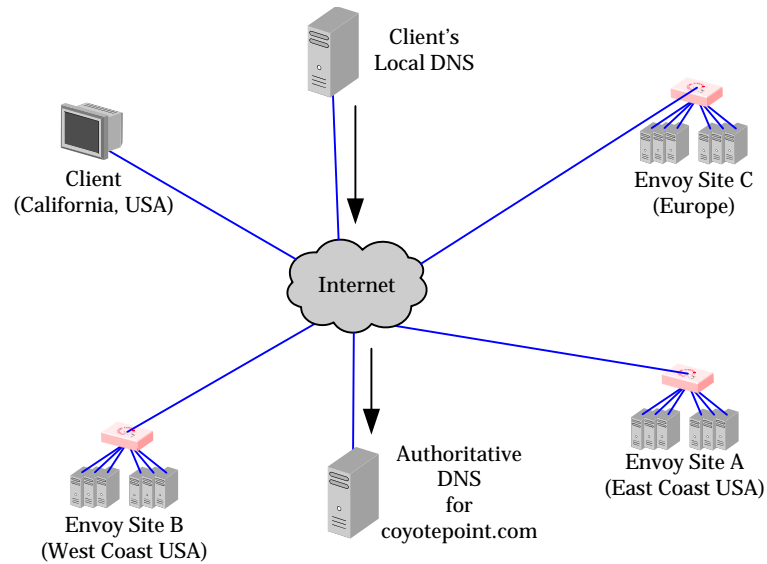


Figure 4 Client's local DNS queries the authoritative name server for coyotepoint.com

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

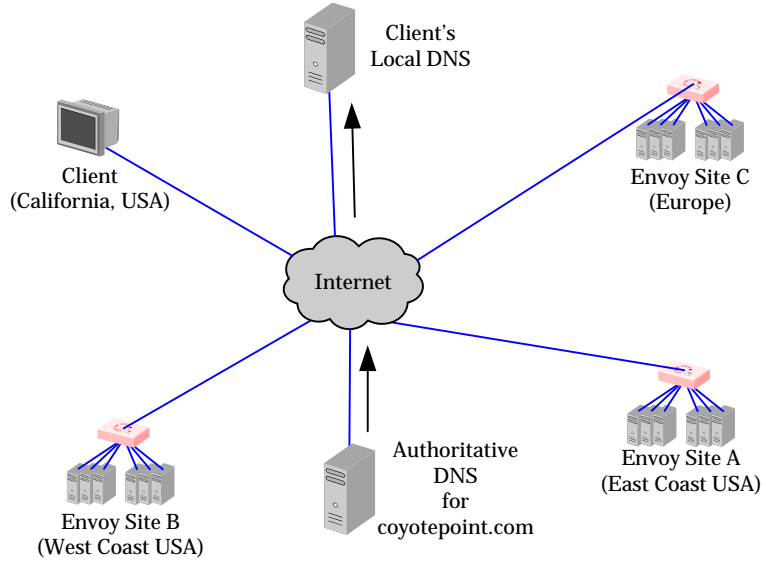


Figure 5 The authoritative name server for coyotepoint.com returns a list of delegates

4. The client's DNS server sends a request for the IP address of `coyotepoint.com` to each Envoy site in the list until one of them responds.
5. The Envoy site contacted returns the IP Address of the virtual cluster best able to handle the client's request. (For an overview of how Envoy chooses the virtual cluster IP to return to the client's DNS, see "Administering GeoClusters" on page 253.)
6. Finally, the client's local DNS server returns the virtual cluster IP to the client, which then sends the request to the virtual cluster.

Adding Equalizer to Your Network

Equalizer is a versatile traffic management and application acceleration solution that is easily configured for your network. Equalizer models E350GX and above have 12 or more front panel network switch ports, are Virtual Local Network (VLAN) capable, and can be configured for tagged and untagged VLANs. Equalizer model E250GX has two front panel ports configured into two port based VLANs.

Note – VLAN management capabilities were introduced in Version 8.6 of the Equalizer O/S software on Equalizer GX hardware. If you are running Version 8.6 or later on pre-GX Equalizer hardware (such as the 'si-R' hardware), the front-panel switches are managed as they were in Version 8.5 of the Equalizer software. Please refer to the Version 8.5 *Installation and Administration Guide* at docs.coyotepoint.com.

Equalizer E250GX Network Configuration

Equalizer model E250GX has two front-panel network interface ports configured into two untagged (or port-based) Virtual Local Networks (VLANs): the External Network VLAN and the Internal Network VLAN. Initial network configuration of Equalizer is performed over the serial port, where you assign an IP address to one or both of the network interface ports. The figure below shows the port configuration of an E250GX model Equalizer.

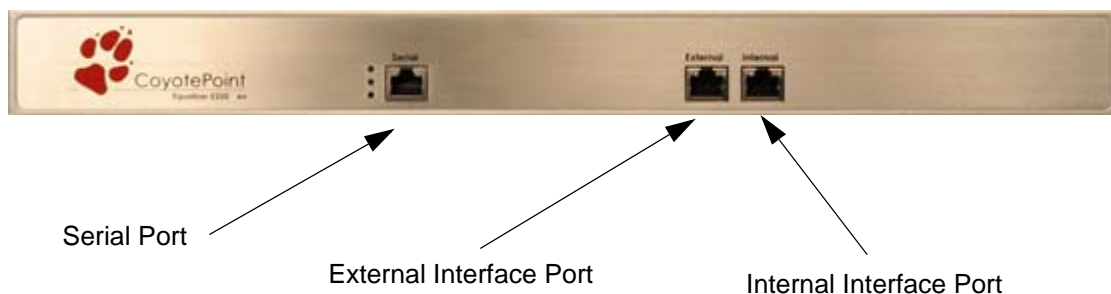


Figure 6 Equalizer E250GX default port configuration

The E250GX can be deployed in either a *single network* or a *dual network* configuration:

- In a *single network* configuration, all cluster IPs and server IPs are on the same subnet, and are connected to Equalizer using the Internal Interface Port; the External Interface Port is unused.
- In a *dual network* configuration, all cluster IPs are on one subnet connected to Equalizer using the External Interface Port, while servers are on another subnet connected to Equalizer's Internal Interface Port.

Using Equalizer E250GX in a Single Network Environment

In single network mode, the client systems, servers, Intranet and/or Internet must all connect to Equalizer through the Internal Interface Port. Figure 7 shows an example.

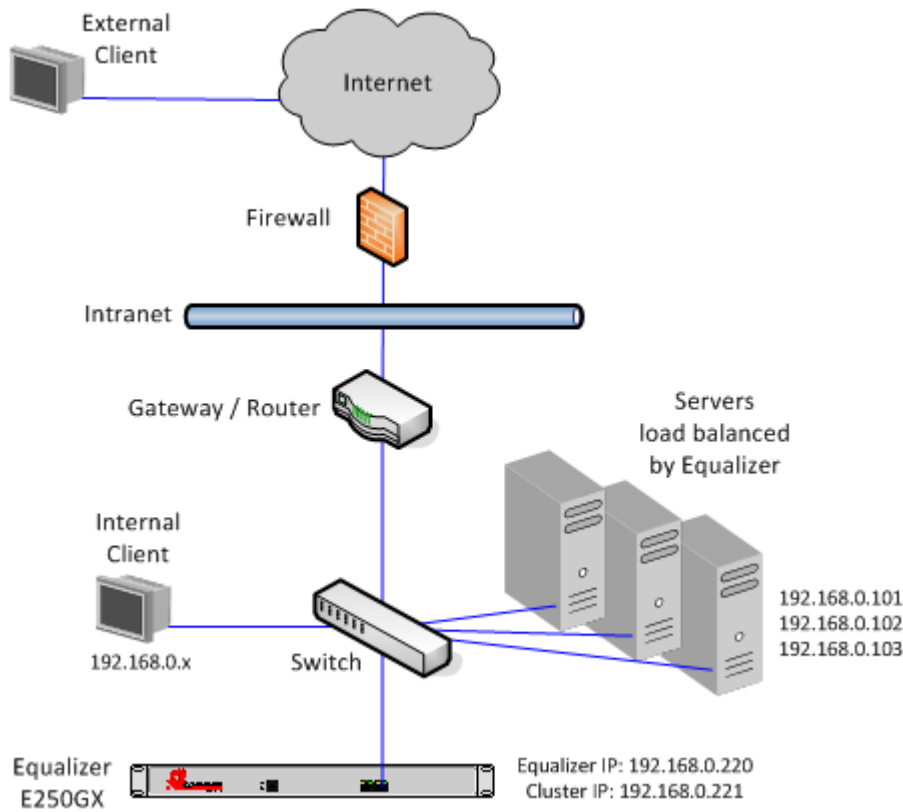


Figure 7 Example single network configuration for Equalizer E250GX

Single network mode is often the simplest way to fit Equalizer into an existing network with minimal changes to the current network infrastructure. Certain protocols and applications that use dynamic port mapping or multiple TCP/UDP ports may also work best in a single network environment.

As you can see in the example above, Equalizer's internal IP, cluster IP, and all server IPs are located on the 192.168.0.x network, and communicate through the same switch. The switch, in turn, is connected to a router which is this subnet's gateway to other subnets on the Intranet and Internet networks. The gateway or router that connects Equalizer's subnet to the Intranet and Internet is assumed to perform all necessary NAT for external clients, so they can access Equalizer's cluster IPs.

Internal clients can access the cluster IPs directly, and so may require selective SNAT on Equalizer or special routing on the servers to ensure that all server responses go through Equalizer.

Using Equalizer E250GX in a Dual Network Environment

In dual network mode, the client systems, Intranet, and Internet connect to Equalizer through the External Interface Port, while servers are connected to Equalizer through the Internal Interface Port. Figure 8 shows an example.

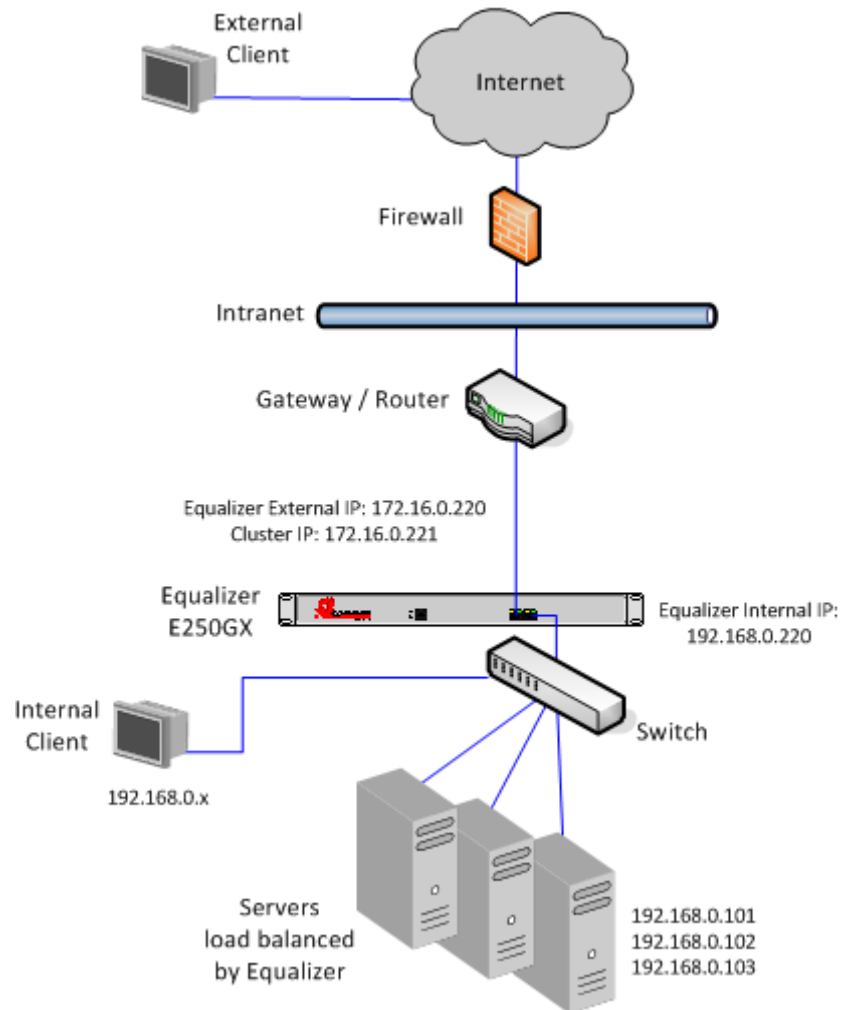


Figure 8 Example single network configuration for Equalizer E250GX

As you can see in the example above, Equalizer has a management IP and cluster IP on the 172.16.0.x network connected to the External Port, and all servers are located on the 192.168.0.x network on the Internal Port.

The External Port is connected to a router which is this subnet's gateway to other subnets on the Intranet and Internet networks. The router is assumed to perform all necessary NAT for external clients, so that clients from outside the 172.16.0.x network can access Equalizer cluster IPs, and Equalizer uses the router as its default gateway.

Internal clients will require special routing to access the cluster IPs, and selective SNAT on Equalizer or special routing on the servers to ensure that server responses to internal clients go through Equalizer.

Equalizer E350GX, E450GX, E650GX Network Configuration

Equalizer models E350GX and above are Virtual Local Network (VLAN) capable devices. Initial network configuration of Equalizer is performed over the serial port, where you assign an IP address to the default management VLAN. Initially, ports 1 and 2 on the front panel are configured for the Default VLAN and all other ports are unassigned. The figure below shows the initial port configuration of an E350GX or E450GX model Equalizer, both of which have 12 front panel ports; the E650GX with 22 ports is configured similarly.

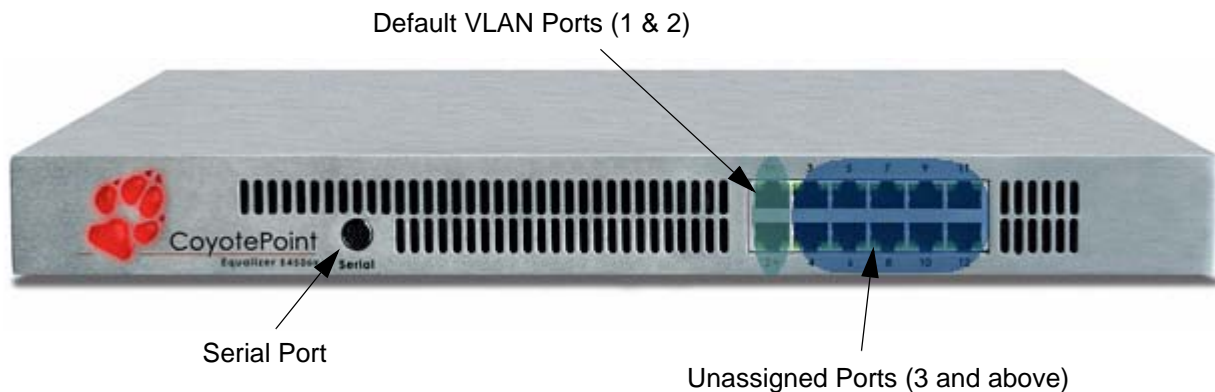


Figure 9 Equalizer E450GX default port configuration

The initial network configuration performed over the Serial Port assigns an IP address to Ports 1 and 2, and these ports are configured in an untagged **Default VLAN** with **VID (VLAN ID) 1**. Additional configuration is performed by logging into the graphical browser-based Administrative Interface on the Default VLAN IP address.

The VLAN Configuration Wizard leads you through the creation of three basic VLAN configurations:

- A single VLAN, the **Default VLAN**, for all management, cluster, and server IP addresses. This is similar to the *single-network configuration* supported in releases prior to 8.6 and on the E250GX.
- Two VLANs: the **Default VLAN** and the **Internal VLAN**. Each of the VLANs has a management IP, and can host clusters and servers. This is similar to the *dual-network configuration* supported in releases prior to 8.6 and on the E250GX.
- Three or more VLANs. The wizard exits to the **VLAN Configuration** tab, where you can set up as many VLANs as you require.

You can also create, modify, and delete VLANs using the **VLAN Configuration** tab. The following sections describe the three basic VLAN configurations; these basic configurations can be modified as needed to fit a variety of network topologies.

Using Equalizer E350/450/650GX in a Single VLAN Environment

In a “single VLAN” or “single network” environment, the client systems, servers, Intranet and/or Internet all connect to Equalizer through a single VLAN (in many configurations, this equates to a single subnet, but may be a segment of a subnet, depending on the network topology). This basic configuration is shown in the diagram below.

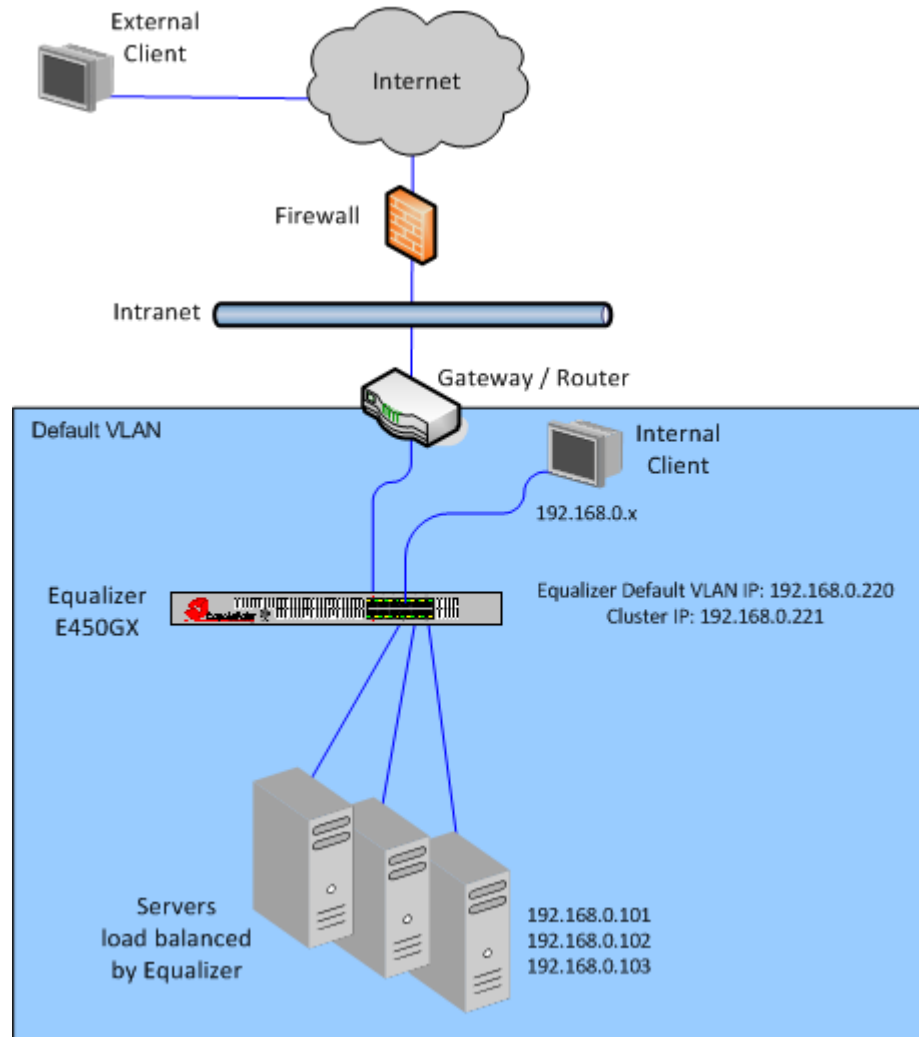


Figure 10 Example of an E450GX single VLAN configuration

Single VLAN mode is often the simplest way to fit Equalizer into an existing network with minimal changes to the current network infrastructure. Certain protocols and applications that use dynamic port mapping or multiple TCP/UDP ports may also work best in a single VLAN environment.

In the example above, all of Equalizer’s ports have been configured for the same VLAN (the Default VLAN), which hosts the 192.168.0.0/24 subnet. Equalizer’s Management IP, cluster IP, and all server IPs are located on the 192.168.0.x network. Equalizer is connected to a router which is this subnet’s gateway to other subnets on the Intranet and Internet networks. The gateway or router that connects Equalizer’s subnet to the Intranet and Internet is assumed to perform all necessary NAT for external clients, so they can access Equalizer’s cluster IPs.

If desired, a separate VLAN can be configured for all cluster and server IPs, and the Default VLAN can be reserved for browser access to the Equalizer Administrative Interface and SSH access to the Equalizer console.

Internal clients can access the cluster IPs directly, and so may require selective SNAT on Equalizer or special routing on the servers to ensure that all server responses go through Equalizer.

Using Equalizer E350/450/650GX in a Dual VLAN Environment

In a dual VLAN environment, the external clients, Intranet, and Internet connect to Equalizer through one VLAN, while servers are connected to Equalizer through a separate VLAN. Figure 11 shows an example.

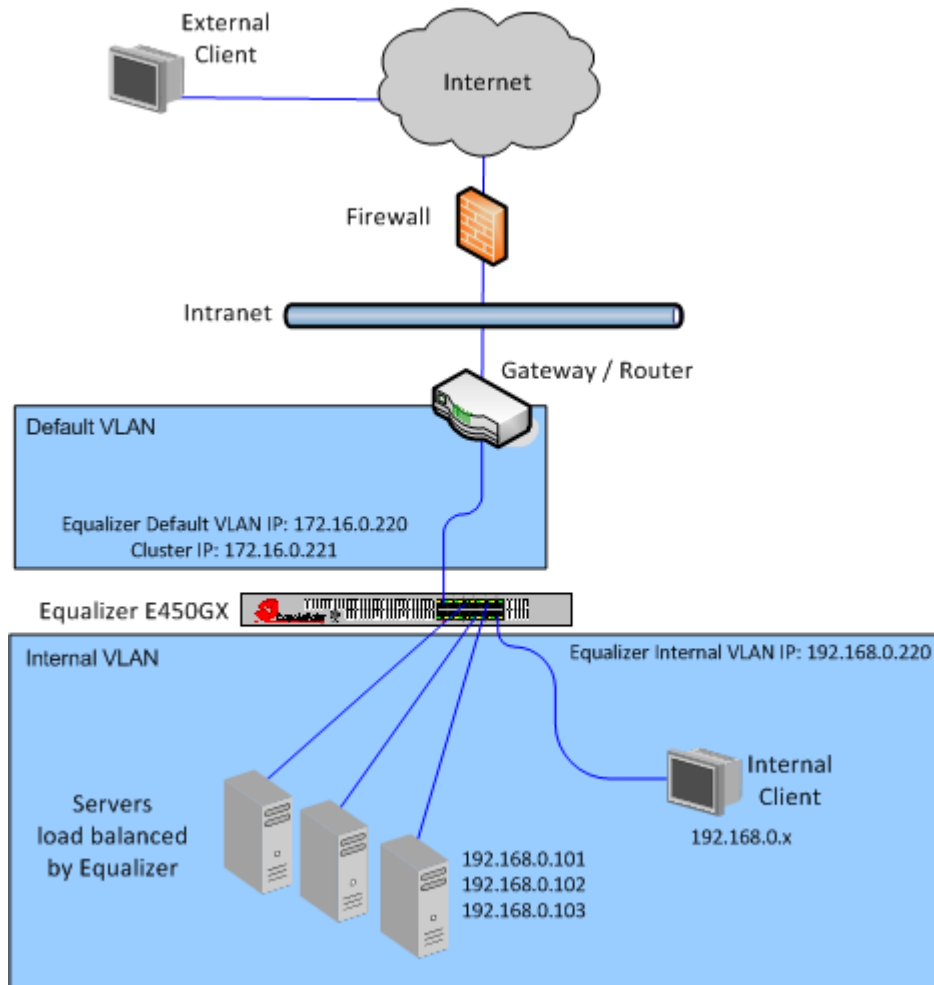


Figure 11 Example of an E450GX dual VLAN configuration

As you can see in the example above, Equalizer has a management IP and cluster IP on the 172.16.0.x network, and is connected to the router for that VLAN on Port 1; ports 3 and above are configured for the Internal VLAN which hosts all servers on the 192.168.0.x subnet.

The router is the Default VLAN's gateway to other subnets on the Intranet and Internet networks. The router is assumed to perform all necessary NAT for external clients, so that clients from outside the 172.16.0.x network can access Equalizer cluster IPs; Equalizer uses the router as its default gateway.

If desired, a separate VLAN can be configured for all cluster IPs, and the Default VLAN can be reserved for browser access to the Equalizer Administrative Interface and SSH access to the Equalizer console.

Internal clients will require special routing to access the cluster IPs, and selective SNAT on Equalizer or special routing on the servers to ensure that server responses to internal clients go through Equalizer.

Using Equalizer E350/450/650GX in a Complex VLAN Environment

The Figure below shows an example of configuring Equalizer into a complex VLAN environment where servers (and clients) are located on several separate VLANs:

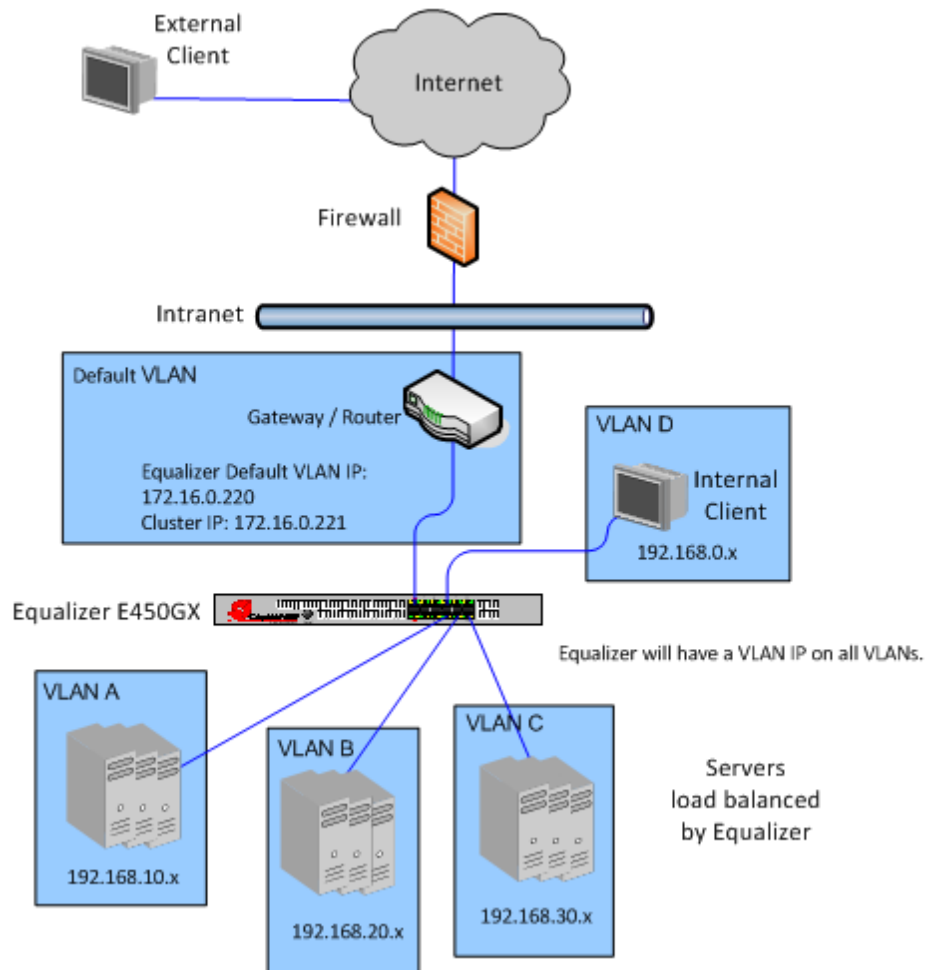


Figure 12 Example of an E450GX complex VLAN configuration

In the example above, Equalizer has a management IP and cluster IP on the 172.16.0.x network, and is connected to the router for that VLAN on Port 1; ports 3 and above are individually configured to carry traffic for VLANs A, B, C, and D -- each of which hosts a different subnet.

The router is the Default VLAN's gateway to other subnets on the Intranet and Internet networks. The router is assumed to perform all necessary NAT for external clients, so that clients from outside the 172.16.0.x network can access Equalizer cluster IPs; Equalizer uses the router as its default gateway.

If desired, a separate VLAN can be configured for all cluster IPs, and the Default VLAN can be reserved for browser access to the Equalizer Administrative Interface and SSH access to the Equalizer console.

Internal clients will require special routing to access the cluster IPs, and selective SNAT on Equalizer or special routing on the servers to ensure that server responses to internal clients go through Equalizer.

For more information on configuring VLANs on Equalizer, please see Chapter 4, "Equalizer Network Configuration".

Link Aggregation

Equalizer E350GX models and above are equipped with two Gigabit network interface cards that are teamed together using *Link Aggregation* to provide up to 2 Gigabits of throughput when redundant links are used. Link aggregation is always enabled and is managed by Equalizer; no administrative settings are necessary.

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 *failover* or *hot-backup* configuration. The two Equalizers are defined as *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 IP 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 failover 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 13 on page 36 shows a sample failover configuration.

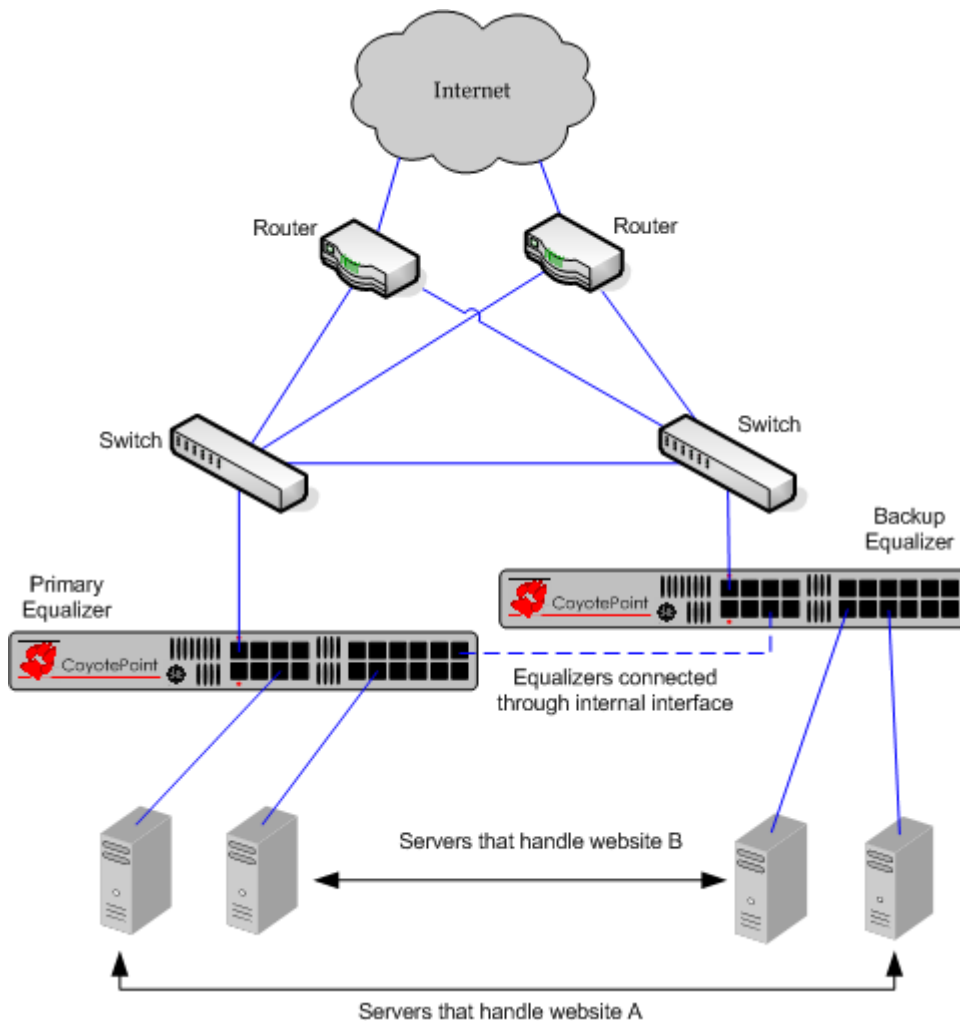


Figure 13 Sample failover configuration

In the sample failover configuration above, there is no single point of failure. If a router goes down, the other router takes over; if a link fails, requests are routed through another link. In this dual network configuration, the Equalizers communicate over both the internal and external subnets. The connection shown between the Equalizers on the internal interface might not be necessary in a single network configuration, or if the Equalizers can communicate over the internal interface through an existing route.

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

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

When the backup Equalizer determines that its failover peer 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 failover peer. Once both Equalizers have synchronized, the newly-started unit assumes the backup role.

For more information on setting up Failover, please see “Setting Up a Failover Configuration” on page 95.

Where Do I Go Next?

Chapter 2 through Chapter 6 tell you how to install Equalizer into your network and configure clusters and servers. The remainder of the manual basically provides additional information to use and administer the features introduced in these chapters:

- Chapter 2, “*Installing and Configuring Equalizer Hardware*”, provides comprehensive instructions for installing Equalizer hardware and setting up Equalizer to work with your networks and servers.
- Chapter 3, “*Using the Administration Interface*”, discusses how to use Equalizer’s HTML-based administration interface, including adding administrative logins with distinct permissions.
- Chapter 5, “*Configuring Equalizer Operation*”, tells you how to configure system and global resources through the Equalizer Administration Interface, including setting up a failover configuration.
- Chapter 4, “*Equalizer Network Configuration*”, covers VLAN and switch port configuration, as well as how to define static routes on Equalizer.
- Chapter 6, “*Administering Virtual Clusters*”, tells you how to add and remove virtual clusters and servers, changing load balancing options, and shutting down servers.

If you are also deploying the optional Envoy Geographic Server Load Balancing product, refer to Chapter 9, “*Administering GeoClusters*”, for Envoy configuration specifics.



This chapter contains all the information you need to get your Equalizer out of the box and onto your network:

Before You Turn Equalizer On for the First Time	40
Stepping Through the Hardware Installation	40
Setting Up a Terminal or Terminal Emulator	41
Serial Connection	41
Performing Basic Equalizer Configuration	41
Starting to Configure Equalizer	42
Configuring External and Internal Interfaces on E250GX	42
Configuring the Default VLAN on E350/450/650GX	43
Setting the Time Zone	44
Setting the Date and Time	44
Adding Administrative Interface Logins	44
Changing Equalizer's Console Password	45
Upgrading Equalizer Software	45
Shutting Down Equalizer	46
Adding Alternate DNS Servers	46
Managing Remote Access to the Equalizer	47
Managing the Remote Access Account	47
Using the Remote Access Account	47
Configuring a Second Equalizer As a Backup (Failover)	48
Configuring DNS and Firewalls for Envoy	48
Configuring the Authoritative Name Server to Query Envoy	48
Using Geographic Load Balancing with Firewalls	49
Testing Basic Connectivity	49

Before You Turn Equalizer On for the First Time

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 I, *Additional Requirements and Specifications*, on page 335 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 H, *License and Warranty*, on page 333.
2. Place the Equalizer in its intended position in an EIA equipment rack or on a flat surface. See Appendix I, *Additional Requirements and Specifications*, on page 335, for a list of environmental limits and power requirements for your Equalizer.
3. Connect a serial terminal or a workstation running terminal emulator software to the serial port on the front panel of the Equalizer (see Figure 9 on page 32). Use the serial cable supplied with Equalizer.
4. 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 one of the RJ-45 ports labelled 1 or 2 on the front panel. Connect Equalizer to the internal network using one or more of the ports numbered 3 and above.
 - b. For a single-network (one subnet) topology, connect Equalizer to the network and the servers using one of the numbered RJ-45 ports numbered 3 and above on the front panel of the Equalizer.
5. 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.
6. Turn on the power using the switch on the rear panel.

Setting Up a Terminal or Terminal Emulator

After the Equalizer hardware, you need to directly connect a terminal to Equalizer 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 9) 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 is freely available at:

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

Performing Basic Equalizer Configuration

The first time you configure Equalizer, you'll need to use the Equalizer Configuration Utility (**eqadmin**) to specify at least the following:

- **Network Access:** On an E250GX, you configure the Internal and External Interface ports using **eqadmin**. On other Equalizer models, ports 1 and 2 on Equalizer's front panel are initially configured on the Default VLAN (VID 1). Once you configure network access via **eqadmin**, further network configuration is performed through the browser-based Administration Interface.
- **Hostname/IP Address:** The DNS hostname or IP address that is assigned to Equalizer.
- **Default Gateway:** The IP address of the router or other network device that Equalizer will use to forward packets to the Internet or Intranet.
- **DNS Server:** The Domain Name Server Equalizer will use.

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. Do *one* of the following:
 - On an E250GX system, continue with “Configuring External and Internal Interfaces on E250GX” on page 42.
 - On an E350GX, E450GX, and E650GX system, continue with “Configuring the Default VLAN on E350/450/650GX” on page 43.

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.

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.

Configuring External and Internal Interfaces on E250GX

On an E250GX, Equalizer has two front panel ports. To configure the Hostname, Network Interfaces, Default Router, and DNS on an E250GX, use the following steps. Even if you are using the E250GX in a single network configuration, you need to enter information for both the external and internal (server) interfaces. See “Equalizer E250GX Network Configuration” on page 29 for an overview of the network configurations supported on the E250GX.

5. In the Equalizer Configuration Menu window, select option 1, **Interfaces**, and press **Enter**. Equalizer displays the **Configure network interfaces** window:
6. Press one or more arrow keys until you highlight **External Ethernet interface**; then press **Enter**. The Equalizer Configuration Utility displays the Network Configuration window.
 - a. 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.
 - b. 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.
 - c. 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.
 - d. 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, you can enter “**NONE**” or “0.0.0.0”; we recommend you use “0.0.0.0”, which will disable DNS lookups on Equalizer.

- Only one DNS server can be specified via eqadmin. If you want to specify additional alternate DNS servers, see the section .
- e. If you will be using a **dual-network configuration**, 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. For **single network configurations** using a switch-based Equalizer, leave the IP address for the external interface blank (or, set to **NONE**) to disable the port.
 - f. When you're finished, highlight **OK**. Then press **Enter**.
7. To specify the internal interface parameters, select **Internal Ethernet interface**. Then press **Enter**.
 - a. 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.
 - b. Highlight **OK**. Then press **Enter**.
 8. Highlight **Back**. Then press **Enter** to return to the main configuration menu.
 9. In the **Equalizer Configuration Menu** window, select option 6, **Commit**; then press **Enter**. The system commits your changes and automatically reboots.

After rebooting, you can continue to configure Equalizer as described in the remainder of this chapter, and test your initial setup as shown in the section “Testing Basic Connectivity” on page 49.

Configuring the Default VLAN on E350/450/650GX

On Equalizer models E350GX and above, the Equalizer's Default VLAN (VLAN 1) Interface is configured first, and additional network configuration is performed by logging into the graphical Administrative Interface. To configure the Default VLAN (including the Equalizer's hostname, default gateway, and DNS), follow these steps. See “Equalizer E350GX, E450GX, E650GX Network Configuration” on page 32 for an overview of the network configurations supported on the E350GX and higher models.

Note – The **Network Configuration** screen is intended to be used only for first-time configuration and when the GUI is inaccessible. Modifying the Default VLAN on an Equalizer that is in production may lead to unpredictable behavior, including dropped connections.

1. In the Equalizer Configuration Menu window, select option 1, **Interfaces**, and press **Enter**. Equalizer displays the **Network Configuration** window:
 - a. 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.
 - b. 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.
 - c. In the **Gateway** field (required), enter the IP address of the router on the Default VLAN. This is the gateway for all the packets Equalizer sends to the outside world on the Default VLAN. For example, if your router is located at IP address 192.22.33.1, enter “192.22.33.1” in the Gateway field.

Note – The **Gateway** that you assign on this screen is the default gateway for all outbound packets whose destination address is not on a configured VLAN. In this release, it is not possible to assign a default gateway IP for each VLAN. Instead, you will need to create static routes either on Equalizer, on the next-hop router for a VLAN, or both, in order to ensure that traffic is routed properly in your configuration.

- d. 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, you can enter “**NONE**” or “**0.0.0.0**”; we recommend you use “**0.0.0.0**”, which will disable DNS lookups on Equalizer.

Only one DNS server can be specified via eqadmin. If you want to specify additional alternate DNS servers, see the section “Adding Alternate DNS Servers” on page 46.

- e. In the **IP address** and **Netmask** fields, respectively, specify the IP address and netmask for the Default VLAN.
 - f. When you’re finished, highlight **OK** and press **Enter**.
2. In the **Equalizer Configuration Menu** window, select option ‘**6**’, **Commit**; then press **Enter**. The system commits your changes and automatically reboots.

After rebooting, you can continue to configure Equalizer as described in the remainder of this chapter, and test your initial setup as shown in the section “Testing Basic Connectivity” on page 49.

Setting the Time Zone

The time zone can be set using the browser-based Administration Interface, which also supports setting up a Network Time protocol (NTP) server, as shown in “Managing System Time and NTP” on page 109. To set the system time zone using **eqadmin**, 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

The current date and time can be set using the browser-based Administration Interface, which also supports setting up a Network Time protocol (NTP) server, as shown in “Managing System Time and NTP” on page 109. To set the system date and time using **eqadmin**, 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**.

Adding Administrative Interface Logins

The browser-based Administrative Interface by default supports two logins: **touch** and **look**. The **touch** login has control access over Equalizer’s configuration, while the **look** login has read access only to the interface. Additional logins can be created with custom permissions on clusters and global configuration. See “Managing Multiple Interface Users” on page 56 for a description of the user management interface.

Option **4 Manage users** allows you to create a full access or read only user login for the Administrative Interface in the event you cannot log in, either because all logins have been accidentally deleted or all administrative passwords lost. To add a user login, do the following:

1. In the **Equalizer Configuration Menu** window, select option 4, **Manage users**, and press **Enter**.
2. Select either **Full access login** or **Read-only login**.
3. Type in a name for the new login. Then, type the 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*).

4. Select **OK** to create the login and return to the main menu.

Changing Equalizer's Console Password

The console password is the password for the **eqadmin** account, which automatically displays the Equalizer Configuration Utility when you log in via **ssh** or the serial port. 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.

Upgrading Equalizer Software

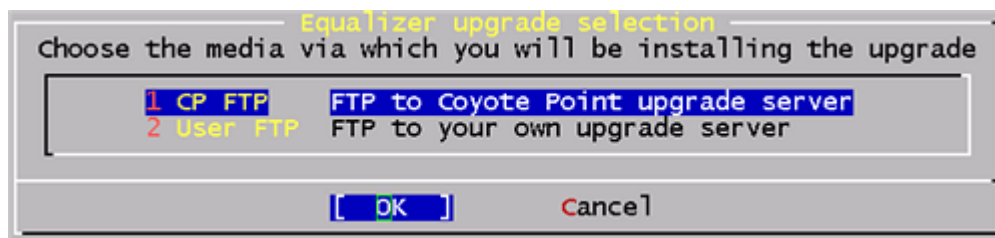
After you have finished connecting Equalizer to your network, you can use the Equalizer Configuration Utility to install the latest Equalizer software upgrade from Coyote Point. (You can also upgrade Equalizer software using the graphical, browser-based Administrative Interface; please see "Upgrading Equalizer Software" on page 117.)

In order to upgrade:

- Equalizer must be licensed; see "Licensing Equalizer" on page 86 for more information.
- Equalizer must be able to access the Internet using FTP, or have access to a local FTP server that already has the upgrade image.

The procedure below contains the basic upgrade instructions for the current Equalizer software release. Please visit the **Documentation Repository** at docs.coyotepoint.com for detailed upgrade instructions (including version compatibility charts for all releases), and other useful documentation.

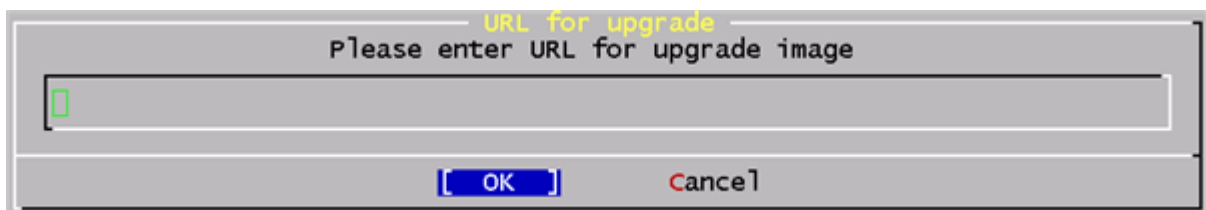
1. In the **Equalizer Configuration Menu** window, select option 8, **Upgrade**, and press **Enter**. You are prompted to choose the location of the upgrade image:



- Use the first option to connect to the Coyote Point FTP server to download the upgrade image.
- Use the second option to specify a local FTP server, to which you have already downloaded the upgrade image from the Coyote Point FTP server.

Use the arrow or number keys to choose the appropriate location and then press **Enter**.

2. The upgrade utility prompts you to enter the upgrade URL:



Enter the URL appropriate for the option you selected in **Step 1**:

- If you chose **Option 1 CP FTP**: Enter the upgrade image URL provided to you by Coyote Point. The latest release of Equalizer software is always located at the following URL:

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

- If you chose **Option 2 User FTP**: Enter the upgrade image URL appropriate for your local FTP server, as provided by your local network administrator.

After entering the URL, select **OK**, and press **Enter**. 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.

Adding Alternate DNS Servers

Only one DNS server can be specified via **eqadmin**, as shown in the section “Configuring the Default VLAN on E350/450/650GX” on page 43. If you want to add additional, alternate DNS servers, follow the procedure below.

Caution – Do not attempt this procedure unless you are comfortable working at the Equalizer command line and editing system files using standard BSD system text editors. Or, contact **Coyote Point Support** for assistance.

1. Log in to Equalizer via **ssh** (as *eqsupport*) or the serial interface (as *root*). If you use **ssh**, enter the following command to switch to the root login:

```
su root
```

2. Enter the following two commands:

```
mount -w /  
cd /var/etc
```

3. Edit the file *resolv.conf* using either of the text editors supplied with Equalizer: **ee** or **vi**. For example:

```
ee resolv.conf
```

Up to three name servers are permitted in the file, as in the following example:

```
domain          mydomain.com  
nameserver      10.0.0.120  
nameserver      10.0.0.122  
nameserver      10.0.0.124
```

For instructions on using the **ee** and **vi** editors, see their manual pages on the FreeBSD Hypertext Manual Pages website: www.freebsd.org/cgi/man.cgi.

4. When you are done updating the file, save your changes and exit the editor.
5. Enter the following two commands:

```
shadow /var/etc/resolv.conf
```

```
mount -r /
```

The new DNS settings take effect for all subsequent DNS queries, and will persist across system reboots.

Managing Remote Access to the Equalizer

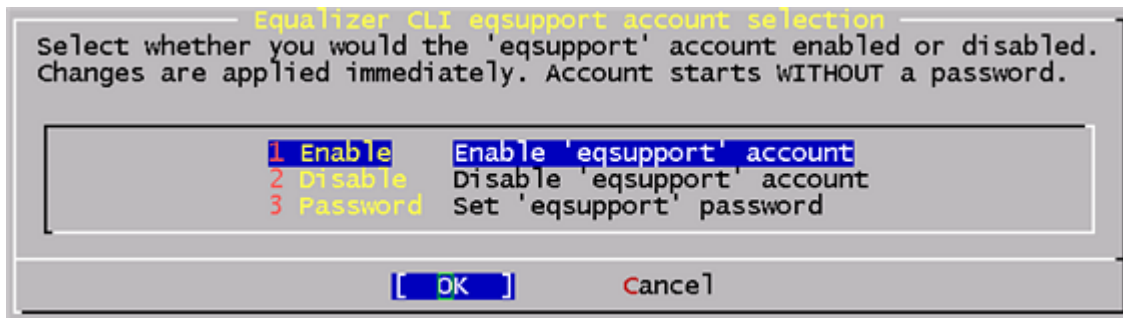
Remote access, when enabled, provides a user account (**eqsupport**) which allows you to log into Equalizer over a Secure Shell (SSH) connection.

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 the **eqsupport** account, do the following:

1. Log into the Equalizer hardware configuration utility using a terminal or terminal emulator (see “Setting Up a Terminal or Terminal Emulator” on page 41 and “Starting to Configure Equalizer” on page 42).
2. In the **Equalizer Configuration Menu**, select option 9, **Manage 'eqsupport'**, and press **Enter**. Equalizer displays the **Equalizer CLI eqsupport account selection** window.



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, using Equalizer’s external or internal interface IP address. The account is not enabled by default, and must first be enabled (see the previous section) in order for this to work. For the best visual output when using **eqadmin** over **ssh**, 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”), and 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’.

Note – When you allow login to Equalizer over SSH, you must ensure that **eqadmin** is not run by more than one login at a time. In other words, do not log into Equalizer over multiple connections (SSH or serial) and run **eqadmin** on more than one connection. This can cause configuration issues if multiple **eqadmin** instances attempt to modify the configuration at the same time.

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 can no longer reach the network, the other Equalizer automatically takes over. This is called a *failover configuration*. When a unit in a failover pair comes back online, it detects that the other Equalizer is active and so assumes the backup role.

If you are going to use two Equalizers in a failover configuration:

- set up *both* Equalizers as described in the section “Performing Basic Equalizer Configuration” on page 41
- configure failover through the Equalizer Administration Interface, as described in the section “Setting Up a Failover Configuration” on page 95

Configuring DNS and Firewalls for Envoy

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 Firewalls

Envoy 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 Basic Connectivity

Once you have installed and configured Equalizer as described in this chapter, do the following to test basic connectivity:

1. Ping Equalizer's Default VLAN IP address from another host on the same subnet. On an E250GX, ping the External and Internal Interface IPs from hosts on the same subnets.
2. If DNS is configured, log into Equalizer and ping a host on the Internet (e.g., `www.coyotepoint.com`) to ensure that DNS and the configured network gateway are functioning properly. If DNS is not configured, use IP addresses instead of fully qualified domain names to test the gateway.
3. Log into the Administrative Interface on the Default VLAN IP address (or one of the Interface IP addresses on an E250GX), as described in "Logging In and Navigating the Administrative Interface" on page 52.



Use Equalizer’s HTML-based Administration Interface to perform the monitoring and administrative tasks described in the subsequent chapters of this guide. This chapter contains the following sections that show you how to log in and configure access to the interface:

Logging In and Navigating the Administrative Interface	52
Logging In	52
Navigating Through the Interface	53
Managing Access to Equalizer	55
Viewing and Changing GUI and SSH Access	55
Updating the Administration Interface Certificate	56
Managing Multiple Interface Users	56
Objects and Permissions	57
Viewing or Modifying Login Permissions	59
Adding a Login	60
Deleting a Login	61
Entering Names for Equalizer Objects	61

Logging In and Navigating the Administrative Interface

The Equalizer Administration Interface can be opened in any Javascript-enabled browser. Two default logins are provided: the **look** login provides read-only access to the interface, and the **touch** login lets you view and edit the configuration. (The section “Managing Multiple Interface Users” on page 56 shows you how to add additional logins as well as define the resource that any login can view or edit.)

Logging In

On an E250GX, log into the management IP for either the Internal Interface or the External Interface (if configured).

On other Equalizer models, log into Equalizer over any VLAN interface that is configured to allow GUI access. If you have a new Equalizer that has just been configured as described in the previous chapter, only the Default VLAN IP address will be available. If you have additional VLANs configured and enabled for GUI access, you can log in using the VLAN IP address on those VLANs as well. The instructions below assume you are logging in on the Default VLAN:

1. Open a Javascript-enabled web browser. We recommend that you use one of these browsers:
 - Internet Explorer Version 7 or later
 - Firefox Version 2 or later
2. From the browser, open the URL that corresponds to Equalizer's Default VLAN IP address, using either the `http` or `https` protocols. For example, if the Default VLAN IP address is `199.146.85.2`, open the Equalizer Administration Interface by typing `http://199.146.85.2` or `https://199.146.85.2` into the browser's address bar. Equalizer displays the login screen:

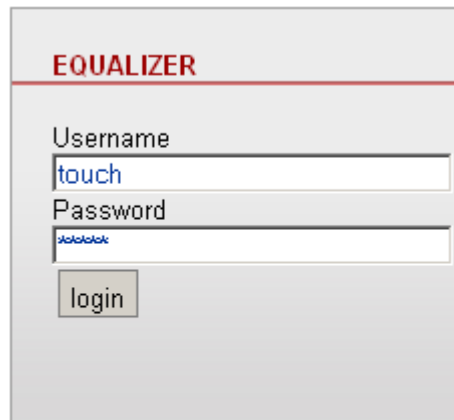


Figure 14 The login screen

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).

3. Enter **touch** (administrator), **look** (read-only), or another defined login as **Username**. Enter the **Password** for the login.

Note – Initial passwords for the **touch** and **look** logins are “touch” and “look”, respectively. These passwords can be changed and additional user logins defined as shown in “Managing Multiple Interface Users” on page 56.

Click the **login** button to log into Equalizer. The **Home** screen of the Administrative Interface is displayed, as shown on the following page.

The screenshot shows the Equalizer Administration Interface. The top header is red with the CoyotePoint Systems Inc logo on the left and 'Log Out' and 'Help' buttons on the right. A left sidebar contains a tree view of system objects: Mode: Standalone, Equalizer (with sub-items L4-HTTP, L4-HTTP-AB, L7-HTTP, sv00, sv01, Default, L7-HTTPS), Responders, Envoy, and Connections. The Connections section shows statistics: L4 processed: 1343825, L4 peak: 15821/s, L4 timeouts: 3, L7 active: 0, L7 processed: 189416, L7 peak: 1000. The main content area has a 'Welcome to the Equalizer Administration Interface!' message with instructions to click or right-click on objects in the tree. Below this is a photograph of the Equalizer Traffic Management Appliance (Model: E650ax) and a table of system information.

Equalizer System Information	
current user	touch
user permissions	administrator
Equalizer version	8.5.0a
system ID	00304866a966
serial no.	CP0807GX-011
platform	e650gx Rev. 1.0
system name	eq650gx-140

Figure 15 The Home Screen of Equalizer's Administration Interface

Navigating Through the Interface

The Equalizer Administration Interface (see Figure 15) is divided into three major sections:

- The left side of the screen displays a hierarchical list of objects, as explained below.
 - Mode: Standalone** or, if failover is configured, the *Failover Peer Name* defined for the other Equalizer in the failover pair.
 - The Coyote Icon indicates the failover status of the peer: a sitting Coyote means the peer is in backup mode (or failover is not enabled); a running Coyote means the peer is in primary mode.
 - Click this item to open the **Failover** configuration tab on the right hand side of the screen.
 - Right-click this item to open a menu that displays the currently configured IP addresses and a menu of failover commands.
 - eq_IP-address**, where *IP-address* is the IP address assigned to the Default VLAN (VLAN 1) -- see "Configuring the Default VLAN on E350/450/650GX" on page 43):
 - The Coyote Icon indicates the failover status of this Equalizer: a sitting Coyote means this Equalizer is in backup mode; a running Coyote means this Equalizer is in primary (or standalone) mode.
 - Click this item to open the **Clusters > General** tab in the right frame -- a summary of the configuration and status of all currently defined clusters and servers.
 - Click the plus sign (+) next to **Equalizer** to open a list of currently defined clusters.

- Click the plus sign next to a cluster name to open a list of currently defined servers and (for Layer 7 clusters) a list of Match Rules.
- Click a cluster, server, or match rule name to open the management tabs for that object.
- Right-click on this item to display the internal and external IP addresses, whether the system is in dual or single network mode, and the **Add Cluster** command.
- **Responders:**
 - Click **Responders** to open the **Responders** management tab, where you can add, modify and delete Responders. Right-click to open a menu containing the **Add New Responder** command.
 - Click the plus sign next to **Responders** to display a list of currently defined Responders in the tree.
 - Click a **Responder** name to open that Responder's configuration tab.
- The **Envoy** item (if the optional Envoy geographic load balancing software is installed):
 - Click the plus sign next to **Envoy** to display a list of currently defined GeoClusters in the tree.
 - Click the plus sign next to a GeoCluster name to display a list of the sites defined for the GeoCluster.
 - Click a GeoCluster or Site name to open the configuration tabs for that object.
- The **Connections** item:
 - Click on **Connections** to open the **Equalizer > Status > Statistics** screen.
 - Click on the plus sign to display L7 Statistics and L4 Statistics.

See "Displaying Global Connection Statistics" on page 201.

2. The top of the Administrative Interface screen displays the following buttons:
 - **Alert:** Displayed when a critical system message has been logged. Clicking on this icon displays the text of the system message.
 - **Log Out:** Logs you out of the Administrative Interface.
 - **Help:** Displays a sub-menu of commands:
 - **View Guide:** opens the *Equalizer Installation and Administration Guide* (this book) in PDF.
 - **View Release Notes:** opens the *Release Notes* for the currently installed version of Equalizer in PDF.
 - **View Transition Guide:** opens the *Equalizer Version 8 Transition Guide*, written to help Version 7 users locate Version 7 functionality in the Version 8 Administrative Interface.
 - **Context Help:** displays the section in the *Equalizer Installation and Administration Guide* PDF file corresponding to the screen currently displayed in the right frame.
 - **About:** the Equalizer **Home** screen displayed when you first log into the Administrative Interface.
3. The right hand side of the Administrative Interface initially displays the **Home** screen as shown in Figure 15 on page 53. For a description of the information contained on the **Home** screen, see "Displaying Equalizer System Information" on page 196.
 - Click on any item in the left frame, or right click to choose a command for that object. The right frame will display the management tabs for the object or the appropriate command dialog.
 - The easy-to-use management tabs organize configuration information into forms and tables that make configuring Equalizer simple. Sub-tabs provide a second level of organization within top-level tabs.

The following section shows you how to enable and disable access to the Administrative Interface over the available IP addresses and protocols, using the **Permissions** tab.

Managing Access to Equalizer


You can control the IP addresses and protocols on which the web-based Administrative Interface (the ‘GUI’ or graphical user interface) is available, and the IP addresses over which SSH (Secure Shell) access to Equalizer is permitted. By default, GUI and SSH access to Equalizer is enabled for each defined VLAN. Access can be restricted on a per-VLAN basis.

For more information on VLAN and network configuration, see the chapter “Equalizer Network Configuration” on page 63.

Note – To use SSH to access the Equalizer console, you must also enable the **eqsupport** login as described in the section “Managing the Remote Access Account” on page 47).

Viewing and Changing GUI and SSH Access

GUI and Secure Shell (**ssh**) access settings are specified for each currently defined VLAN. To view or change them, follow the procedure below. [Note that if Failover is configured, you cannot modify the VLAN configuration.]

1. Log into the Administrative Interface over one of the currently configured IP addresses. Use a login that has read or write access to global parameters (see “Objects and Permissions” on page 57).
2. Select **Equalizer > Networking > VLAN Configuration**.
3. Each row in the table corresponds to a VLAN. Click the modify icon  in the **Actions** column to display the **Modify VLAN** screen.
4. Under **Permissions**, the enabled flags indicate current access permissions:
 - **GUI http** enables GUI access on the VLAN IP via the **http://** protocol.
 - **GUI https** enables GUI access on the VLAN IP via the **https://** protocol.
 - **GUI Failover http** enables GUI access on this VLAN’s Failover IP (if specified) via the **http://** protocol.
 - **GUI Failover https** enables GUI access on this VLAN’s Failover IP (if specified) via the **https://** protocol.
 - **ssh** enables login to the console via SSH on the VLAN IP.
 - **Failover ssh** enables login to the console via SSH on this VLAN’s Failover IP (if specified).

Update the permission settings as desired.

5. Click **Commit** to save your changes.

Updating the Administration Interface Certificate

The Administration Interface is delivered with a default SSL certificate for **https://** connections. Clients use this certificate to authenticate a connection with the interface. You can replace this certificate by doing the following:

1. Log in to Equalizer using a login that has **add/del** access on global parameters (see “Logging In” on page 52).
2. Select **Equalizer > Maintenance > Certificates**.
3. Use the **Browse** button to select the certificate file from your local file system. The certificate file must be in PEM (*.pem*) or PKCS12 (*.pfx*) format, and must contain the private key and the entire certificate chain. (For more information on certificates, see Appendix E, “Using Certificates in HTTPS Clusters”.)
4. Select **upload** to install the new certificate on Equalizer. You’ll be prompted for a password, which is the password (or pass phrase) you provided when you generated the CSR for the certificate (or created the self-signed certificate).

Note: If you select a file that is not in PEM or PKCS12 format (or select no file at all), the following error message is displayed:

```
Certificate must be in PEM or PKCS12 format.
```

Following the error message is the output the SSL tools returned after they were run on the uploaded file. The output may be unreadable and poorly formatted; this is normal, because the file you uploaded was not in the correct format. Click **dismiss** on the error popup and then go back to the previous step to select a file that is in PEM or PKCS12 format.

Once the certificate file is successfully uploaded to Equalizer, the tab displays the certificate information at the bottom of the screen.

Managing Multiple Interface Users

Equalizer is shipped with two logins for the browser based Administrative Interface: **look** (read-only mode) and **touch** (administrator or edit mode). The definitions of these users and any additional users you create specify the permissions each has on Equalizer objects.

On Equalizer, there are two object types that are assigned permissions: clusters and global parameters. Cluster parameters include all cluster settings and the settings for the servers in the cluster. System parameters include network interface settings and user definitions. As installed, the **touch** user can add, modify, and delete global parameters, as well as add, modify, and delete clusters. The **look** user has read-only access to all global parameters and clusters.

These two logins are usually sufficient for sites that have a small number of system administrators. For sites where multiple administrators with different responsibilities exist, you can create additional logins that reflect the administrative roles assigned to each user who logs in to Equalizer.

Let’s say, for example, that your site has one person who is responsible for the overall administration of Equalizer’s clusters, users, and operating parameters (the Equalizer Administrator), and several junior system administrators, each of which is responsible for maintaining a single cluster (the Cluster Administrators). The Equalizer Administrator could use the **touch** login to create additional logins for each Cluster Administrator, and give each login permission to modify the configuration of a single cluster only.

Objects and Permissions

The following table shows the permissions and objects defined on Equalizer:

Permissions	Objects
none read write add/del	global parameters ALL cluster parameters

The permission set on the **ALL** object specifies the user's permission on all clusters with their permission set to **none** (the default), unless a different permission is set on the cluster. The table on the following page explains the permissions used on objects.

<p>none</p>	<p>The user cannot view, modify, or delete the object.</p> <p>For global parameters: the user cannot view any of the global parameter tabs displayed when you click on <i>Equalizer</i> in the left frame.</p> <p>For clusters: the left frame and all global tabs display only clusters that the user has been given explicit permission to view by assigning a higher permission to those clusters.</p> <p>When none is set for the ALL object, the user cannot view any clusters on the system. This can be overridden by setting a higher permission on individual clusters.</p>
<p>read</p>	<p>The user can only view the object's definition.</p> <p>For global parameters: the user can open all of the global parameter tabs displayed when you click on <i>Equalizer</i> in the left frame, but cannot use the commit button to make any changes.</p> <p>For clusters: cluster definitions for which the user has read permission are displayed in the left frame and all global tabs. The user can select clusters and view their definitions.</p> <p>A user with the read permission set for the ALL object has read permission on all clusters, unless write or add/del is set on an individual cluster.</p>
<p>write</p>	<p>In addition to read permission, the user can modify existing objects, but cannot add new objects or delete existing objects.</p> <p>For global parameters: the user can update all global parameters (including parameters that are not already assigned a value). The user cannot, however, add or delete global objects (for example: logins, clusters, and responders).</p> <p>For clusters: the user can modify the values assigned to all cluster parameters (including parameters that are not already assigned a value). The user cannot add or delete a cluster object (for example, a server or match rule.)</p> <p>A user with the write permission set for the ALL object has write permission on all clusters, unless add/del is set on an individual cluster.</p>
<p>add/del</p>	<p>In addition to write permission, the user can add new objects and delete existing objects.</p> <p>For global parameters: the user can add and delete global objects (for example: logins, clusters, and responders).</p> <p>For clusters: the user can add and delete a cluster object (for example, a server or match rule.)</p> <p>A user with the add/del permission set for the ALL object has add/del permission on all clusters with their permission set to none; if a permission other than none is set on the cluster, then the cluster permission applies.</p>

Figure 16 Permissions in user definitions

Viewing or Modifying Login Permissions

To view or modify the permissions for a login, do the following:

1. Log into the Administrative Interface using a login that has at least read access for global parameters (see “Logging In” on page 52).
2. Select **Equalizer > Permissions > Users**. The following table is displayed:






User Name	Description	Type	Actions
touch	user touch	Administrator	 
look	user look	Limited	 
			

Figure 17 Users table

3. To view or modify login details, select the modify icon  in the **Actions** column in the same row as the login name you want to view. The user definition appears, as shown in this example for the default **touch** login.

Modify User touch (?) (X)

user details

description

password

confirm password

Permission to modify system parameters and users

none read write add/del

cluster permissions

ALL	none	<input type="radio"/>	read	<input type="radio"/>	write	<input type="radio"/>	add/del	<input checked="" type="radio"/>
cl01	none	<input checked="" type="radio"/>	read	<input type="radio"/>	write	<input type="radio"/>	add/del	<input type="radio"/>
cl04	none	<input checked="" type="radio"/>	read	<input type="radio"/>	write	<input type="radio"/>	add/del	<input type="radio"/>
cl02	none	<input checked="" type="radio"/>	read	<input type="radio"/>	write	<input type="radio"/>	add/del	<input type="radio"/>
cl03	none	<input checked="" type="radio"/>	read	<input type="radio"/>	write	<input type="radio"/>	add/del	<input type="radio"/>

This screen contains the following information about the login:


user details	The description field contains a text description of the purpose of the login. The password field is empty when viewing a login definition.
Permission to modify system parameters and users	Specifies the permission the user has on the global system parameters (displayed when you select Equalizer > Global Configuration).

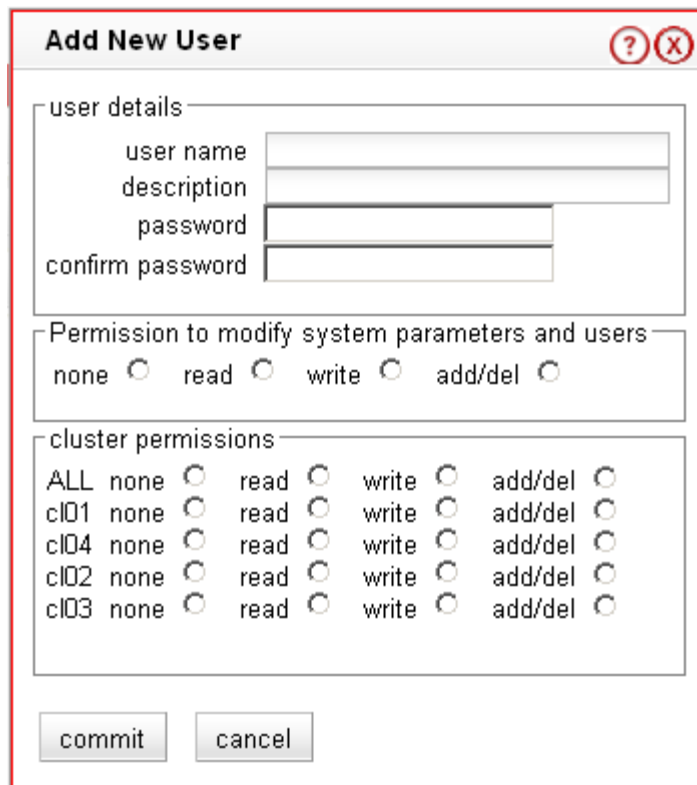
cluster permissions	ALL specifies the minimum permission the user has on all clusters below. Below ALL is a line for each cluster that specifies the permission that the user has on that cluster. If the permission on the cluster is to the left of the permission given for ALL , then the ALL permission applies to the cluster instead.
----------------------------	--

The screen above shows that the default user **touch** has **add/del** permission on system parameters and user definitions, and **add/del** on **ALL**. This means that **touch** has complete control over the system parameters, the user definitions, and all the clusters on Equalizer.

4. If you make any updates to the login password or permissions, click **Commit** to save your changes. Otherwise, click **Cancel** to return to the **Users** table.

Adding a Login

1. Log into the Administrative Interface over one of the currently configured IP addresses. Use a login that has at least read access for global parameters (see “Logging In” on page 52).
2. Select **Equalizer > Permissions > Users**.
3. Select the add icon . The **Add New User** screen is displayed:



Add New User [?] [X]

user details

user name

description

password

confirm password

Permission to modify system parameters and users

none read write add/del

cluster permissions

ALL	none <input type="radio"/>	read <input type="radio"/>	write <input type="radio"/>	add/del <input type="radio"/>
cl01	none <input type="radio"/>	read <input type="radio"/>	write <input type="radio"/>	add/del <input type="radio"/>
cl04	none <input type="radio"/>	read <input type="radio"/>	write <input type="radio"/>	add/del <input type="radio"/>
cl02	none <input type="radio"/>	read <input type="radio"/>	write <input type="radio"/>	add/del <input type="radio"/>
cl03	none <input type="radio"/>	read <input type="radio"/>	write <input type="radio"/>	add/del <input type="radio"/>

commit cancel


Figure 18 The add user screen

4. Type a **user name** and a **description** for the login. User names may only contain alphanumeric characters, periods (.), dashes (-), and underscores (_).
5. Type a **password** for the login and re-type it into the **confirm password** text box. Passwords must be between 6 and 128 characters long and should contain a mix of letters (uppercase and lowercase), numbers, and metacharacters. A blank password is not permitted.

6. Select the desired **permission to modify system parameters and users**. See the section “Objects and Permissions” on page 57 for help.
7. Select the desired **cluster permissions**. See the section “Objects and Permissions” on page 57 for help.
8. Select **commit** to save the user definition.

Deleting a Login

The Administrative Interface prevents you from deleting the login that you are currently using. For example, you cannot log in as **touch** and delete the **touch** login; to do this, you must log in using a different user name that has the **add/del** permission on users. This also prevents you from deleting all logins via the interface. However, it is possible that all user logins could be deleted by manually editing the configuration file, or in the unlikely event the configuration file becomes corrupted. If this occurs, the **eqadmin** utility can be used to create a new Administrators Read-Only login; see

1. Log in to Equalizer using a login other than the one you want to delete; the login you use must have the **add/del** permission on users (see “Logging In” on page 52).
2. Select **Equalizer > Permissions > Users**.
3. Select the delete icon  on the same row as the name of the user login you want to delete.
4. A confirmation box appears. Select **Commit** to delete the login.

Entering Names for Equalizer Objects

Equalizer identifies administrative objects, such as clusters and servers, by name. For example, object names and icons are displayed in a hierarchy in the Administrative Interface’s left frame as described earlier in this chapter. Keep in mind the following guidelines when typing in a name for an Equalizer object:

- The characters used in names are limited to standard ASCII letters (‘A’ through ‘Z’ and ‘a’ through ‘z’), numbers (0 through 9), and the characters ‘.’ (period), ‘-’ (dash) and ‘_’ (underscore).
- The first character in a name must be a letter.
- Names can be at most 63 characters long.
- The readability of lists presented in the interface is increased by using short names that use as many unique characters at the beginning of the name as possible.



VLAN Basics	64
Configuring VLANs on Equalizer	65
Initial VLAN Configuration Using the VLAN Wizard	66
Adding a VLAN in Standalone Mode	67
Modifying a VLAN in Standalone Mode	69
Deleting a VLAN in Standalone Mode	71
Adding a VLAN with Failover Enabled	71
Modifying a VLAN with Failover Enabled	72
Modifying a VLAN Name or Heartbeat Setting	72
Modifying VLAN Settings Other Than the Name or Heartbeat Setting	72
Deleting a VLAN with Failover Enabled	74
Managing Interface Ports	75
interface Administration Interface	76
Viewing Link Status	76
Viewing Current Port Settings	76
Editing Port Settings	77
Committing and Applying interface Port Configuration Changes	78
Switch Interface Usage Scenarios	79
Resetting the Front-Panel Interface Ports	79
Interface Notes for Pre-GX Equalizer Hardware	80
Configuring Static Routes	80
Adding a Static Route	80
Modifying a Static Route	81
Deleting a Static Route	81
Configuring Servers on Your Network	82
Configuring Routing on Servers	82
Server Configuration Constraints	82

VLAN Basics

Starting with Version 8.6, Equalizer models E350GX and above support tagged VLANs on both network interfaces. This section provides a basic technical introduction to VLAN technology.

For an overview of the VLAN configurations supported on Equalizer, see “Equalizer E350GX, E450GX, E650GX Network Configuration” on page 32.

Many networking technologies use a technique called *broadcasting* to provide services on a Local Area Network (LAN). Like traditional television or radio signals that are broadcast over the airwaves, broadcast network transmissions are received by every node on the same LAN segment, or *broadcast domain*. The Address Resolution Protocol (ARP), the Dynamic Host Configuration Protocol (DHCP), and the Router Information Protocol (RIP) are all examples of protocols that provide network services through broadcasting.

A LAN is a single broadcast domain composed of all the systems that are physically connected to the same switches, hubs, and other devices that communicate at the Data Link Layer (Layer 2) of the OSI Networking Model. These devices communicate using Layer 2 protocols, like Ethernet and ARP.

Virtual Local Area Network (VLAN) technology was developed to overcome these physical limitations of traditional LAN technology. A VLAN is essentially a means of grouping systems at the Data Link Layer (Layer 2 of the OSI networking model), using methods that are independent of the physical connection of the device to the network.

By exchanging *broadcast packets* -- packets that are essentially sent to all systems connected to a Layer 2 switching device -- switches can maintain a list of all MAC addresses connected to them and to the other switches to which they are connected. A set of Layer 2 devices and the systems connected to them form a *broadcast domain* -- meaning that all the systems can talk to one another using broadcast packets.

Conversely, broadcast packets are not forwarded beyond the boundaries of the broadcast domain. For example: if two LANs are connected by a router (a Network Layer, or Layer 3, device), the broadcast traffic for one LAN is never forwarded to the other LAN. The layout of a traditional LAN is therefore restricted to those systems that can be wired together using Layer 2 devices -- a physically distant system that requires connectivity to the LAN would require special routing and address translation (at Layer 3) in order to reach the LAN.

The dependence of LAN technology on physical connectivity at Layer 2 leads to two basic difficulties:

- broadcasts are received by all systems in the broadcast domain -- and if there is sufficient broadcast traffic, it can significantly reduce the overall performance of the LAN, to the point where some services may simply not be able to function properly due to latency or other factors introduced by a high level of broadcast traffic
- if you want to include a system that is not physically connected to the LAN in the LAN's broadcast domain, you need to physically connect the system to the LAN

One problem with broadcasting is that lots of broadcast traffic on a LAN can slow network traffic down, as well as slow individual systems down. If there is so much broadcast traffic on the LAN that other non-broadcast traffic is significantly delayed (or never delivered), this is called a *broadcast storm*. Broadcast storms typically arise when network loops are created through faulty network configuration, but can also happen as the result of a malicious attack. For example, a classic Denial of Service attack is to send an ICMP echo request ('ping') over the LAN that specifies the source address of a system and a broadcast address for the destination. Every system receiving the ping will respond to it -- flooding the system specified as the source of the ping with ICMP echo replies.

There are also other security concerns associated with broadcasting. Since all the systems in the broadcast domain can see broadcast packets, the information in them is susceptible to discovery, intercept, and modification. This is of particular concern in industrial Ethernet environments (where, for example, manufacturing processes are controlled directly by computers) and in any environment (such as government and finance) where sensitive data is regularly transmitted over the LAN.

A number of methods can be used to mitigate problems and threats associated with large broadcast domains, including broadcast filtering and physically separating large broadcast domains into smaller domains. The problem with these solutions is that they are typically implemented at the Network Layer (Layer 3), and require Layer 3 devices (such as routers and firewalls) to implement them. These Layer 3 devices require separate subnets, and themselves emit a significant amount of broadcast traffic.

What we really want is a way of abstracting the idea of a LAN so that large broadcast domains can be separated into smaller domains *without requiring any network rewiring or physical movement of systems*. We'd also like the ability to extend broadcast domains across Layer 3 devices to physically remote systems.

With a VLAN, the broadcast domain for a particular system is determined by the *software settings on the Layer 2 switch port to which the system is connected*.

So, for example, in a traditional LAN, all the systems connected to Switch A would be part of Broadcast Domain A. If the switch is a VLAN-capable switch, then it is possible to configure several ports on the switch for VLAN A, several others to VLAN B, others to VLAN C, and so on.





This allows you to both:

- reduce the number of devices in local broadcast domains
- extend broadcast domains across devices separated by more than one switch



The predominant VLAN standard is 802.1q. This standard adds a VLAN tag to the information in the Ethernet packet. Since they operate at the switching level, VLANs are Layer 2 technologies -- though they are often confused with Layer 3 subnetting, because in many configurations there is one VLAN configured per subnet. This is usually done for the practical purpose of allowing the systems on a VLAN to be managed as a group by other network management devices/software that work by IP address ranges, for example, rather than VLAN tags.

Configuring VLANs on Equalizer

Configured VLANs are listed on the **VLAN Configuration** tab. To open it, click on the Equalizer system name in the left frame and click on the **Networking** tab in the right frame:

Name	VID	Contents	Tagged Ports	Untagged Ports	Actions
Default	1		None	1, 2	
vlan2	2		None	3, 4, 5, 6, 7, 8, 9, 10, 11, 12	 

The following information is shown for each VLAN:

Name	The VLAN name.
VID	The VLAN ID, an integer between 1 and 4095.
Contents	Icons in this column indicate whether the VLAN hosts clusters () , servers () or both. Moving the mouse pointer over these icons lists the names of the clusters or servers configured on that VLAN.
Tagged Ports	The port numbers of the Tagged ports assigned to this VLAN. Tagged ports can be assigned to more than one VLAN.

Untagged Ports

The port numbers of the Untagged ports assigned to this VLAN. Untagged ports can be assigned to exactly one VLAN.

To see the detailed configuration for a VLAN, click on the **Modify** icon on that row. See the section “Modifying a VLAN in Standalone Mode” on page 69 for a description of the information displayed.

Initial VLAN Configuration Using the VLAN Wizard

A VLAN Wizard is provided to simplify initial VLAN configuration for common basic configurations.

Note – The VLAN Wizard has the following limitations:

It can configure at most a single management port and two VLANs for clusters and servers, as explained in Step 3 of the procedure below. More complex VLAN configurations must be set up using the VLAN Configuration tabs as explained in the sections “Adding a VLAN in Standalone Mode”, “Modifying a VLAN in Standalone Mode”, and “Deleting a VLAN in Standalone Mode”, later in this chapter.

It cannot be used to modify your VLAN configuration while Failover is enabled. Instead, use the VLAN Configuration tabs as explained in the sections “Adding a VLAN with Failover Enabled”, “Modifying a VLAN with Failover Enabled”, and “Deleting a VLAN with Failover Enabled”, later in this chapter.

To use the wizard:

1. Log into the Administrative Interface using a login that has **write** access for global parameters (see “Logging In” on page 52).
2. Select Equalizer’s system name in the left frame and open the **Networking > VLAN Configuration** tab in the right frame.
3. Click on the **VLAN Wizard** link, and choose the desired options for **Separate Default VLAN? (yes/no)** and for **Select VLAN mode (single/dual/advanced)**. The various option combinations create the following VLANs:
 - **no, single:** A single untagged VLAN where Equalizer, all clusters, and all servers reside on the same network segment. All ports will be assigned to this VLAN.
 - **yes, single:** 2 untagged VLANs: one for Equalizer (with port 1 assigned) and one for all clusters and servers (with the remaining ports assigned).
 - **no, dual:** 2 untagged VLANs: one VLAN for the Equalizer and all clusters (with ports 1 and 2 assigned), and one VLAN for servers (with the remaining ports assigned).
 - **yes, dual:** 3 untagged VLANs are configured, one for Equalizer management (with port 1 assigned), one for clusters (with port 2 assigned), and one for servers (with the remaining ports assigned).
 - **no/yes, advanced:** choosing **advanced** for **Select VLAN mode** closes the wizard and opens the **VLAN Configuration** tab so you can create a more complex VLAN topology.


Select the VLAN options desired and click the next icon (>) to continue.

4. Do *one* of the following:
 - If you selected **no, single** in the previous step, click **commit** to use the Default VLAN for all clusters, servers, and Equalizer management.
 - If you selected any of these options in the previous step:
 - **yes, single**
 - **no, dual**
 - **yes, dual**

Type in the **VID** and **VLAN IP** for the VLANs to be created in addition to the Default VLAN. Click **commit** to save your settings.

- If you selected **advanced** in the previous step, the wizard is closed and the **VLAN Configuration** tab is opened. Use the procedures in the following sections to add, modify, and delete VLANs.

Adding a VLAN in Standalone Mode

1. Log into the Administrative Interface using a login that has **write** access for global parameters (see “Logging In” on page 52).
2. Select Equalizer’s system name in the left frame and open the **Networking > VLAN Configuration** tab in the right frame.
3. Click on the **add** icon . The **Add New VLAN** screen appears:

Add New vlan

Name: VLAN IP:

VID: Netmask:

Permissions:

GUI http GUI Failover http

GUI https GUI Failover https

ssh Failover ssh

Failover IP:

Failover Netmask:

Use IP for Failover Heartbeat

port	status		type	
1	<input type="radio"/> assigned	<input checked="" type="radio"/> unassigned	<input type="radio"/> tagged	<input type="radio"/> untagged
2	<input type="radio"/> assigned	<input checked="" type="radio"/> unassigned	<input type="radio"/> tagged	<input type="radio"/> untagged
3	<input type="radio"/> assigned	<input checked="" type="radio"/> unassigned	<input type="radio"/> tagged	<input type="radio"/> untagged
4	<input type="radio"/> assigned	<input checked="" type="radio"/> unassigned	<input type="radio"/> tagged	<input type="radio"/> untagged
5	<input type="radio"/> assigned	<input checked="" type="radio"/> unassigned	<input type="radio"/> tagged	<input type="radio"/> untagged
6	<input type="radio"/> assigned	<input checked="" type="radio"/> unassigned	<input type="radio"/> tagged	<input type="radio"/> untagged

4. Fill out the fields as described in the table below:


Name	A descriptive name for the VLAN; must begin with an alphabetic character.
VID	A unique integer identifier for the VLAN, between 1 and 4095.
VLAN IP	Equalizer’s IP address on the VLAN.
Netmask	The netmask for the VLAN.
Failover IP	If failover is to be configured, the failover IP for the VLAN. The Failover IP has two purposes: it can be used as the floating gateway IP for servers behind Equalizer, and to provide GUI and SSH access to the Equalizer in the <i>primary</i> failover mode.

Failover Netmask	If failover is to be configured, the failover netmask for the VLAN.
Use IP for Failover Heartbeat	When enabled, Equalizer perform failover health check probes using the VLAN IP of the peer Equalizer on this VLAN. If this check box is <i>not</i> enabled, then failover will not occur if connectivity between the failover peers on this VLAN is lost.
Permissions:	<p>Enable any, all, or none of these check boxes to allow GUI access to Equalizer using the indicated protocols and IP addresses:</p> <p>GUI httpHTTP on the VLAN IP.</p> <p>GUI httpsHTTPS on the VLAN IP.</p> <p>GUI Failover httpHTTP on the Failover IP.</p> <p>GUI Failover httpsHTTPS on the Failover IP.</p> <p>Enable ssh access to Equalizer via one, both, or neither of the following IP addresses:</p> <p>sshssh on the VLAN IP.</p> <p>Failover sshssh on the VLAN IP.</p> <p>Note: The number of VLANs that can be enabled for ssh access is limited to 16; this limit cannot be changed.</p>
port status type	<p>Each row in the table at the bottom of the Add New VLAN screen corresponds to a front panel port. For a new VLAN, all ports have the status unassigned, meaning that they are not assigned to this VLAN. Clicking the assigned button allows you to set the VLAN type for the port: tagged or untagged. If a type appears with strikethrough text (e.g., untagged), then that type cannot be selected for that port.</p> <p>Note: it is not permitted to declare the same port as tagged in one VLAN and untagged in another.</p>

5. Click **commit** to create the VLAN.

Modifying a VLAN in Standalone Mode

This procedure explains how to modify your failover configuration when Equalizer is in **Standalone** mode; that is, it is *not* configured for failover. If you have a pair of Equalizers in a failover configuration, see the section ??? for how to make modifications to your VLAN configuration while failover is configured.

1. Log into the Administrative Interface using a login that has **write** access for global parameters (see “Logging In” on page 52).
2. Select **Equalizer > Networking > VLAN Configuration**.
3. Highlight the VLAN you want to change in the table and select the **Modify** icon . The **Modify VLAN** screen is displayed:

Modify VLAN

Name: VLAN IP:
 VID: Netmask:
 Failover IP:
 Failover Netmask:

Permissions:

GUI http GUI Failover http Use IP for Failover Heartbeat
 GUI https GUI Failover https
 ssh Failover ssh

port	status		type	
1	<input type="radio"/> assigned	<input checked="" type="radio"/> unassigned	<input type="radio"/> tagged	<input type="radio"/> untagged
2	<input type="radio"/> assigned	<input checked="" type="radio"/> unassigned	<input type="radio"/> tagged	<input type="radio"/> untagged
3	<input checked="" type="radio"/> assigned	<input type="radio"/> unassigned	<input type="radio"/> tagged	<input checked="" type="radio"/> untagged
4	<input checked="" type="radio"/> assigned	<input type="radio"/> unassigned	<input type="radio"/> tagged	<input checked="" type="radio"/> untagged
5	<input checked="" type="radio"/> assigned	<input type="radio"/> unassigned	<input type="radio"/> tagged	<input checked="" type="radio"/> untagged
6	<input checked="" type="radio"/> assigned	<input type="radio"/> unassigned	<input type="radio"/> tagged	<input checked="" type="radio"/> untagged

4. Edit the values shown as desired:


Name	A descriptive name for the VLAN; must begin with an alphabetic character.
VID	A unique integer identifier for the VLAN, between 1 and 4095.
VLAN IP	Equalizer’s IP address on the VLAN. Note that if you change the VLAN IP and you are using that IP for your current GUI session, you will lose your current GUI connection. Warning: Modifying the VLAN IP address of the Default VLAN requires a reboot. A popup window will request confirmation before rebooting the system.
Netmask	The netmask for the VLAN.

<p>Failover IP</p>	<p>If failover is to be configured, the failover IP for the VLAN. The Failover IP has two purposes: it can be used as the floating gateway IP for servers behind Equalizer, and to provide GUI and SSH access to the Equalizer in the <i>primary</i> failover mode.</p> <p>Warning: if you are logged into the GUI over the failover IP and you change the Failover IP, you will lose your current GUI connection.</p>
<p>Failover Netmask</p>	<p>If failover is to be configured, the failover netmask for the VLAN.</p>
<p>Use IP for Failover Heartbeat</p>	<p>When enabled, Equalizer perform failover health check probes using the VLAN IP of the peer Equalizer on this VLAN. If this check box is <i>not</i> enabled, then failover will not occur if connectivity between the failover peers on this VLAN is lost.</p>
<p>Permissions:</p>	<p>Enable any, all, or none of these check boxes to allow GUI access to Equalizer using the indicated protocols and IP addresses:</p> <p>GUI httpHTTP on the VLAN IP.</p> <p>GUI httpsHTTPS on the VLAN IP.</p> <p>GUI Failover httpHTTP on the Failover IP.</p> <p>GUI Failover httpsHTTPS on the Failover IP.</p> <p>Enable ssh access to Equalizer via one, both, or neither of the following IP addresses:</p> <p>sshssh on the VLAN IP.</p> <p>Failover sshssh on the VLAN IP.</p> <p>Note: The number of VLANs that can be enabled for ssh access is limited to 16; this limit cannot be changed.</p> <p>Warning: if you disable access permission for the protocol or IP address that you are currently using for your current GUI session, you will lose your current GUI connection.</p>
<p>port status type</p>	<p>Each row in the table at the bottom of the Add New VLAN screen corresponds to a front panel port. For a new VLAN, all ports have the status unassigned, meaning that they are not assigned to this VLAN. Clicking the assigned button allows you to set the VLAN type for the port: tagged or untagged. If a type appears with strikethrough text (e.g., untagged), then that type cannot be selected for that port.</p> <p>Note: it is not permitted to declare the same port as tagged in one VLAN and untagged in another.</p> <p>Warning: if you change VLAN port assignments, any current connections on the modified ports may be lost.</p>

5. Click **commit** to save your changes.

Deleting a VLAN in Standalone Mode

Note that you cannot delete a VLAN if failover is currently configured. This is because failover depends upon the failover IP addresses assigned on each VLAN. The Disable Failover flag on the Failover Peers tab allows you to temporarily disable failover in case you need to alter your VLAN configuration. See the section “Disabling the Failover Configuration” on page 105.

1. Log into the Administrative Interface using a login that has **write** access for global parameters (see “Logging In” on page 52).
2. Select **Equalizer > Networking > VLAN Configuration**.
3. Click the **Delete** icon  on the same row as the VLAN you want to delete. A confirmation screen appears.
4. Select **commit** to delete the route.

Note that you cannot delete the Default VLAN.

Adding a VLAN with Failover Enabled

When failover is enabled, you can add a VLAN to the configuration *without disabling failover* (that is, without going into Standalone mode) by following the procedure in this section. This allows you to add VLANs without disrupting cluster traffic and normal failover operation.

1. Log into the GUI on the *current primary* failover peer. Open the **Equalizer > Networking > VLAN Configuration** tab and click the add icon. Set up the VLAN following the instructions in the section “Adding a VLAN in Standalone Mode” on page 67, making sure that the **Use Failover IP for Heartbeat** check box is *disabled*.
2. Log into the GUI on the *current backup* failover peer. Open the **Equalizer > Networking > VLAN Configuration** tab and click the add icon. Set up the VLAN following the instructions in the section “Adding a VLAN in Standalone Mode” on page 67. If desired, you can enable the **Use Failover IP for Heartbeat** check box, or leave it disabled.
3. If you enabled the **Use Failover IP for Heartbeat** check box in the previous step, log into the GUI on the *current primary* failover peer. Open the **Equalizer > Networking > VLAN Configuration** tab and click the modify icon on the row that corresponds to the VLAN you added in Step 1. Enable the **Use Failover IP for Heartbeat** check box and click **commit**.

Modifying a VLAN with Failover Enabled

When failover is enabled, you can modify a VLAN *without disabling failover* (that is, without going into Standalone mode) by following the procedure in this section. This allows you to modify VLANs without disrupting cluster traffic and normal failover operation.

Note – The VLAN parameters you can modify when failover is enabled depend on the setting of the **Use Failover IP for Heartbeat** check box:

If the **Use Failover IP for Heartbeat** check box is *enabled*, only the following fields can be modified:

- **Name**
- **Use Failover IP for Heartbeat**

Both of the above fields can be modified at the same time.

If the **Use Failover IP for Heartbeat** check box is *disabled*, the following fields can also be modified:


- **VID (VLAN ID)**
- **VLAN IP Address**
- **Netmask**
- **Failover IP Address**
- **Failover Netmask**
- All check boxes under **Permissions**
- **Ports**

You must modify only one of the above fields at a time, as explained below.

Modifying a VLAN Name or Heartbeat Setting

Do the following to modify either or both of the following settings while failover is enabled:

- **Name**
- **Use Failover IP for Heartbeat**

1. Log into the Equalizer that is currently in *backup mode* and do the following:
 - a. Open the **Equalizer > Networking > VLAN Configuration** tab, highlight the VLAN you want to modify, and click the modify  icon.
 - b. Modify one or both of the **Name** and **Use Failover IP for Heartbeat** parameters.
 - c. Click **commit**.







Equalizer displays a progress dialog while your changes are saved. **Wait until the progress dialog closes before proceeding with the next step.**

2. Log into the Equalizer that is currently in *primary mode* and perform exactly the same change on same VLAN on the primary unit, as described in Step 1.

Modifying VLAN Settings Other Than the Name or Heartbeat Setting

Do the following to modify the following settings while failover is enabled:

- **VID (VLAN ID)**
- **VLAN IP Address**
- **Netmask**
- **Failover IP Address**
- **Failover Netmask**
- All check boxes under **Permissions**
- **Ports**

1. Log into the Equalizer that is currently in *backup mode* and do the following:
 - a. Open the **Equalizer > Networking > VLAN Configuration** tab, highlight the VLAN you want to modify, and click the modify  icon.
 - b. Disable the **Use Failover IP for Heartbeat** check box.
 - c. Click **commit**.
2. Log into the Equalizer that is currently in *primary mode* and do the following:
 - a. Open the **Equalizer > Networking > VLAN Configuration** tab, highlight the same VLAN that you modified in the previous step, and click the modify  icon.
 - b. Disable the **Use Failover IP for Heartbeat** check box.
 - c. Click **commit**.
3. On the Equalizer in *backup mode*, do the following:
 - a. On the **Equalizer > Networking > VLAN Configuration** tab, highlight the VLAN you modified in Step 1 and click the modify  icon.
 - b. Modify *one* of the following parameters:
 - VID (VLAN ID)
 - VLAN IP Address
 - Netmask
 - Failover IP Address
 - Failover Netmask
 - One of the **Permissions** check boxes
 - Ports
 - c. Click **commit**.
4. On the Equalizer in *primary mode*, do the following:
 - a. On the **Equalizer > Networking > VLAN Configuration** tab, highlight the same VLAN that you modified in the previous step and click the modify  icon.
 - b. Make the same modification as you did in the previous step.
 - c. Click **commit**.
5. If you want to modify another parameter on this VLAN, perform Step 3 and Step 4 again. Otherwise, go to the next step.
6. If desired, re-enable the **Use Failover IP for Heartbeat** check box on the VLAN that you modified on the *current backup*:
 - a. On the **Equalizer > Networking > VLAN Configuration** tab, highlight the VLAN you modified in Step 3 and click the modify  icon.
 - b. Enable the **Use Failover IP for Heartbeat** check box.
 - c. Click **commit**.
7. If you re-enabled the **Use Failover IP for Heartbeat** check box on the VLAN that you modified on the *current backup* in the previous step, re-enable it on the *current primary*:
 - a. On the **Equalizer > Networking > VLAN Configuration** tab, highlight the same VLAN modified in the previous step and click the modify  icon.
 - b. Enable the **Use Failover IP for Heartbeat** check box.
 - c. Click **commit**.
8. If you want to modify another VLAN, perform this procedure again.

Deleting a VLAN with Failover Enabled

To delete a VLAN on an Equalizer peer in a failover configuration, do the following:

1. If the **Use Failover IP for Heartbeat** check box is *enabled* on the VLAN you want to delete, disable it following the appropriate procedure in the previous section, “Modifying a VLAN with Failover Enabled” on page 72. Otherwise, go to the next step.
2. Log into the GUI on the *current backup* failover peer. Open the **Equalizer > Networking > VLAN Configuration** tab, highlight the VLAN you want to delete and click the delete icon. Click the **delete** button in the confirmation popup that appears to delete the VLAN.
3. Log into the GUI on the *current primary* failover peer. Open the **Equalizer > Networking > VLAN Configuration** tab, highlight the VLAN you want to delete and click the delete icon. Click the **delete** button in the confirmation popup that appears to delete the VLAN.

Managing Interface Ports

This section does not apply to the E250GX and to legacy E350si and E450si Equalizers. See the Note in the text.

All Equalizer GX models have two Ethernet adapters on the motherboard. The interface ports on the front panel are connected to both adapters. This allows any port to be configured to work with either adapter.

In the default configuration, interface ports #1 & #2 are connected by VLAN to one of the motherboard interfaces (emulating the 'external interface' port provided on older 'si' hardware models). The remainder of the ports are configured on a separate VLAN connected to the other motherboard port (emulating the 'internal interface' ports provided on the 'si' hardware models).

Note – Equalizer model E250GX does not support VLANs and does not support the **Switch Configuration** interface described in this section. The E250GX has two ports, each of which is connected to one of the Ethernet adapters on the motherboard. The port configuration of the E250GX is managed using the **eqadmin** console interface, as described in the section “Configuring External and Internal Interfaces on E250GX” on page 42.

Also note that the **Switch Configuration** tab is unavailable (grayed-out) on legacy E350si and E450si model Equalizers. It is supported on the E550si and E650si. (The E250si is not supported in EQ/OS Version 8.6.)

Version 8.5 provides GUI-based user configuration for these "port based" VLANs, so that the user can select any set of ports as internal or external. Note that the #1 and #2 ports are identified with red labels to remind the user that these are configured by default into the Default VLAN, as shown in the figure below.

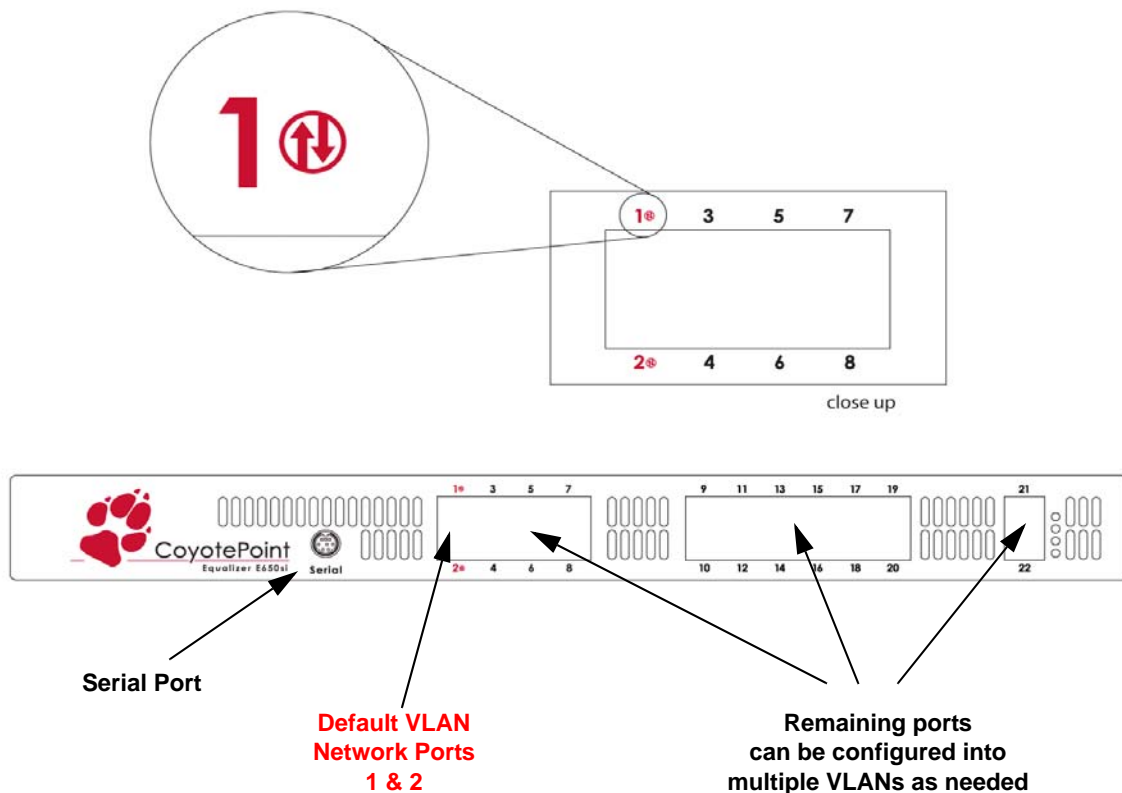
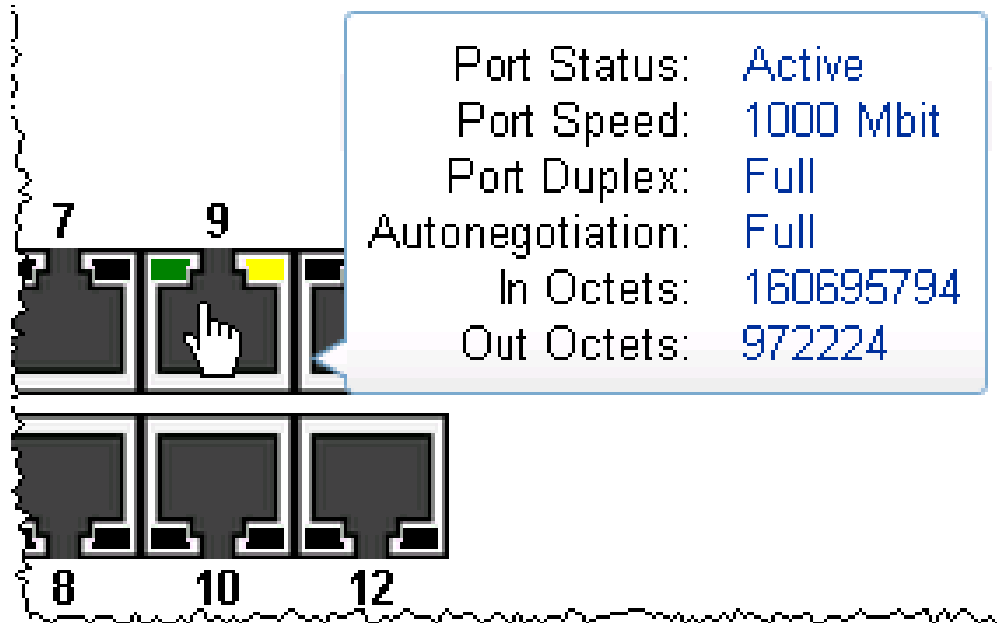


Figure 19 Equalizer 650GX Front Panel

interface Administration Interface

The **Switch Configuration** interface allows you to easily view and modify the configuration of each port on Equalizer's front panel. Click on *Equalizer* in the left frame, and then click **Status > Networking Configuration** in the right frame to display the **Switch Configuration** tab:

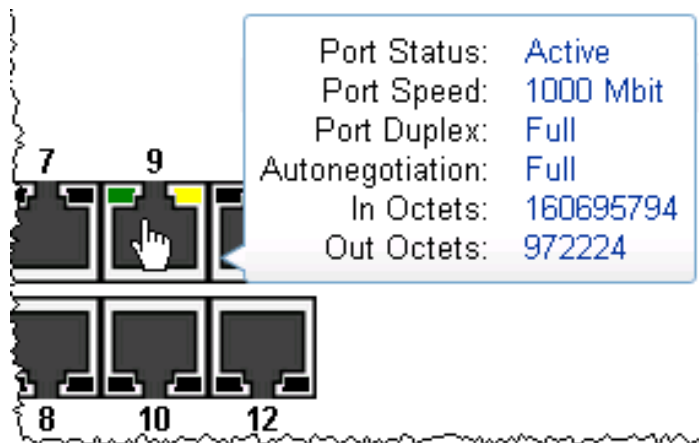


Viewing Link Status

The current link status of each port is displayed graphically, as shown in the Legend. Ports whose configuration have been modified but not committed (saved) are displayed with a red border and a coyote paw print.

Viewing Current Port Settings

To display the configuration settings for any port, mouse over the port to display a popup screen with the current parameters. Mousing over a modified port displays the modified (i.e., not yet committed) parameters.



The port setting displayed in the mouseover popup are explained in the table below.

Port Status	Displays Active if the port has an active link, No Link if not.
Port Speed	If the port is Active , this is the current port speed. If there is No Link , this is the highest speed that can be negotiated, or the forced speed setting.
Port Duplex	If the port is Active , this is the current port duplex setting. If there is No Link , this is the highest duplex that can be negotiated, or the forced duplex setting.
Autonegotiation	Can be one of the following: Full (full autonegotiation at all support speeds and duplex settings), Select (autonegotiation at only selected speeds and duplex setting), and Force Speed/Duplex (set the port to a particular speed and duplex).
In Octets Out Octets	In Octets and Out Octets are the current count of bytes received and sent (respectively) through this interface port -- this includes <i>all</i> traffic, including broadcast traffic as well as traffic not destined for the Equalizer. Note that these counters do <i>not</i> indicate the number of bytes sent and received through the port since the last reboot. Each counter has a limit of $2^{32} - 1$, and when this limit is reached, the counter is restarted.

Editing Port Settings

To edit the settings for any port, click on the port. A popup screen is displayed with the current parameters. Make changes to the port parameters and click **Apply**. Changes to port settings take effect immediately, but will not persist across system reboots until saved using the **commit** button on the **Switch Configuration** tab.

The parameters that appear on the **Configure switch port** popup depend on the current **Autonegotiation** setting of the port, which can be set to **Full**, **Select**, or **Force Speed/Duplex**.

When **Autonegotiation** is set to **Full** (the default setting), the popup looks like this:

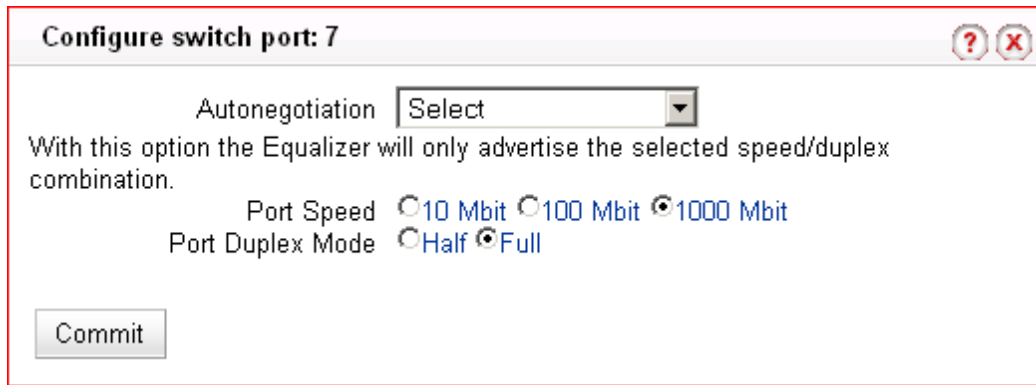
Configure switch port: 7 ? X

Autonegotiation

With this option the Equalizer will auto-negotiate over its full range of supported speed and duplex capabilities. That is, it will advertise 1000BASE-T, 100BASE-X, and 10BASE-T, full duplex and half duplex.

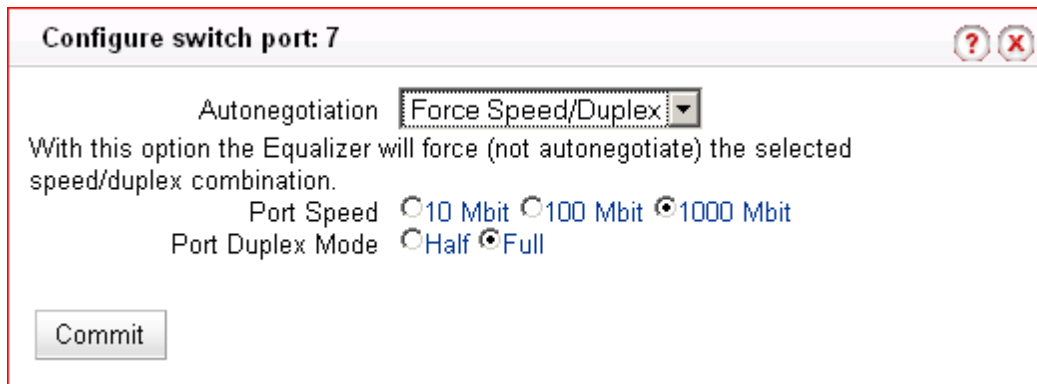
With **Autonegotiation** set to **Full**, Equalizer will advertise its full range of speed and duplex settings on this port. The actual negotiated speed and duplex depend on the outcome of the autonegotiation process between the port and the device on the other end of the connection. This is the recommended setting for the majority of modern networking devices, and is the default autonegotiation setting for all interface ports.

When **Autonegotiation** is set to **Select**, the popup looks like this:



Setting **Autonegotiation** to **Select** allows you to set a specific speed and duplex to be negotiated with the device on the other end of the connection to this port. Only this speed and duplex combination is advertised for autonegotiation.

When **Autonegotiation** is set to **Force Speed/Duplex**, the popup looks like this:



The **Force Speed/Duplex** setting is similar to the **Select** setting in that only one speed and duplex combination is chosen, but with the Force setting there is no negotiation with the device on the other end of the connection.

Whether you choose **Full**, **Select**, or **Force Speed/Duplex** really depends on the operating characteristics of the device on the other end of the connection. Some older network devices do not correctly autonegotiate, and Equalizer may (for example) be able to determine the correct speed setting, but cannot determine the duplex setting. For such devices, the Force setting is probably appropriate. Similarly, some devices set to autonegotiate may not work with a forced speed/duplex setting, even if that setting is correct for the device.

Committing and Applying interface Port Configuration Changes

As shown in the previous section, the **Configure switch port** popup has a **commit** button. When you click **commit** on the popup, the changes you have made are written to the interface configuration. They will not, however, take effect on the port until you click **apply** on the **Switch Configuration** screen (or reboot the system).

Caution – Be sure that any changes you make to Equalizer’s VLAN configuration are compatible with your current network configuration. It is possible to lock Equalizer out of the network if you mis-configure the interface VLANs. If you misconfigure the interface ports and can no longer access Equalizer over the network, see the section “Resetting the Front-Panel Interface Ports” on page 79.

Switch Interface Usage Scenarios

Some suggestions for using the new switch interface:

- In a single network configuration, the two external VLAN ports are unused. Ports #1 & 2 can be re-configured as part of the internal VLAN, adding two more server ports.
- Switch ports can be configured to match specific port settings required by the connected server. For example, you could use the switch interface to configure a particular switch port to be 100Mb/s and half-duplex to accommodate older hardware.
- The graphic interface port display provides dynamic status feedback. The lights for each port, for example, indicate the current status of the port without a browser refresh. This is a handy way for checking port status when you are accessing the system remotely.

Resetting the Front-Panel Interface Ports

A new **eqadmin** console utility option allows you to reset the VLAN configuration of the interface ports in the unlikely event that misconfiguration of the switch disables access to Equalizer's VLAN IP addresses.

Caution – Resetting VLANs via **eqadmin** should only be done when the system is not passing any traffic, since all existing connections will be dropped when the VLAN configuration is reset. The following operations are performed:

- All VLANs except for the Default VLAN are removed from the system. The Default VLAN is reset to use ports 1 and 2, and retains its currently configured IP address.
- The failover configuration is completely removed.
- The system is rebooted immediately after the VLAN configuration is reset.

To reset the VLAN configuration:

1. Log into Equalizer via the serial port or **ssh** (if enabled). As *root*, enter **eqadmin** and press **Enter** to display the **Equalizer Configuration Menu**:
2. Press **'0'** or use the arrow keys to scroll down to the last option on the screen, **Reset VLANs**, and press **Enter**.
3. A confirmation screen is displayed. Select **Yes** and press **Enter**. The change takes effect immediately.
4. Do one of the following:
 - Select option **'1'** to configure the default VLAN, as shown in “Performing Basic Equalizer Configuration” on page 41.
 - Select option **'6'** to **Commit and Reboot** Equalizer. After the system comes back up, configure the Default VLAN, as shown in “Performing Basic Equalizer Configuration” on page 41.

Note that the **Network Configuration** screen will still display the previously configured interface settings, even though they are no longer in effect. You can accept the previous settings or supply new settings.

Interface Notes for Pre-GX Equalizer Hardware

The switch management interface is primarily intended for the managed gigabit switches included in ‘GX’ model Equalizers (as described above), and two earlier models: the E550si and E650si. If Version 8.5 or later is installed on an older model E550si or E650si Equalizer, the switch management interface is also available on these models with reduced functionality:

1. The **Ext** port on the front panel of the ‘si’ unit is *not* managed by the **Switch Configuration** interface.
2. The **speed**, **duplex**, and **autonegotiation** settings for all internal ports (ports numbered starting at “1”) can be set via the **Switch Configuration** tab, but VLANs are not supported (the VID cannot be set).

Note – Tthe **Switch Configuration** tab is unavailable (grayed-out) on legacy E350si and E450si model Equalizers.

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 Administrative Interface using a login that has **add/del** access for global parameters (see “Logging In” on page 52).
2. Select **Equalizer > Networking > Static Routes**:

Name	Type	Network/Host	Gateway	Actions
net172	-net	172.16/16	10.0.0.172	

Figure 20 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.
----------------	---

- Click on the **Add** icon . The **Add New Route** screen appears:

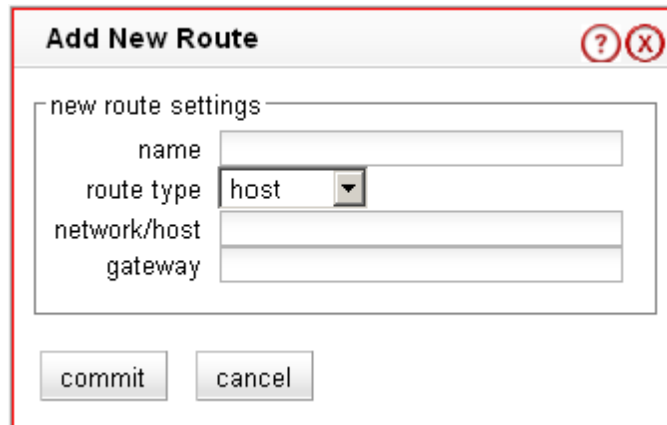




Figure 21 The add new route screen

- Enter the parameters for the route, and select **commit**. You are returned to the **Static Routes** table, which now displays the route you added.

Modifying a Static Route

- Log into the Administrative Interface using a login that has **write** access for global parameters (see “Logging In” on page 52).
- Select **Equalizer > Networking > Static Routes**.
- Highlight the route you want to change in the table and select the **Modify** icon . The **Modify Route** screen is displayed:
- 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

- Log into the Administrative Interface using a login that has **write** access for global parameters (see “Logging In” on page 52).
- Select **Equalizer > Networking > Static Routes**.
- Highlight the route you want to delete in the table and select the **Delete** icon . A confirmation screen appears.
- Select **commit** to delete the route. You are returned to the **Static Routes** screen, from which the route has been removed.

Configuring Servers on Your Network

Configuring Routing on Servers

In configurations where the cluster **spoof** option is enabled, you should configure your servers so that Equalizer gateways the packets the servers send to clients. In most cases, the easiest way to do this is to specify an IP address on Equalizer as the server default gateway in its routing tables. If you do not adjust the routing on your servers when the **spoof** option is enabled, servers will not route responses through Equalizer and clients receiving such responses directly from servers will drop the responses and the client connection will time out.

Direct Server Return (DSR) configurations, however, are an exception to this rule. In DSR configurations, client requests coming through Equalizer are routed to servers, which then respond directly back to the clients without going through Equalizer. Therefore, servers in a DSR configuration typically have a default gateway other than Equalizer.

Otherwise, in non-DSR clusters with **spoof** enabled, you should use one of the following Equalizer addresses as the default gateway on the servers in the cluster:

- **If the servers are connected to a single (standalone) Equalizer**, the default gateway IP address that you should use on the server is Equalizer's IP address on the VLAN associated with the Equalizer front-panel port to which the server is connected.
- **If the servers are connected to two Equalizers in a failover configuration**, the default gateway IP address that you should use on the server is always Equalizer's failover IP address on the VLAN associated with the Equalizer front-panel port to which the server is connected.

The commands or utilities that you use to configure routing on a server depends on the server's operating system, but usually involves some form of the `route` command; check your server's operating system documentation. 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 routing on each server from the server's system console, not through a telnet session. This will avoid any disconnects that might otherwise occur as you adjust the network settings on the server.

Server Configuration Constraints

When configuring servers into clusters on Equalizer, you must observe the following constraints:

- In order for Equalizer to initialize properly when it boots, there must be at least one running server that is responding to Equalizer probes, and there must be no Layer 3 devices (e.g., such as a router) between that server and Equalizer.
- Servers can be located behind routers, but we do not recommend or support such configurations. Various Equalizer subsystems (such as the health check probing subsystem) depend on reliable communication between Equalizer and the servers behind it, and the latency introduced by Layer 3 devices such as routers can, for example, result in connection timeouts even when the server is available.
- If you do require remote servers in your clusters, you must be very careful to configure network routes that allow Equalizer and the server to communicate.

An example of a configuration with both directly connected servers and remotely accessible servers is illustrated in the diagram below.

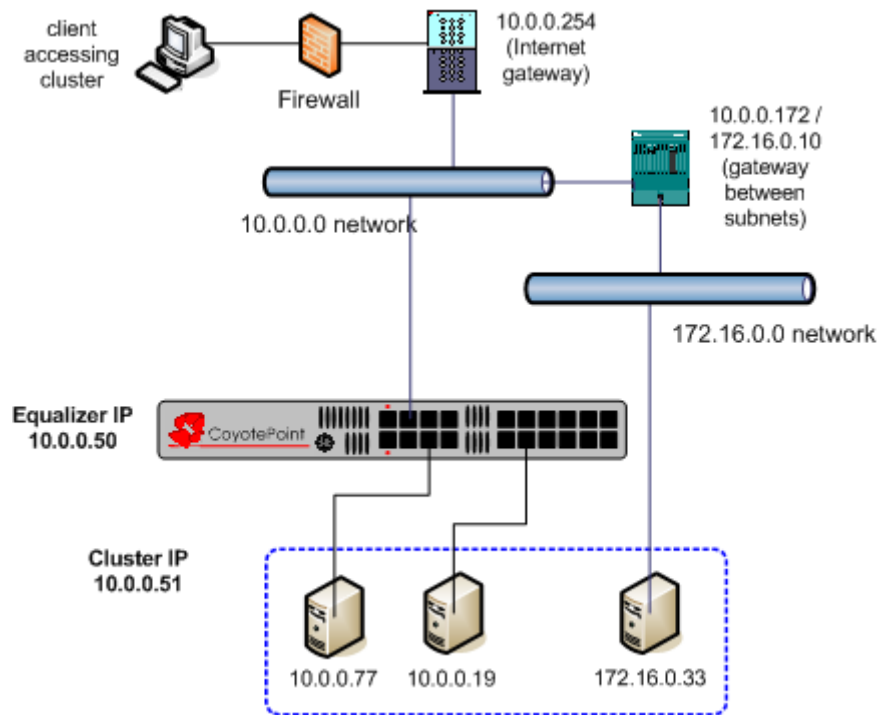


Figure 22 Example single VLAN configuration with local and remote servers

The configuration in Figure 22 is an example of a single VLAN configuration, where Equalizer communicates with all servers and clients via the same subnet. The example cluster shown above contains three servers, two on the local 10.0.0.0 subnet, and one on another subnet.

In this example, a static route would be needed on Equalizer to forward all packets for the 172.16.0.0 network to the gateway at 10.0.0.172. Similarly, the server at 172.16.0.33 would need a static route that forwards all traffic for the 10.0.0.0 network through 172.16.0.10, the gateway address for the 10.0.0.0 network.

Configuring Equalizer Operation



This chapter describes the global parameters, resources, and procedures that you can use to specify Equalizer's operating characteristics and perform system maintenance tasks, including failover:

Licensing Equalizer	86
Requesting a License Online	86
Requesting a License Offline	88
Modifying Global Parameters	89
Global Probe Parameters	90
Global Networking Parameters	91
Setting Up a Failover Configuration	95
General Failover Operation	95
Failover Issues with Spanning Tree	95
Failover Determination	95
Failover Between Different Hardware & Software Releases	96
Using Release 8.6 with Release 8.5.1, and GX or 'si' Hardware	96
Using a GX and an 'si' in Failover with Version 8.6.....	97
Required VLAN Configuration for Failover in Release 8.6	97
Failover and VLAN Configuration in Version 8.6.0f and Later Versions.....	97
Failover and VLAN Configuration in Versions Prior to Version 8.6.0f	98
Setting Up or Modifying Failover Using the Failover Wizard	99
Enabling Failover Using the Failover Tabs	100
Manually Enabling Failover	101
Modifying the Failover Configuration	105
Disabling the Failover Configuration	105
Re-enabling Failover After Disabling	106
Clearing the Failover Configuration	107
Changing from Multi-VLAN to Single-VLAN Configuration	107
Managing System Time and NTP	109
NTP and Plotting	109
Selecting an NTP Server	110
General System Maintenance	112
Creating a Backup Archive	112
Restoring a Backup Archive	113
Restoring a Backup Archive on an Equalizer in Standalone Mode	113
Restoring a Backup Archive on an Equalizer in a Failover Pair	114
Shutting Down Equalizer	116
Rebooting Equalizer	116
Creating a System Information Archive	116
Upgrading Equalizer Software	117

Licensing Equalizer

You must register and license your Equalizer before performing any other configuration using the Equalizer Administration Interface (described in Chapter 3, “Using the Administration Interface”). The License Manager helps you register and request a license, as well as view your current license information.

You’ll need to request a license if:

- The left frame of the Equalizer Administrative Interface displays an unlicensed system error.
- You’ve purchased the Envoy Geographic Clustering product after previously licensing Equalizer.
- You want to upgrade to a new release that requires a new license.

You can register and license your Equalizer either *online* using Coyote Point’s Licensing Server, or *offline* using a license file obtained from Coyote Point Support via email.

Requesting a License Online

To request a license using this method, Equalizer needs to be able to connect to the internet on port 127. You may need to work with your local network administrator to ensure that Equalizer can connect to the license server through any firewalls or other network devices on your network. If this is not possible, you can use the offline method (see “Requesting a License Offline” on page 88).

Follow this procedure to obtain a license directly from the Coyote Point License Server:

1. Log into the Administrative Interface using the default **touch** login, or another login that has **add/del** access for global parameters to request a license; **read** access or greater to view (see “Logging In” on page 52).
2. Select the Equalizer system name in the left frame and open the **Maintenance > License Information** tab in the right frame. The following screen is displayed:



Figure 19 The License Information screen -- Online License

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

product	Equalizer product model number. Displays “unlicensed” if the system is not licensed or the current license is invalid.
feature	Lists any add-on products (such as Envoy) that are enabled by your current license.
servers per cluster	The number of servers per cluster allowed, as specified by your license.
supported clusters	If your Equalizer model has a limit on the number of clusters, this line displays the limit (e.g., the E250GX supports up to 64 clusters)
serial no.	The serial number of the Equalizer unit (also printed on the back or bottom of the unit).
system ID	The internal system ID.

If you don’t need to license Equalizer, stop now. Otherwise, continue with the next step.

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

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** into the registration form (see Figure 19 on page 86).

4. Do *one* of the following:

- 4a. 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.
- 4b. If Equalizer is not connected to the Internet or DNS is not configured, then see the section “Requesting a License Offline” on page 88, below.

After the system comes back up, there should be no unlicensed error in the left frame or on the **Help > About** screen. If you licensed Envoy, the **Help > About** screen should show **Envoy geographic load balancing enabled** when the **Equalizer System Information** box is expanded.

Requesting a License Offline

You might also create a license request for other reasons, such as the first time you install Equalizer.

Requesting an offline license is necessary when you either power up Equalizer for the first time or upgrade it to a new software release, and Equalizer cannot connect to the Internet (usually because of network restrictions). If for any reason you are manually installing a new license for a production Equalizer, you should use these procedures only when you are able to completely shut the Equalizer down and start it up again.

On powering up Equalizer for the first time, the GUI will display an unlicensed error, which also appears in the logs.

After an upgrade image is downloaded onto Equalizer, the upgrade script checks for a valid license and stops the upgrade process until a valid license for the release that is being installed is detected.

If Equalizer is not currently able to connect to the Internet for an online license, you will need to request a license offline, install the license you get back from Coyote Point, and then restart the upgrade. License requests are created using the GUI.

Note that when an upgrade image is present on the system and you open the offline licensing GUI, you will be able to choose between creating a license request for the upgrade release or the currently running release. This is shown in the procedure below. If no upgrade release is present, then the option to generate a request for an upgrade release will not be displayed, and the GUI will only create a license request for the currently running release.

To request an offline license, do the following:

1. Follow Steps 1 through 3 of the procedure above.
2. Select **Offline License** on the **License Information** screen. The **Offline License** panel expands:

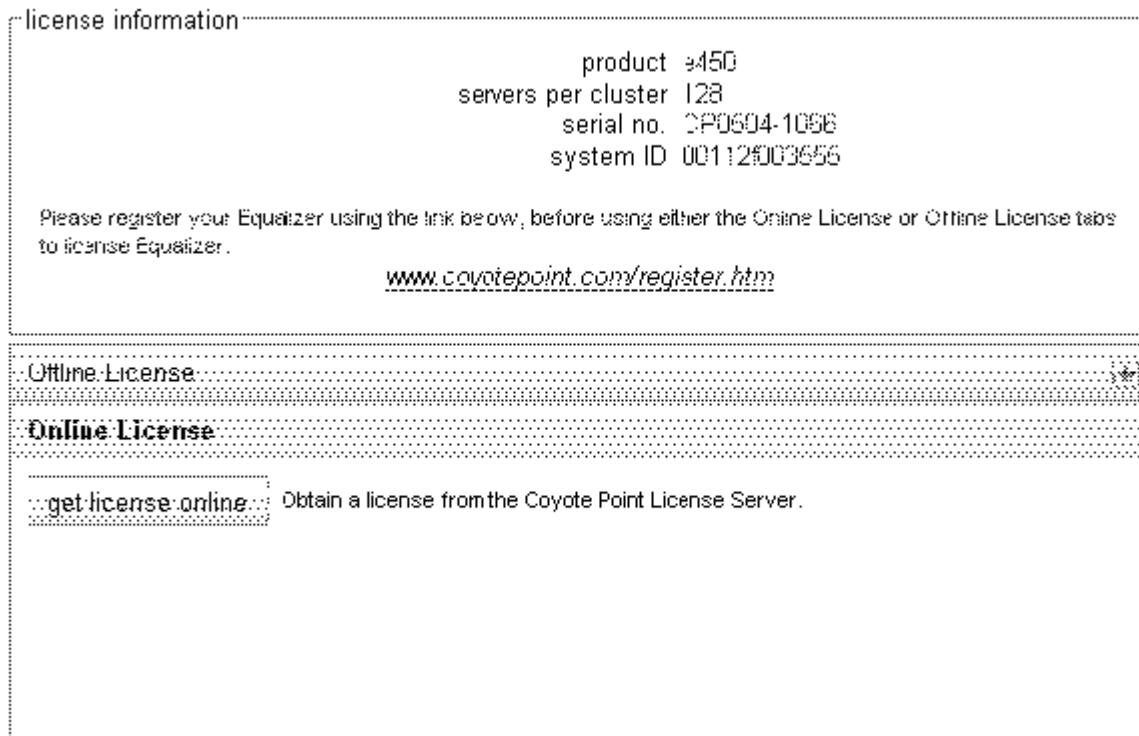


Figure 20 The License Information screen -- Offline License

3. If the two radio buttons in the illustration above are not present, skip this step. Otherwise, the system has detected that an upgrade release is present on the system, and you must choose the release for which you want to create a license request:
 - To create a license request for the upgrade version that is present on your system, select **Generate request for upgrade version** (the default).
 - To create a license request for the currently running version, select **Generate request for current version**.
4. Select **create license request file** and save the file to an appropriate location on your local system.
5. 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.
6. 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.
7. Select **install signed license file** and use the browse box to select the signed license file you saved in the previous step.
8. Equalizer installs the license and asks you if you want to reboot to apply the license. Select **reboot** to reboot.

If the license you installed above was for the currently running system, then after the system comes back up, there should be no unlicensed error in the left frame or on the **Help > About** screen. If you licensed a separately licensable feature (such as Envoy), the **Help > About** screen should show that the feature is enabled when the **Equalizer System Information** box is expanded.

If the license you installed above was for an upgrade image, you may now restart the upgrade procedure.

Modifying Global Parameters

Global or System Parameters are divided into two tabs, Probes and Networking. Most clusters will work with default values on these tabs. To view or modify the default global parameter values:

1. Log into the Administrative Interface using a login that has **add/del** access for global parameters to add, remove, and update parameters; **write** access to update parameters with existing values; **read** access or greater to view (see "Logging In" on page 52).
2. Select **Equalizer > Probes** to view/modify the global probe parameters, or **Equalizer > Networking** to view/modify the global networking parameters.
3. Change the appropriate fields.
4. Click the **commit** button.

The following two sections explain the global probe and networking parameters.

Global Probe Parameters

Selecting **Equalizer > Probes** displays the global probe parameters:

probe parameters

probe interval	<input style="width: 80%;" type="text" value="20.0"/>
probe timeout	<input style="width: 80%;" type="text" value="10.0"/>
strikeout threshold	<input style="width: 80%;" type="text" value="3"/>
event interval	<input style="width: 80%;" type="text" value="15.0"/>
probe delay	<input style="width: 80%;" type="text" value="10.0"/>
agent delay	<input style="width: 80%;" type="text" value="10.0"/>
require agent response	<input type="checkbox"/>
ICMP probe	<input checked="" type="checkbox"/>

The global probe parameters are described below:

probe interval	The target interval between TCP probes of a cluster that has been marked <i>failing</i> in the load balancing daemon's internal tables. If the server does not respond to strikeout threshold (see below) additional TCP probes after it is marked <i>failing</i> , then the server is marked down. These additional probes are at least probe interval seconds apart. This value is solely a target; the monitoring process adjusts itself based on a number of factors, including system load. The default value is 20 seconds.
probe timeout	The time in seconds that the probe daemon waits for a response from a server to a TCP or ACV probe. The default is 10 seconds.
strikeout threshold	The number of additional TCP probes sent to a server that is marked <i>failing</i> (see probe delay , below), and after which the server is marked <i>down</i> if no response is received. The default value is 2; must be between 1 and 6.
event interval	The number of seconds between evaluation of all Smart Events for all clusters. The default is 15 seconds. See "Configuring Smart Events" on page 171.
probe delay	The minimum time in seconds (default is 10) between successive TCP probes of servers by the probe daemon. If a server fails to respond to a probe, the probe daemon marks it <i>failing</i> in its internal server status table. You can override this value for each cluster. Specifying 0 to 5 seconds for probe delay means a 5-second delay (due to the fact that Equalizer's probe daemon goes through a probing cycle about every 5 seconds). Specifying 6 or more seconds increases the delay to at least that number of seconds, plus additional time due to load, latency, and other factors.

agent delay	<p>The minimum time in seconds (default is 10) between successive probes of server agents by the probe daemon.</p> <p>Specifying 0 to 5 seconds for agent delay means a 5-second delay (due to the fact that Equalizer's probe daemon goes through a probing cycle about every 5 seconds). Specifying 6 or more seconds increases the delay to at least that number of seconds, plus additional time due to load, latency, and other factors.</p>
require agent response	<p>Applies when a cluster uses either server agents or VLB agents. When you check this box, Equalizer will mark a server <i>down</i> when it receives no response from the appropriate agent. See Appendix A, "Server Agent Probes" and Appendix F, "Equalizer VLB".</p>
ICMP probe	<p>When enabled (the default), turns on ICMP echo request (ping) server probes. These probes are 5 seconds apart and this interval is not configurable. If a server does not respond to an ICMP probe, it is marked <i>down</i> only if the server has responded to at least one ICMP probe and been marked <i>up</i> by ICMP probing since the last reboot. This avoids marking a server <i>down</i> if it has been configured to not respond to ICMP echo requests.</p>

See "Server Health Check Probes and Timeouts" on page 283 for a complete description of Equalizer's server health checks and the global probe parameters.

Global Networking Parameters

Selecting **Equalizer > Clusters > Networking** displays the global cluster networking parameters. These settings apply to all clusters of the appropriate type (Layer 4 or Layer 7 or both) as indicated in the table:

networking parameters

send buffer	128
receive buffer	128
connect timeout	10.0
client timeout	5.0
server timeout	60.0
idle timeout	0.0
stale timeout	15.0
sticky netmask	off ▼
enable outbound NAT	<input type="checkbox"/>
passive FTP translation	<input checked="" type="checkbox"/>
ICMP drop redirects	<input type="checkbox"/>
ignore case	<input checked="" type="checkbox"/>
no outbound RST	<input type="checkbox"/>
abort server	<input type="checkbox"/>
allow extended chars	<input type="checkbox"/>
RST on server failure	<input type="checkbox"/>

commit
show defaults
reset

The global networking parameters are described below:

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. Default: 32. Minimum: 4. Maximum: 128. If this value is set for a cluster, the cluster value overrides the global value.
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. Default: 16. Minimum: 4. Maximum: 128. If this value is set for a cluster, the cluster value overrides the global value.
connect timeout	The time in seconds that Equalizer waits for a server to respond to a connection request. Layer 7 clusters only. See “HTTP and HTTPS Connection Timeouts” on page 278.
client timeout	The time in seconds that Equalizer waits before closing an idle client connection. Layer 7 clusters only. See “HTTP and HTTPS Connection Timeouts” on page 278.
server timeout	The time in seconds that Equalizer waits before closing an idle server connection. Layer 7 clusters only. See “HTTP and HTTPS Connection Timeouts” on page 278.
idle timeout	The time in seconds before reclaiming idle Layer 4 connection records. See “Layer 4 Connection Timeouts” on page 281.
stale timeout	The length of time that a partially open or closed Layer 4 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. See “Layer 4 Connection Timeouts” on page 281.
sticky netmask	<p>Enables sticky network aggregation for a subnet. Sticky network aggregation is applicable only for Layer 4 TCP and UDP clusters.</p> <p>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.)</p> <p>The sticky 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:</p> <ul style="list-style-type: none"> 8 bits corresponds to a Class A network 16 bits corresponds to a Class B network 24 bits corresponds to a Class C network <p>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.</p> <p>Note – If you are using two Equalizers in a failover configuration, you must set the sticky network aggregation mask identically on both Equalizers.</p>

<p>enable outbound NAT</p>	<p>When outbound NAT (Network Address Translation) is enabled, Equalizer modifies all packets <i>originating</i> from the servers behind it, substituting either the Default VLAN IP address or the address specified in the server's Outbound NAT tab for the source (server) IP address before forwarding the packets on to the recipient. [On the E250GX and legacy 'si' systems, the External Interface IP is used instead of the Default VLAN IP.] This option applies to both Layer 4 and Layer 7 and is disabled by default.</p> <p>See "Configuring Outbound NAT" on page 156 for a description of the server Outbound NAT tab.</p> <p>Outbound NAT is usually necessary when you are using reserved (i.e., non-routable) IP addresses for your servers on one VLAN and cluster IP addresses on another VLAN. NAT ensures that external hosts won't see packets originating from non-routable IP addresses. There is a performance cost for using NAT, since Equalizer must modify every outbound packet's source address.</p> <p>If your cluster and server IPs are all on the same subnet (sometimes called <i>single network mode</i>), outbound NAT should be <i>disabled</i>. Because the clusters and servers are all on the same subnet, NAT may interfere with other features (such as spoof). In this case, NAT should be configured on the network gateway device.</p>
<p>passive FTP translation</p>	<p>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.</p>
<p>ICMP drop redirects</p>	<p>Tells Equalizer to drop (i.e., ignore) incoming ICMP redirect messages.</p>
<p>ignore case</p>	<p>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, "Using Match Rules".</p>
<p>no outbound RST</p>	<p>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.</p>

<p>allow extended chars</p>	<p>By default, support for extended characters (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 chars flag.</p> <p>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 <i>before</i> you enable extended characters.</p>
<p>abort server</p>	<p>By default, when a client closes a connection, Equalizer waits for a response from the server before closing the server connection. If this flag is enabled, Equalizer will not wait for a response before closing the connection to the server; instead it sends a TCP RST (reset) to the server when the client closes the connection.</p>
<p>RST on server failure</p>	<p>Applies to Layer 4 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. The RST packet is sent when Equalizer removes the connection or when another packet using the same connection is received from the client, whichever happens first. By default, this option is disabled and Equalizer does not send RST packets to clients. Enabling this option is useful when load balancing an application that requires a TCP RST to close a connection; for example, Network File System (NFS).</p>

Setting Up a Failover Configuration

Two Equalizers can be configured into a hot backup, or *failover*, configuration that provides non-stop high availability web services. In such a configuration, one of the systems handles incoming requests and is called the *current primary* system, while the other (the *current 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 a *failover pair* in such a configuration.

General Failover Operation

When an Equalizer in failover is brought online, it checks to make sure that its configured network interfaces are link active by probing the default gateway, configured servers, and the failover peer on the appropriate VLANs. If no responses are received, Equalizer remains in the **initializing** state and does not assume any IP addresses until successful probes are received.

Once the network is active and the peer Equalizer can be contacted successfully, the failover peers begin a negotiation in which one system becomes the primary unit and the other becomes the backup unit. The primary Equalizer assumes all cluster and failover IP addresses, and begins serving incoming client requests. The other Equalizer assumes the backup role and continuously probes the current primary.

If the backup Equalizer cannot probe the current primary, it determines to identify whether the problem is with the backup's network connectivity or whether the primary unit has failed. If it determines the primary has failed, it will attempt to assume the current primary role: that is, if no other system has configured the cluster and failover IP addresses, the backup Equalizer assumes those IP addresses and starts handling traffic.

You must designate one Equalizer as the *preferred primary*, so that if both Equalizers attempt to assume primary role, the preferred primary Equalizer will assume primary role. The other unit will reboot and then assume the backup role.

Failover Issues with Spanning Tree

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. 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, and 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.

Failover Determination

Failover units constantly heartbeat one another over the VLAN IP address of every VLAN that has the **Use IP for Failover Heartbeat** flag enabled. Heartbeating fails if:

- there are at least 3 failed heartbeat probes on all VLANs and at least one failed heartbeat probe on each VLAN with the **Use IP for Failover Heartbeat** flag enabled
- OR
- there are at least 3 failed heartbeat probes on one VLAN with the **Use IP for Failover Heartbeat** flag enabled

Whenever the above conditions are met on either the current primary or the current backup, then Equalizer decides whether it should:

- remain in its current role (primary or backup),
- assume the primary role,
- or reboot.

This decision is based on the health of Equalizer’s network connectivity, whether the peer Equalizer can be contacted, and whether the cluster and failover IP aliases can be assumed. For a complete description, see the supplementary document *Understanding Failover Determination*, on the documentation website at docs.coyotepoint.com.

Failover Between Different Hardware & Software Releases

Since different Equalizer models and software revisions have varying configuration parameters, we recommend that you use the same model Equalizer hardware (e.g., E350GX, E450GX, etc.) and the same version of the EQ/OS software (e.g., 8.6.0a) for both systems in the failover pair. This is recommended because Equalizer by default maintains 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 other system. If different hardware or software is used on the failover systems, configuration transfers between the systems will need to be disabled.

Practically speaking, it is usually necessary to upgrade failover pairs to new hardware and software one at a time rather than upgrading both failover systems at the same time. 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 – Upgrading a failover pair of Equalizers currently running a version prior to Version 8.6 is described in a detailed upgrade document available at <http://docs.coyotepoint.com>.

Failover in Release 8.6 is supported between all hardware models that are supported for Version 8.6 installation: E250si, E350si, E450si, E550si, E650si, E250GX, E350GX, E450GX, and E650GX. A system running Version 8.6 can also be run in a failover configuration with an Equalizer running Version 8.5.1. The remainder of the failover configuration instructions in this chapter apply to all these situations, with the exception of the specific considerations listed in the following sub-sections.

Using Release 8.6 with Release 8.5.1, and GX or ‘si’ Hardware

When you are running Version 8.6 on an Equalizer GX or ‘si’ system in a failover configuration with an Equalizer running Version 8.5.1, the following exceptions apply to the instructions in this chapter:

- Version 8.5 of EQ/OS and ‘si’ model Equalizers support up to two untagged (port-based) VLANs. This limits the VLAN configuration on both failover Equalizers to two untagged VLANs. In Version 8.6 on GX models, the VLAN Wizard can help you automatically mirror the VLAN configuration supported on Version 8.5 and ‘si’ hardware. See “Initial VLAN Configuration Using the VLAN Wizard” on page 66.
- On the Equalizer running Version 8.6, the Failover Wizard cannot be used to setup failover. Failover must be configured using the **Failover** tabs in the Administrative Interface; see the section “Enabling Failover Using the Failover Tabs” on page 100.
- On the Equalizer running Version 8.6, entering a signature for the 8.5.1 peer is required (even though signatures are not supported on Version 8.5.1), or the Version 8.6 system may remain in standalone mode. To add the signature, log into the Version 8.6 system and open the **Equalizer > Failover** tab. Add any 48 character string (e.g.: **AA**) to the **Signature** text box for the **Peer Equalizer** and click **commit**.

- On the Equalizer running Version 8.6, the **use ssl only** flag on the **Failover > Synchronization** tab must be disabled (unchecked). Version 8.5.1 does not support this option.
- On the Equalizer running Version 8.6, the **dont transfer** flag on the **Failover > Synchronization** tab must be enabled (checked). Since the Equalizers are running different releases with different configuration file formats, we do not want configuration transfers to be performed between the peers. This means that changes made to the Equalizer configuration will need to be performed on both peers.

As stated above, the **Peer Signature** is not used when one peer is running 8.5 and one is running 8.6. If the peer running Version 8.5 is subsequently upgraded to Version 8.6, the newly upgraded system will remain in the **'initializing'** failover mode until you add each peer's signature to the other unit, as follows:

1. Log into the Equalizer Administration Interface on the newly upgraded Equalizer.
2. Select the *Failover Peer Name* at the top of the left frame.
3. In the **This Equalizer** field, copy the contents of the **Signature** text box using the mouse or keyboard.
4. Log into the Equalizer Administration Interface on the current primary Equalizer.
5. Select the *Failover Peer Name* at the top of the left frame.
6. In the **Peer Equalizer** field, paste the **Signature** you obtained in the previous step into the **Signature** text box. Click **commit**.
7. In the **This Equalizer** field, copy the contents of the Signature text box using the mouse or keyboard.
8. Go back to the newly upgraded Equalizer. In the **Peer Equalizer** field, paste the **Signature** you obtained in the previous step into the **Signature** text box. Click **commit**.

Using a GX and an 'si' in Failover with Version 8.6

Note the following when configuring a GX and an 'si' Equalizer (both running Version 8.6) in a failover pair:

- The GX Equalizer must have exactly the same VLAN configuration as the 'si' system. Note that 'si' systems support up to two untagged (port-based) VLANs.
- Both Equalizers must be configured using the **Failover** tabs as described in "Enabling Failover Using the Failover Tabs" on page 100. The Failover Wizard cannot be used.

Required VLAN Configuration for Failover in Release 8.6

In Version 8.6.0f, many of the restrictions on VLAN configuration for failover that existed in earlier versions have been modified or removed. The following sections describe the VLAN configuration requirements for all Release 8.6 versions.

Failover and VLAN Configuration in Version 8.6.0f and Later Versions

In order for failover to work between two systems running Version 8.6.0f (or later) software, the following requirements must be met by the VLAN configuration on both peers *before* failover can be configured successfully:


1. At least one VLAN must have both:
 - a **Failover IP** address and
 - have the **Use IP for Failover Heartbeat** option enabled.

A failover configuration requires an additional IP address on each VLAN, called the *failover IP address*. These IP addresses are assumed by the current primary system and are Equalizer's network-visible interfaces. When a failover occurs, the failover aliases are assumed by the backup system.

There is no restriction on how many VLANs you configure with a **Failover IP** address and **Use IP for Failover Heartbeat** enabled.

2. Any VLANs that host cluster IP addresses that are considered critical *must* have a **Failover IP** address defined and **Heartbeat** enabled. VLANs that host non-essential or test clusters need not have a Failover IP or have the **Use IP for Failover Heartbeat** flag enabled
3. If you define a **Failover IP** on a VLAN, you must also enable the **Use IP for Failover Heartbeat** flag on the VLAN. Failover may not work properly if a VLAN has a **Failover IP** and the **Use IP for Failover Heartbeat** flag is disabled.
4. All VLANs that have a **Failover IP** address defined and **Heartbeat** enabled *must* have identical settings on *both* failover peers for the following VLAN parameter settings:
 - **VID**
 - **VLAN IP**
 - **Netmask**
 - **Failover IP**
 - **Failover Netmask**
 - **Use IP for Failover Heartbeat**
 - **Permissions** (GUI access, etc.)
5. If you are using any Layer 4 TCP or UDP clusters with the **spoof** option *disabled*, then both peers *must* list VLANs in the same order on the **VLAN Configuration** screen. This is required so that the Layer 4 source address used for non-spoofing Layer 4 clusters is on the same VLAN on both systems. The Failover Wizard enforces this restriction when you configure failover. When configuring failover using the failover tabs, a VLAN ordering mismatch is not detected until *after* failover is configured, at which time a message is written to the log indicating that there is a VLAN ordering mismatch.

Do the following on *both* failover systems using separate browser windows to inspect and compare VLAN settings:

1. Click on the Equalizer system name in the left frame and then open the **Networking > VLAN Configuration** tab in the right frame.
2. Use the Modify button  in the **Actions** column of the table to examine the settings for each VLAN.
3. See the table on page 67 for a description of the **Modify VLAN** dialog parameters, and configure the VLANs appropriately.
4. If you make changes to a VLAN, be sure to click **commit** to update the configuration.

Failover and VLAN Configuration in Versions Prior to Version 8.6.0f

In Versions *prior to* Version 8.6.0f, VLAN failover configuration was more restrictive, in that all VLANs were required to have both a defined **Failover IP** address and the **Use IP for Failover Heartbeat** option enabled.

Therefore, it is necessary when configuring an Equalizer running Version 8.6.0f (or later) into failover with an earlier version, the VLAN configuration on Version 8.6.0f (or later) must meet these restrictions as well; that is, both systems must have a **Failover IP** defined and **Heartbeat** enabled on *all* VLANs.

Setting Up or Modifying Failover Using the Failover Wizard

The Failover Wizard allows you to setup failover on one machine and transfer the configuration over to the other. The only information that both machines need to know are each other's peer fingerprints. The wizard simplifies the failover setup process, so that most information is entered on one Equalizer and then transferred to the other.

Note – The Failover Wizard can only be used when configuring failover between two GX model Equalizers running Version 8.6. Failover between a GX and an 'si' Equalizer requires that you use the Failover tabs, as described in the section "Using a GX and an 'si' in Failover with Version 8.6" on page 97.

Launch the Failover Wizard from the preferred primary Equalizer. Before the Failover Wizard on the preferred primary can communicate with the other Equalizer and establish failover, you need to ensure that the VLAN configuration on both systems is the same, and copy the failover IP of the preferred primary to the preferred backup. Follow the procedure below:

1. Ensure that the VLAN configuration on both systems is the same; see the section "Required VLAN Configuration for Failover in Release 8.6" on page 97.
2. Copy the *fingerprint* of the Equalizer that will assume the *preferred primary* failover role to the *backup* system (the *preferred primary* is the system that will process traffic if both Equalizers reboot at the same time). To do this, do the following:
 - 2a. Log into the Equalizer Administration Interface on the failover peer that will assume the *preferred primary* role.
 - 2b. Select **Mode: Standalone** (or this Equalizer's *Failover Peer Name*) at the top of the left frame.
 - 2c. Open the **Info** tab in the right frame.
 - 2d. Copy the fingerprint displayed.
 - 2e. Log into the other failover peer (the *preferred backup*).
 - 2f. Right-click on **Mode: Standalone** (or this Equalizer's *Failover Peer Name*) at the top of the left frame, and select **Add Peer Signature** from the popup menu. The **Add Peer Signature** dialog appears.
 - 2g. Paste the fingerprint obtained in Step 2d, above, into the **Peer Signature** text box.
 - 2h. Click **commit**.
3. Check the sequence number on both systems. To do this click the **Help** icon on the top bar of the Administrative Interface and select **About** from the menu. Point at the **System Information** box to expand it and show the sequence number. The configuration file with the highest sequence number will be transferred to the other when failover is established. See the section "Updating the Configuration File Sequence Number" on page 330 if you need to edit the sequence number to preserve the configuration you require.
4. Log into the Equalizer Administration Interface on the failover peer that will assume the *preferred primary* role.
5. Right-click on **Mode: Standalone** (or this Equalizer's *Failover Peer Name*) at the top of the left frame, and select **Failover Wizard** from the popup menu.
6. If you have not previously configured failover on this system, go to the next step.

If you have previously configured failover, a popup window appears from which you must choose one of the following options:

Modify	Each screen of the wizard will be automatically filled with the current failover settings.
Clear	The wizard will clear the failover settings from the configuration file before beginning.
Cancel	Exit the Failover Wizard.

7. The **Prerequisites** screen of the failover wizard reminds you to perform the tasks listed in Steps 1 and 2 of this procedure before continuing:
 - 7a. If you have not yet copied this Equalizer’s peer signature to the other Equalizer, you can do so by copying the fingerprint value displayed and following the directions shown on this screen.
 - 7b. If you need to examine or change the VLAN configuration on this system, you’ll need to exit the wizard and go to the **VLAN Configuration** tab (click the **vlan config** button).

Once you have completed these prerequisites, click the **next** button to continue.

8. Follow the directions on the **Connect to Peer** screen to copy the peer signature of the other Equalizer into the text box displayed. Click the **next** button to continue.
9. On the **Set Peer Names and Primary Peer** screen:
 - 9a. Enter names for both peers (or accept the defaults provided).
 - 9b. Enable the **Preferred Primary** check box to make this Equalizer the preferred primary. If this check box is disabled, the other Equalizer will be made the preferred primary.

Click the **next** button to continue.

10. The **VLAN Settings** screen displays the current VLAN settings that will be used to establish failover with the other Equalizer. If there is some error in the settings displayed, you can click the **vlan config** button to go to the **VLAN Configuration** tab and exit the wizard. Otherwise, click **next** to continue.
11. On the **Configuration Transfer Settings** screen, select whether you want to transfer the configuration between systems when a change is made, and which elements of the configuration are transferred:

Enable transfers	Transfers the configuration file and any others items selected from the options below. If this option is disabled, then nothing is transferred between the failover peers.
SSL Certificates	Transfer all certificates associated with HTTPS clusters.
Envoy settings	Transfer the configuration for the optional Envoy GSLB product.
DNS	Transfer the Domain Name Service (DNS) settings.
Logging configuration	Transfer the remote logging configuration.

Click the **next** button to continue.

12. Click the **next** button on the **Start Failover Negotiation** screen to start the process of establishing failover between the peers.
13. The **Failover Negotiation Status** screen displays the progress of the negotiation between the peers. Once synchronization is complete and the failover process has been restarted on both systems, the **Status** for one peer should read **STABLE_PRIMARY**. and the other should read **STABLE_BACKUP**. Click **close** to exit the wizard.

Enabling Failover Using the Failover Tabs

The Failover Wizard allows you to setup failover on one machine and transfer the configuration over to the other. The only information that both machines need to know are each other’s peer fingerprints. This section shows you how to setup failover manually using the tabs in the Administrative Interface. This method requires that you supply all configuration information manually on *both* Equalizers.

Note – The **Add Peer Signature** command (used in the previous procedure to setup failover using the Wizard) does not populate the **Failover** tab with the peer’s signature; it is intended for use with the Failover Wizard only.

Manually Enabling Failover

1. Ensure that the VLAN configuration on both systems is the same; see the section “Required VLAN Configuration for Failover in Release 8.6” on page 97.
2. Check the **sequence** number on both systems. To do this click the **Help** icon on the top bar of the Administrative Interface and select **About** from the menu. Point at the **System Information** box to expand it and show the sequence number. The configuration file with the highest sequence number will be transferred to the other when failover is established. See the section “Updating the Configuration File Sequence Number” on page 330 if you need to edit the sequence number to preserve the configuration you require.
3. Log into the Equalizer Administration Interface on the failover peer that will assume the **preferred primary** role. Use a login that has **add/del** access on global parameters to initially define the configuration, or **write** access on global parameters to update an existing failover configuration. (Configuring the preferred primary Equalizer first ensures that it assumes the primary role.)
4. Select **Mode: Standalone** (or the *Failover Peer Name*) at the top of the left frame object tree, and then open the **Synchronization** tab in the right frame to set failover configuration transfer options:

dont transfer	By default, changes committed to the configuration on the primary system are transmitted to the backup system when the next heartbeat occurs. Enabling this flag tells Equalizer not to transfer configuration changes to the peer Equalizer. Used when the peer Equalizers are different hardware models or are running different software versions. This is normally necessary only during the process of upgrading a failover pair to a new software version, and you want to upgrade the peers at different times to maintain service. After both peers are upgraded to the new release, you can disable this flag on both peers.
Transfer envoy configuration	When enabled (the default), the current Envoy configuration is included when a configuration transfer to the failover peer occurs.
Transfer Certificates	When enabled (the default), the all HTTPS cluster and client security certificates stored on Equalizer are included when a configuration transfer to the failover peer occurs.
Transfer DNS Configuration	When enabled (the default), the current DNS configuration is included when a configuration transfer to the failover peer occurs.
Transfer Syslog Configuration	When enabled (the default), the current remote syslog configuration is included when a configuration transfer to the failover peer occurs.
Use SSL Only	By default, configuration transfers between failover peers are performed using non-secure connections. If this option is enabled, SSL is used for all configuration transfers. Note – When this option is <i>disabled</i> , responders are not transferred between failover peers -- they must be manually created on both units. When Use SSL Only is <i>enabled</i> , responders are synchronized properly between the failover peers.

If you make any changes, click **commit**.

5. Open the **Timing** tab to update the default failover timer settings, if necessary:

receive timeout	Time in seconds (default: 0.6) to wait for a heartbeat response from peer before timing out.
connection timeout	Time in seconds (default: 0.5) to wait for a connection attempt to the other peer to succeed before timing out.

probe interval	Time in seconds (default: 5.0) between successive heartbeat checks of the peer.
-----------------------	---

When either the **receive timeout** or the **connection timeout** occurs on the backup system, that counts as one “strikeout”, and the system attempts to check the heartbeat on the primary peer again. If three strikeouts occur in succession, the backup takes the primary role).

You should usually accept the default failover timing parameters, and only change them if there is a problem with heartbeat detection between the peers. For example, 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.

If you make any changes, click **commit**.

- Open the **Failover Peers** tab, which should look like this when failover has not yet been enabled:

Uncheck the **Disable Failover** check box. (See “Disabling the Failover Configuration” on page 105 for an explanation of this option.)

- The information below for **This Equalizer** is filled in automatically; only the **Equalizer Name** can be changed:

Equalizer Name:	A unique system name for this Equalizer. The default is “eq_” followed by the VLAN IP address of the Default VLAN interface.
Signature	The unique identifying signature for this Equalizer.

VLAN name	For each currently defined VLAN on this Equalizer, the VLAN IP for the VLAN is listed. If you need to make any changes to the VLANs defined, use the VLAN Configuration button at the bottom of the dialog.
Preferred Primary:	Indicates that this system should assume the primary role when both peers come up together. Check this box.

8. Enter the following information for the **Peer Equalizer**:

Equalizer Name:	A unique name for the failover peer you already configured. We suggest “ eq_ ” followed by the IP address of the Default VLAN on the peer Equalizer, but any name can be used.
Signature	The unique identifying signature for the peer Equalizer. To enter this value, do the following: <ol style="list-style-type: none"> 1. Log into the Equalizer Administration Interface on the other failover peer. 2. Select Mode: Standalone (or the <i>Failover Peer Name</i>) at the top of the left frame. 3. In the This Equalizer field, copy the contents of the Signature text box using the mouse or keyboard. 4. Go back to the Administration Interface on the Equalizer you are currently configuring and paste the Signature you obtained in the previous step into the Signature text box for the Peer Equalizer.
VLAN name	Fill in the VLAN IP for each VLAN listed. The list is populated with the currently defined VLANs on the Equalizer you are configuring. The VLANs defined on the two Equalizers must match exactly. If you need to make any changes to the VLANs defined, use the VLAN Configuration button at the bottom of the dialog.
Preferred Primary:	Indicates that this system should assume the primary role when both peers come up together. Do not enable this check box .

5. In the **Failover Parameters** field, the currently defined **Failover IP** addresses for all VLANs are displayed. These must match between both systems. If you need to make any changes to the VLANs defined, use the **VLAN Configuration** button at the bottom of the dialog.
6. Click **commit** to save your configuration and enable failover on this Equalizer (the preferred primary).
7. Log into the Equalizer Administration Interface on the failover peer that will assume the **preferred backup** role. Use a login that has **add/del** access on global parameters to initially define the configuration, or **write** access on global parameters to update an existing failover configuration.
8. Select **Mode: Standalone** (or the *Failover Peer Name*) at the top of the left frame object tree, and then open the **Synchronization** tab in the right frame to set failover configuration transfer options. Make sure that these match the settings you chose in Step 4 on page 101.
If you make any changes, click **commit**.
9. Open the **Timing** tab to update the default failover timer settings, if necessary. Make sure that these match the settings you chose in Step 5 on page 101.

If you make any changes, click **commit**.

- Open the **Failover Peers** tab. The **Disable Failover** check box should be unchecked. The information below for **This Equalizer** is filled in automatically; only the **Equalizer Name** can be changed:

Equalizer Name:	A unique system name for this Equalizer. The default is "eq_" followed by the VLAN IP address of the Default VLAN interface.
Signature	The unique identifying signature for this Equalizer.
VLAN name	For each currently defined VLAN on this Equalizer, the VLAN IP for the VLAN is listed.
Preferred Primary:	Indicates that this system should assume the primary role when both peers come up together. Check this box.

Checking the **Disable Failover** check box disables failover (and configuration transfers) between the two peer Equalizers. No information is deleted from the failover configuration. Leave this option disabled.

- The information below for **This Equalizer** is filled in automatically; only the **Equalizer Name** can be changed:

Equalizer Name:	A unique system name for this Equalizer. The default is "eq_" followed by the VLAN IP address of the Default VLAN interface.
Signature	The unique identifying signature for this Equalizer.
VLAN name	For each currently defined VLAN on this Equalizer, the VLAN IP for the VLAN is listed. If you need to make any changes to the VLANs defined, use the VLAN Configuration button at the bottom of the dialog.
Preferred Primary:	Indicates that this system should assume the primary role when both peers come up together. This box should be unchecked.

12. Enter the following information for the **Peer Equalizer**:

Equalizer Name:	A unique name for the other failover peer. We suggest “ eq_ ” followed by the IP address of the Default VLAN on the peer Equalizer, but any name can be used.
Signature	The unique identifying signature for the peer Equalizer. To enter this value, do the following: <ol style="list-style-type: none"> 1. Log into the Equalizer Administration Interface on the failover peer you configured earlier in this procedure. 2. Select Mode: Standalone (or the <i>Failover Peer Name</i>) at the top of the left frame. 3. In the This Equalizer field, copy the contents of the Signature text box using the mouse or keyboard. 4. Go back to the Administration Interface on the Equalizer you are currently configuring and paste the Signature you obtained in the previous step into the Signature text box for the Peer Equalizer.
VLAN name	Fill in the VLAN IP for each VLAN listed. The list is populated with the currently defined VLANs on the Equalizer you are configuring. The VLANs defined on the two Equalizers must match exactly.
Preferred Primary:	Enable this check box .

13. In the **Failover Parameters** field, the currently defined **Failover IP** addresses for all VLANs are displayed. These must match on both systems. If you need to make any changes to the VLANs defined, use the **VLAN Configuration** button at the bottom of the dialog.
14. Click **commit** to save your configuration and enable failover on this Equalizer (the preferred backup).
15. On both Equalizers, check the failover status:
- The icons at the top of the left frame tree should indicate that one of the Equalizers is in primary mode (the running coyote) while the other is in backup mode (the sitting coyote).
 - Click **Help > About** and then expand the **Equalizer System Information** box; the **failover mode** should indicate that one of the systems is in primary mode and the other in backup mode.
 - Messages indicating that failover has been established should appear in the Equalizer log. Click the Equalizer system name in the left frame and then open the **Status > Event Log** tab.

Modifying the Failover Configuration

Changes to settings on the **Failover > Synchronization** and **Failover > Timing** tabs can be made without disabling failover. Any changes made in the **Failover > Required** tab will also disable failover on commit. This is the equivalent of explicitly disabling failover as described in the following section.

Instead, we recommend that you use the failover wizard to change the configuration settings found on the **Required** tab, so that the changes are validated, communicated to the other system, and the two systems can renegotiate failover status. By using the failover wizard, you ensure that there will be minimal interruption of failover services, and no attempt by the backup system to assume the cluster IP addresses.

Disabling the Failover Configuration

It is sometimes desired to disable failover on Equalizer, without clearing the failover configuration. This is usually done to perform system maintenance or make significant configuration changes. Disabling failover releases all Failover IP addresses and transitions Equalizer into *standalone* mode. If you disable failover on the current backup,

it will immediately assume all cluster IPs in the configuration -- which will lead to IP address conflicts with the current primary, if it is still available. It is therefore advisable that if you are going to disable failover on one of your failover peers, you should shut down the other failover peer to avoid IP address conflicts.

Note that, starting with Version 8.6.0f, you can modify the VLAN configuration *without* disabling failover. See the special procedures found in Chapter 4, “Equalizer Network Configuration.” on page 63.

Caution – If you disable failover on one or both units in a failover pair, and plan to later re-enable failover between the two systems on which you have disabled failover, it is important that you only update the configuration file on one of the systems while failover is disabled. Updating the configuration will increase the sequence number of the configuration file. When failover is re-established, the system with the highest configuration file is transferred to the other Equalizer during the first synchronization between the systems. See the section, “Re-enabling Failover After Disabling” on page 106.

To temporarily disable the failover configuration without deleting the failover settings, do the following:

1. On one of the Equalizers, click on the other Equalizer’s peer name at the top of the left frame tree.
2. Check the **Disable Failover** check box in the right frame.
3. Click **commit**.
4. Perform Steps 1 through 3 on the other Equalizer.
5. Shut down one of the Equalizers. Open that Equalizer’s Administrative Interface, click on its peer name in the left frame tree, and open the **Maintenance** tab. Click on the **shutdown** button.

Re-enabling Failover After Disabling

If you are re-enabling failover between two Equalizers that previously were enabled in the same failover pair -- and failover was disabled as described in the section “Disabling the Failover Configuration”-- then it is possible that configuration file updates have been made on one or both of the units.

Each time a configuration change is made, the sequence number in the configuration file is updated. When failover is established, the configuration file with the highest sequence number is transferred to the other during the first synchronization between the units (unless the **dont transfer** option is enabled on the **Failover > Synchronization** tab). For this reason, check the sequence number on both systems before re-enabling failover between two running systems.

1. If both Equalizers are running, check the sequence number on both systems. To do this click the **Help** icon on the top bar of the Administrative Interface and select **About** from the menu. Point at the **System Information** box to expand it and show the sequence number.

The configuration file with the highest sequence number will be transferred to the other when failover is established. See the section “Updating the Configuration File Sequence Number” on page 330 if you need to edit the sequence number to preserve the configuration you require.

2. Do the following on the currently running Equalizer, or the Equalizer with the highest configuration file number:
 - 2a. Click on the other Equalizer’s peer name at the top of the left frame tree.
 - 2b. Uncheck the **Disable Failover** check box.
 - 2c. Click **commit**.
3. If one of the Equalizers in the failover pair was shut down when failover was disabled or cleared, power on that Equalizer now.
4. On the Equalizer you powered up in the previous step, or the Equalizer that had the lower configuration file number in Step 2:

- 4a. Click on the other Equalizer's peer name at the top of the left frame tree.
- 4b. Uncheck the **Disable Failover** check box.
- 4c. Click **commit**.

Both units should now indicate that they are in primary or backup mode, as described in Step 15 on page 105.

Clearing the Failover Configuration

Clearing the failover configuration removes the failover peer information for the other Equalizer from the configuration file. No other settings (including the VLAN configuration) are affected. To disable failover completely, you need to clear the configuration on *both* Equalizers.

Clearing the configuration causes Equalizer to go into standalone mode. As described under “Disabling the Failover Configuration” on page 105, you should shut down one of the Equalizers until you are ready to re-establish failover between the two units. Otherwise, they will both try to assume the IP addresses for all clusters. Also see the Caution in that section.

1. Right-click on **eq_IPaddress** (the peer name of the other Equalizer) at the top of the left frame, and select **Failover Wizard** from the popup menu. The **Failover Already Configured** screen opens.
2. Click the **Clear** button to clear the configuration.
3. Click next to reconfigure failover, or cancel to leave the wizard.
4. Perform Steps 1 through 3 on the peer Equalizer.

To re-establish failover between the two systems, see “Setting Up or Modifying Failover Using the Failover Wizard” on page 99 or “Enabling Failover Using the Failover Tabs” on page 100.

Changing from Multi-VLAN to Single-VLAN Configuration

Note that if you change the VLAN configuration of your Equalizer from using two or more VLANs to using a single VLAN, you should *clear* your failover configuration before doing so (see “Clearing the Failover Configuration” on page 107). If you do not, the failover configuration and the VLAN configuration may no longer match, and the (read-only) IP address information displayed in the **Failover** tabs may be incorrect. The workaround if you do not clear failover before changing to a single-VLAN configuration is to edit the configuration file as explained in the procedure below, reboot the system, and re-configure failover.

If you need help using the command line interface, contact Coyote Point Support (support@coyotepoint.com) and follow this procedure with the assistance of a member of the technical support team.

1. Log into the Equalizer via SSH using the *eqsupport* account (if enabled), or via the serial port using the *root* account. If you do not log in as *root*, enter the following command to switch to the *root* account after login:


```
# su
```
2. Copy the configuration file to */tmp*:


```
# cp /var/eq/eq.conf /tmp
```
3. Edit the file */tmp/eq.conf*:


```
# ee /tmp/eq.conf
```

[The **vi** editor may also be used.]
4. Remove the `interface` stanza from the file -- that is, remove all the text between the `interface` keyword at the top of the file and the curly brace that ends the interface stanza. An example `interface` stanza is shown below, in **bold**, with some detail removed (...). This section can be quite long, since it contains the VLAN configuration for both failover peers:

```
interface {
```

```

if_flags                = !disable;
sibling eq140 {
  switch sw01 {
    ...
  }
  intaddr                = "192.168.0.140";
  extaddr                = "172.16.0.140";
  sysid                  = "00:30:48:66:a9:66";
  fingerprint            = "E99041C0FD010BA6084E8C01042EC7563270A83DAC10008C";
  flags                  = preferred_primary;
}
sibling eq_172.16.0.230 {
  switch sw01 {
    ...
  }
  intaddr                = "192.168.0.230";
  extaddr                = "172.16.0.230";
  sysid                  = "00:30:48:d3:ee:b6";
  fingerprint            = "2698C0ED8E67FB2C3C48E489C88185F534EAF7B7AC1000E6";
  flags                  = !preferred_primary;
}
}
(remove everything above this line)
sequence                = 283;
strikeout_threshold     = 3;
sticky_netmask          = "255.255.255.255";
...

```

5. Save your changes to the file.

6. Enter the following two commands:

```

# cp /tmp/eq.conf /var/eq/eq.conf
# shadow /var/eq/eq.conf

```

7. Reboot Equalizer:

```

# shutdown -r now

```

After Equalizer comes back up, you can re-create your failover configuration.

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

NTP is a protocol designed to synchronize the clocks of computers over a network. NTP on Equalizer is compatible with servers running versions 1, 2, 3, or 4 of the NTP protocol. NTPv4 is described in RFC 5905; NTPv3 is described in RFC 1305.

On Equalizer, NTP is used primarily to time various operations, to ensure accurate timestamps on log entries (with respect to server and client log timing), and to allow for examination of the timing of log entries on two Equalizers in a failover configuration.

NTP on Equalizer works by polling an NTP server defined through the Administrative Interface. The time between polls of the NTP server is controlled by the **minpoll** and **maxpoll** NTP parameters, which default to 64 seconds (1 min 4 sec) and 1024 seconds (~17 mins), respectively. The behavior of NTP is to poll with a frequency starting at **minpoll** and then decrease polling frequency over time to **maxpoll**, as the accuracy of the local clock approaches the accuracy of the remote server clock. The time it takes for the polling delay to increase from **minpoll** to **maxpoll** will vary based on a number of factors, including the accuracy of the clocks on the client and server, network latency, and other timing factors.

NTP calculates when the local and remote system clocks are sufficiently in sync to begin increasing the polling delay towards **maxpoll**. When the accuracy of the two clocks is significantly different, or there is significant latency, for example, the two clocks may never be in sufficient agreement to increase the delay towards **maxpoll**. In this case, Equalizer will continue to sync approximately every 64 seconds. This behavior indicates that a different NTP server should be chosen.

We do NOT recommend changing the default **minpoll** and **maxpoll** delays in the NTP configuration file, in order to ensure an accurate system clock. NTP packets are very small and should not cause any problems with Equalizer or network operation, except as described in the following section.

NTP and Plotting

When you initially configure NTP, this may effectively disable plotting until NTP completes the initial synchronization of Equalizer's system clock with the NTP server -- which may take from several hours to several days. This is because plotting depends on accurate timestamps in the plot log. Since initially NTP is adjusting the time at frequent intervals, the timestamps in the plot log may become out of sync with the system clock, and so no plot data may be returned. Once NTP is no longer making adjustments to the system clock, plotting will function normally.

To manage system time on Equalizer, follow this procedure:

1. Log into the Administrative Interface using a login that has **add/del** access for global parameters (see “Logging In” on page 52).
2. Select **Equalizer > Maintenance > System Time**:

Figure 21 The System Time tab

3. To set the time zone, make a selection from the drop down box in the **timezone setting** section, and select the **commit** button in that section. To configure the system time or NTP, go to the next step.
4. You can set the system time manually or using a Network Time Protocol (NTP) server. Do one of the following:
 - 4a. Use the drop-down boxes at the top of the **date and time** field to manually set the date and time. Make sure the **enable NTP synchronization** check box is disabled.
 - 4b. 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 110 for help choosing an appropriate NTP server. The above is an example appropriate for locations in the United States.
5. Select the **commit** button in the **date and time** section to save your changes.

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:

Table 22:

Worldwide	pool.ntp.org
Asia	asia.pool.ntp.org
Europe	europa.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
```

General System Maintenance

The **Equalizer > Maintenance > General** tab contains buttons for the system maintenance tasks described in the following sections:

Creating a Backup Archive	106
Restoring a Backup Archive	107
Restoring a Backup Archive on an Equalizer in Standalone Mode	107
Restoring a Backup Archive on an Equalizer in a Failover Pair	108
Shutting Down Equalizer	110
Rebooting Equalizer	110
Creating a System Information Archive	110
Upgrading Equalizer Software	111

Creating a Backup Archive

A backup contains the following Equalizer configuration information:

- The Equalizer configuration file containing the cluster/server configurations that appear in the left frame of the administrative interface, the failover configuration, interface IP addresses, and GUI logins.
- The Envoy configuration file containing the GeoCluster and GeoSite information as shown in the left pane of the administrative interface. This file is saved whether Envoy is configured or not.
- System configuration files for authentication, network configuration, logging, remote access, etc.
- Equalizer licenses.
- SSL Certificates used by the GUI and by the failover subsystem. [Note: cluster SSL certificates are *not* backed up for security reasons.]

To create a backup archive of your current system configuration, follow the steps below.

This procedure includes saving a system information archive as well as a backup archive. While creating the system information archive is not required when taking a backup, it is highly recommended. The system information archive can be used to positively identify the source of the backup archive (model Equalizer, running software version, etc.) and also provide system state information that will help resolve any issues that occur after you restore a backup archive onto Equalizer.

1. Log into the Administrative Interface using a login that has **add/del** access for global parameters (see “Logging In” on page 52).
2. Select **Equalizer > Maintenance > General**. Then click the **backup** button. When prompted, specify the location and filename to use for the backup archive. The default backup archive name is of the form *hostname-mm.dd.yyyy-HH.MM.bkp*, where *hostname* is the Equalizer system name, *mm* is the month, *dd* is the day, *yyyy* is the year, *HH* is hours and *MM* is minutes. Click **OK** to save the backup archive.
3. Create a system information archive. On the **Equalizer > Maintenance > General** tab, click the **save state** button. Once Equalizer collects the information for the archive, a dialog box is displayed by your browser to open or save the archive. Save the archive to the same location to which you saved the backup archive you created above.

The default save state archive name is *eqcollect.tgz*; we recommend you use a unique file name that includes the name of the system from which the archive was taken and the date, as in: *eqcollect_system-name_dd-mm-yy.tgz*.

Restoring a Backup Archive

Please read this procedure entirely before beginning a restore to be sure you understand the restore process.

Equalizer's backup and restore feature is designed to create backups of your existing configuration, so that the same configuration can be restored in case of administrative error or the need to replace a unit after a hardware failure.

Backup archives are designed to be used to restore an Equalizer in two basic situations:

- Restoring a previous configuration to the same Equalizer on which the backup was taken. In this scenario, a backup archive was created on an Equalizer and you want to return the system to a previously saved state.
- Using an archive created on another Equalizer to bring a new or re-purposed Equalizer into service. In this scenario, you have taken an Equalizer out of service (perhaps because of hardware failure), and are replacing it with another Equalizer that is either new, or was previously used in some other configuration.

Note – The restore facility is *not* designed as a means of downgrading or upgrading Equalizer to a different software version -- the target Equalizer to be restored must already be running the same version number on which the backup was created. This is essential to preventing conflicts between the configuration settings and features supported in the currently running release and the release contained in the backup archive.

Restoring a backup archive onto an Equalizer returns the configuration to the settings contained in the archive. This includes the following:

- **Network configuration:** saved IP address information could cause conflicts on the network if the restored file comes from another Equalizer or if the IP addresses in the configuration file have been reused.
- **User names and passwords:** The login and password settings saved in the archive may not be the same as in the current configuration. If you cannot log in to Equalizer or the Administrative Interface, you may need to define a new GUI user and reset the eqsupport account password.
- **Administrative interface:** Restoring a backup archive restores the previous version of the browser based Administrative Interface, which may be incompatible with the interface used in the current version. You should clear your browser cache or restart your browser after the restore is complete so that you are sure to be using the correct version of the interface.
- **Equalizer failover, cluster, and server settings:** This includes failover IP addresses and settings, cluster and server IP addresses and ports, as well as match rules, responders, and smart events. After a restore is complete, you must thoroughly check the configuration for correctness to ensure the system is functioning properly.

Some issues that arise after a restore are the result of two common practices:

- Restoring two units from the same backup.
- Restoring a new unit using a backup taken from a unit already in service.

In both these situations, issues arise because of using the same network and failover configuration settings on two Equalizers. The procedures in this section are structured to minimize and recover from such misconfigurations.

There are two restore procedures below, one for an Equalizer in Standalone mode (i.e., failover is not configured) and the other for an Equalizer that will be part of a failover pair.

Restoring a Backup Archive on an Equalizer in Standalone Mode

To restore a saved configuration to an Equalizer that is in Standalone mode follow these steps.

1. Ensure that you have serial access to the Equalizer to be restored. A serial connection will be needed after the restore is completed if there are IP conflicts that prevent you from accessing the GUI.
2. Ensure that the backup archive you are using was obtained from the same model Equalizer as the one you are going to restore. For example, a backup obtained from an E350GX Equalizer should be restored onto an

E350GX model Equalizer *only*. This can be determined by examining the system information archive saved with the backup (see “Creating a Backup Archive” on page 112).

3. Ensure that the Equalizer you are going to restore is running the same *version number* as the release contained in the backup archive, with the same *version letter* or later. For example, if the backup archive was obtained from an Equalizer running Version 8.6.0c, then it *must* be restored onto a system running Version 8.6.0c or a later version of 8.6.0.

On the Equalizer to be restored, check the **Equalizer version** number on the **Help > About** screen. If necessary, upgrade or downgrade your Equalizer software to match the version on which the backup archive was taken. The version in the backup archive can be determined by checking the file *quickreport.txt* in the system information archive saved with the backup (see Step 3 on page 112).

4. Ensure that the Equalizer you are going to restore is properly licensed. Check the **Help > About** screen. If your Equalizer is unlicensed, the Equalizer picture will be blank and “unlicensed” appears underneath. To license Equalizer, see “Licensing Equalizer” on page 86.
5. On the Equalizer to be restored, do the following:
 - 5a. Select **Equalizer > Maintenance > General**. Click the **restore** button.
 - 5b. Click **Browse...** to locate and select the backup archive file that you want to restore onto this Equalizer.
 - 5c. Click **restore** to upload and install the backup archive. Equalizer asks for confirmation before rebooting to the new configuration.
6. While the system is rebooting, clear your browser cache (or restart your browser). After the system is available again, re-open the Equalizer GUI. Note the following:
 - Remember to use the Equalizer IP address, logins, and passwords from the backup archive.
 - If you have forgotten the logins and passwords used in the backup archive, you will need to add a new login using the CLI over the serial interface before you can continue. See “Adding Administrative Interface Logins” on page 44 and “Managing Remote Access to the Equalizer” on page 47.
7. Thoroughly check the entire restored configuration, verifying the system license, system ID, VLAN configuration, cluster IP addresses, server IP addresses, etc.

The IP addresses restored onto Equalizer may cause one or more IP conflicts on the network. If this is the case and you can still access the GUI, then use the GUI to adjust the VLAN configuration. If you cannot access the GUI, then you must reset the default VLAN (and delete all other VLANs) using the CLI over the serial interface and then re-configure all required VLANs. See “Resetting the Front-Panel Interface Ports” on page 79 and “Configuring VLANs on Equalizer” on page 65.

Restoring a Backup Archive on an Equalizer in a Failover Pair

In this procedure, you are replacing a failed unit in a failover pair. The new unit is being restored using a backup created previously on the unit that failed.

1. Ensure that you have serial access to the Equalizer to be restored. A serial connection will be needed after the restore is completed if there are IP conflicts that prevent you from accessing the GUI.
2. Ensure that the backup archive you are using was obtained from the same model Equalizer as the one you are going to restore. For example, a backup obtained from an E350GX Equalizer should be restored onto an E350GX model Equalizer *only*. This can be determined by examining the system information archive saved with the backup (see “Creating a Backup Archive” on page 112).
3. Ensure that the Equalizer you are going to restore is running the same *release number* as the release contained in the backup archive, with the same letter designator or later. On the Equalizer to be restored, check the **Equalizer version** number on the **Help > About** screen. If necessary, upgrade or downgrade your Equalizer software to match the version on which the backup archive was taken.

For example, if the backup archive was obtained from an Equalizer running Version 8.6.0c, then it *must* be restored onto a system running Version 8.6.0c or a later version of 8.6.0. The version in the backup archive can be determined by examining the system information archive saved with the backup (see “Creating a Backup Archive” on page 112).

4. Ensure that the Equalizer you are going to restore is properly licensed. Check the **Help > About** screen. If your Equalizer is unlicensed, the Equalizer picture will be blank and “unlicensed” appears underneath. To license Equalizer, see “Licensing Equalizer” on page 86.
5. Make sure that the failover configuration on *both* the Equalizer to be restored and the other Equalizer to be deployed in the failover pair are cleared (reset to defaults). Do the following on *both* Equalizers:
 - 5a. Left-click the top-most icon in the left-pane object tree to display the failover **Configuration** tab. If the parameters in the **Peer Equalizer** section are blank, then failover is already cleared on this unit.
 - 5b. Otherwise, to clear failover, right click on the top-most icon in the left-pane object tree and select **Failover Wizard**. Select the **clear** button to clear the failover configuration. Then, click **no** to exit the wizard.

The top-most object in the left frame tree on both units should now display: **Standalone**.

6. On the Equalizer to be restored:
 - 6a. Select **Equalizer > Maintenance > General**. Click the **restore** button.
 - 6b. Click **Browse...** to locate and select the backup archive file that you want to restore onto this Equalizer.
 - 6c. Click **restore** to upload and install the backup archive. Equalizer asks for confirmation before rebooting to the new configuration.
7. While the system is rebooting, clear your browser cache (or restart your browser). after the system is available again, re-open the Equalizer GUI. Note the following:
 - Remember to use the Equalizer IP address, logins, and passwords from the backup archive.
 - If you have forgotten the logins and passwords used in the backup archive, you will need to add a new login using the CLI over the serial interface before you can continue. See “Adding Administrative Interface Logins” on page 44, “Managing Remote Access to the Equalizer” on page 47, and “Managing Multiple Interface Users” on page 56 .
8. Thoroughly check the entire restored configuration, verifying the system license, system ID, VLAN configuration, cluster IP addresses, server IP addresses, etc.

The IP addresses restored onto Equalizer may cause one or more IP conflicts on the network. If this is the case and you can still access the GUI, then use the GUI to adjust the VLAN configuration. If you cannot access the GUI, then you must reset the default VLAN (and delete all other VLANs) using the CLI over the serial interface and then re-configure all required VLANs. See “Resetting the Front-Panel Interface Ports” on page 79 and “Configuring VLANs on Equalizer” on page 65.

9. Follow the instructions in Step 5 on page 115 to determine if the newly restored unit has a failover configuration and, if it does, clear it.
10. Check the sequence number on the **Help > About** screen of both units and ensure that the configuration that you want to keep has the *higher* sequence number. Edit the configuration file if necessary to increase the sequence number on the appropriate unit.
11. Configure the two Equalizers into failover. See “Setting Up a Failover Configuration” on page 95.
12. Check the logs on both units (click **Equalizer > Status > Event Log**). Ensure that one unit has entered ‘primary’ mode while the other is in ‘backup’ mode. If either unit remains in ‘initializing’ mode, there is most likely a configuration issue. Messages in the log should indicate the source of the problem. If necessary, you can contact Coyote Point Support to help resolve the issue.

Shutting Down Equalizer

Before turning off Equalizer or disconnecting the power, you should perform a clean shutdown. Once Equalizer shuts down, it must be power cycled to boot.

To shut down Equalizer cleanly, follow these steps:

1. Log into the Administrative Interface using a login that has **add/del** access for global parameters (see “Logging In” on page 52).
2. Select **Equalizer > Maintenance > General**. Then select the **shutdown** button.
3. In the confirmation dialog box, click **shutdown** to confirm that you really want to shut down Equalizer (or click **Cancel** to abort the shutdown request). If you click **shutdown**, 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 Administrative Interface using a login that has **add/del** access for global parameters (see “Logging In” on page 52).
2. Select **Equalizer > Maintenance > General**. Then select the **reboot** button.
3. In the confirmation dialog box, click **reboot** to confirm that you really want to reboot Equalizer. A progress dialog is displayed while the system shuts down and reboots. Once the progress screen closes, refresh the browser display.

Creating a System Information Archive

You can 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. (In earlier releases, creating this archive was performed by logging into Equalizer and executing the **eqcollect** command.)

To create the system information archive:

1. Log into the Administrative Interface using a login that has **read** access for global parameters (see “Logging In” on page 52).
2. Select the **Equalizer > Maintenance > General** tab.
3. Select the **save state** button to create the archive. Once Equalizer collects the information for the archive, a dialog box is displayed by your browser to open or save the archive. Save the archive to a file on your local hard disk and note its location.

The default archive name is *eqcollect.tgz*; we recommend you use a unique file name that includes the name of the system from which the archive was taken and the date, as in: *eqcollect_system-name_dd-mm-yy.tgz*. This ensures that you don’t overwrite an existing archive, and helps identify the archive to Coyote Point Support.

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.

Upgrading Equalizer Software

After you have finished connecting Equalizer to your network, you can use the Equalizer Configuration Utility to install the latest Equalizer software upgrade from Coyote Point. (You can also upgrade Equalizer software using the **eqadmin** command on the system console; please see “Upgrading Equalizer Software” on page 45.)

In order to upgrade:

- Equalizer must be licensed; see “Licensing Equalizer” on page 86 for more information.
- Equalizer must be able to access the Internet using FTP, or have access to a local FTP server that already has the upgrade image.

The procedure below contains the basic upgrade instructions for the current Equalizer software release. Please visit the **Coyote Point Documentation Website** at docs.coyotepoint.com for detailed upgrade instructions, release notes, and other useful release documentation.

1. Log into the Administrative Interface using a login that has **add/del** access for global parameters (see “Logging In” on page 52).
2. Select **Equalizer** in the left frame and open the **Maintenance** tab in the right frame.
3. Click on the **upgrade** button to start the upgrade wizard.
4. Do one of the following:
 - Click **Coyote Point FTP Server** to connect to the Coyote Point FTP server to download the upgrade image.
 - Click **User FTP Server** to connect to a local FTP server, to which you have already downloaded the upgrade image from the Coyote Point FTP server.
5. Enter the appropriate upgrade URL:
 - If you chose **Coyote Point FTP Server**: Enter the upgrade image URL provided to you by Coyote Point. The latest release of Equalizer software is always located at the following URL:

```
ftp://ftp.coyotepoint.com/pub/patches/upgrades/latest/upgrade.tgz
```
 - If you chose **User FTP Server**: Enter the upgrade image URL appropriate for your local FTP server, as provided by your local network administrator.

Click **commit** to start the download.
6. The wizard displays the progress of the download and prompts you to continue the upgrade once the download is complete. There may also be other prompts during the upgrade process, which may take as long as five minutes. When the upgrade is complete, the system must be rebooted to run the newly installed software. The upgrade wizard will display a progress window during the reboot and return to the **Maintenance** tab once the system is available again.



Working with Virtual Clusters	121
Adding a Layer 7 Virtual Cluster	122
Modifying a Layer 7 Virtual Cluster	122
Adding a Layer 4 Virtual Cluster	131
Modifying a Layer 4 Virtual Cluster	132
Deleting a Virtual Cluster	136
Copying an Existing Virtual Cluster	136
Configuring a Cluster's Load-Balancing Options	137
Equalizer's Load Balancing Policies	137
Configuring a Cluster to Use Server Agents	138
Enabling Persistent Server Connections	139
Enabling the Once Only and Persist Options	140
Enabling Once Only and No Header Rewrite for HTTPS	143
Enabling Once Only and Compression	144
Using Active Content Verification (ACV)	144
HTTPS Header Insertion	146
Specifying a Custom Header for HTTP/HTTPS Clusters	146
Performance Considerations for HTTPS Clusters	147
Providing FTP Services on a Virtual Cluster	148
Managing Servers	150
The Server Table	150
Server Software Configuration	151
Adding a Server to a Cluster	152
Modifying a Server	154
Configuring Outbound NAT	156
Adjusting a Server's Initial Weight	157
Setting Maximum Connections per Server	158
Interaction of Server Options and Connection Processing	160
Shutting Down a Server Gracefully	160
Deleting a Server	161
Automatic Cluster Responders	162
Managing Responders	162
Adding a Responder	162
Modifying a Responder	164
Plotting Responder Statistics	164
Using Regular Expressions in Redirect Responders	164
Using Responders in Match Rules	168
More Responder Examples	170
Responders and Hot Spares	170
Configuring Smart Events	171
Smart Events Components	171
Smart Event Trigger Expressions	171
Smart Event Action Functions and Variables	173
Smart Event Operators	175
Smart Event Configuration Parameters	175

Using IPMI to Power Servers On/Off	177
Complex Smart Event Expressions	177
Managing Smart Events	178
Using the Smart Event Expression Editor	179
Smart Event Examples	180
Configuring Direct Server Return (DSR)	188
Configuring Servers for Direct Server Return	190
Testing Virtual Cluster Configuration	193
Testing Your Basic Configuration	194

Working with 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 figure below shows a conceptual diagram of an Equalizer with three clusters.

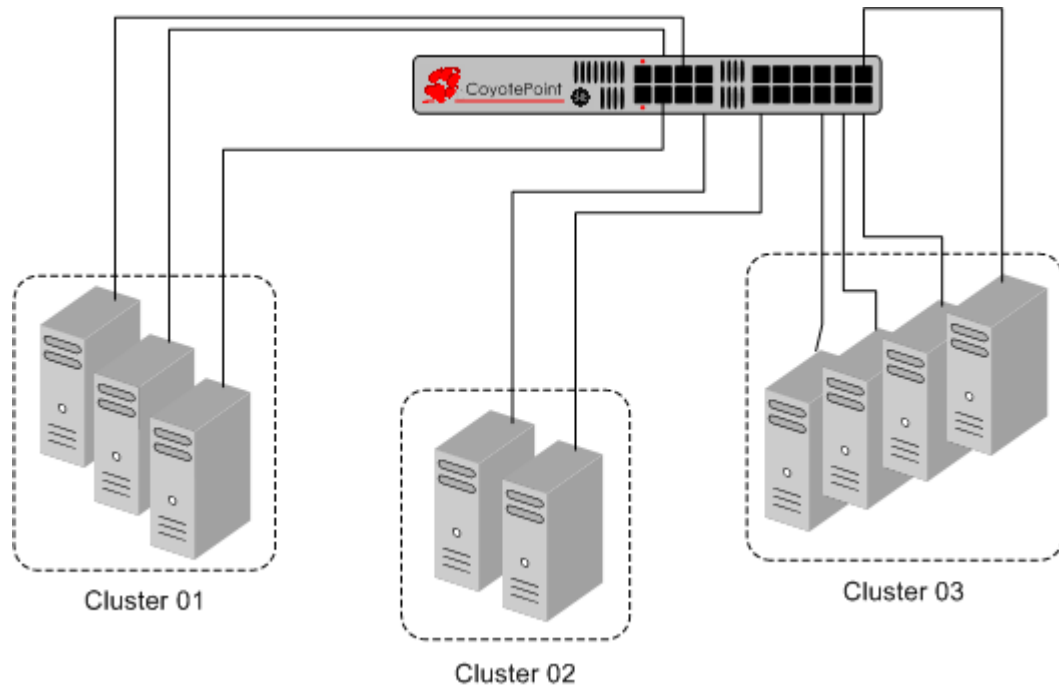


Figure 23 An Equalizer with three defined clusters

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 you read this chapter in its entirety so that you can:

1. Determine the **IP addresses** to use for each cluster, and for every server in each cluster.
2. Determine the **protocol** (Layer 4 TCP, Layer 4 UDP, Layer 7 HTTP, Layer 7 HTTPS) that will be used to communicate between the Equalizer and the servers in each cluster:
 - *In Layer 4 TCP and UDP clusters*, Equalizer routes requests based on configured load balancing criteria, the IP address, and the TCP or UDP port number. Load balancing decisions do not take into account the content of the request.

Any TCP-based protocol (HTTP, HTTPS, FTP, etc.) can be load balanced by an L4 TCP cluster.

L4 UDP clusters are appropriate for connectionless (stateless) applications, such as DNS, TFTP, Voice over IP (VoIP), and streaming applications -- any application that exchanges short packets with many clients, and where dropped packets are preferred to delayed packets (i.e., the highest possible network performance is required).

- *In Layer 7 HTTP and HTTPS clusters*, Equalizer routes requests to particular servers based on configured load balancing criteria, the IP address, the port, *and the content of the request*. Because Equalizer examines the request content, load balancing decisions can be made based on application specific criteria through the use of Match Rules.


Also note that in HTTPS clusters, Equalizer accepts HTTPS connections from clients, performs all the SSL operations necessary to examine the request, and sends the request on to a server in the cluster using HTTP.

This offloads resource-intensive SSL operations from the server to Equalizer, improving overall server and cluster performance.

3. 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.
4. Determine the additional settings and flags to be used on the cluster and its servers. For most options, start with the defaults and make incremental changes as you examine traffic passing through your clusters.

Adding a Layer 7 Virtual Cluster

To add a new virtual cluster, follow these steps:

1. Log into the Administrative Interface using a login that has **add/del** access for global parameters (see “Logging In” on page 52).
2. Right click on **Equalizer** (or the configure *Failover Peer Name* for this Equalizer) at the top of the left frame, and select **Add Cluster** from the menu that appears. The **Add New Cluster** dialog appears.
3. Select **Layer 7 HTTP** or **Layer 7 HTTPS** and then click the **Next** icon .
4. Enter the following information:

Cluster Name	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, <i>CPImages1</i>). The cluster name is limited to 63 characters.
Cluster IP Address	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.
Cluster Port	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 cluster port defaults to 80. For HTTPS clusters, the cluster port defaults to 443.

Click the **Next** icon .

5. A confirmation screen appears; click commit to create the cluster with the parameters shown.
6. The **Configuration** tab for the new cluster is opened. See the following section for an explanation of the Layer 7 cluster configuration tabs and parameters.

Modifying a Layer 7 Virtual Cluster

The configuration tabs for a cluster are displayed automatically when a cluster is added to the system, or by selecting the cluster name from the left frame Configuration Tree. HTTP and HTTPS clusters parameters are divided among the following tabs:

- **Layer 7 Required Tab**
- **Layer 7 Probes Tab**
- **Layer 7 Persistence Tab**
- **Layer 7 Networking Tab**
- **Layer 7 Security > Certificates Tab (HTTPS only)**
- **Layer 7 Security > SSL Tab (HTTPS only)**

These are described in the following sections. To update the settings on any tab, make changes and select the **commit** button to save them.

Layer 7 Required Tab

ip	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.
port	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).
netmask	Used to define an IP subnet that is different from the subnet defined for the cluster IP interface. If this is defined, it is assumed that the customer has the proper routing in place for clients to access multiple IP subnets defined on the Equalizer. The default is the netmask of the network interface for the cluster IP.
VLAN	The VLAN ID and name of the VLAN on which the cluster IP address resides. The VLAN is automatically selected by Equalizer when a cluster is created. It can be manually adjusted by choosing another VLAN from the drop down box.
disable	Disable this cluster. The cluster IP address will not accept requests when this flag is enabled.
ignore case	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.
spooof	<p>When the spooof option is enabled on a cluster (the default), Equalizer uses the client's IP address as the source IP address in all packets sent to a server in that cluster. All server responses to client requests that came through the Equalizer cluster IP address must be routed by the server back to the client through Equalizer. In many cases, the easiest way to do this is to set the default gateway on the servers in the cluster to Equalizer's IP address on the server VLAN. If this is not possible, you can establish static routes on the server to send responses to specific client IP addresses to Equalizer's IP address on the VLAN.</p> <p>If you disable spooof, the server receiving the request will see Equalizer's IP address as the client address because the TCP connection to the client is terminated when the request is routed. The server will therefore send its response back to Equalizer's IP address. Disabling the spooof option enables Source Network Address Translation (SNAT).</p>
compress	If Express Hardware GZIP Compression is installed, the compress flag appears in the HTTP and HTTPS cluster configuration screens. When the compress cluster flag is enabled, Equalizer automatically detects requests to the cluster from compression-capable browser clients and performs GZIP compression on all cluster responses sent to that client. This effectively enables compression for all clients using recent browser versions. Also see "Layer 7 Networking Tab" on page 128.

Layer 7 Probes Tab

probe port	<p>The port on the Equalizer to be used to for all TCP and ACV server health check probes for this cluster. The port specified here becomes the default probe port used when a new server is added to the cluster.</p> <p>By default, the probe port field is set to zero and the Equalizer uses the Layer 7 port field value for the probe port when a new server is created. To change the default behavior, set probe port to a specific port number.</p> <p>A probe port value can also be set on individual servers as well; see Adding a Server to a Cluster.</p> <p>(Note that the server agent port remains a separate cluster port that is used only for server agent communication; see below.)</p>
ACV probe	<p>The optional active content verification (ACV) probe string. This string, if specified, is sent to the server as part of a TCP probe when ACV is enabled. The server must respond with the ACV response string (see below) or it is marked down. The ACV probe is only required when the service running on the probe port requires an input string in order to return the ACV response string. The ACV probe is limited to 1024 characters, and must use only the printable ASCII characters (decimal 32 to 126). For more information, refer to "Using Active Content Verification (ACV)" on page 144.</p>
ACV response	<p>Specify an active content verification (ACV) response string to enable ACV. When ACV is enabled, Equalizer augments its high-level TCP probe mechanism by searching for the ACV response string in the first 1024 characters of the server's response to high-level TCP probes. If the ACV response string is not found, the server is marked down. An ACV probe (see above) can be specified if the service running on the probe port requires input in order to respond. For more information, refer to "Using Active Content Verification (ACV)" on page 144.</p>
probe delay	<p>The minimum number of seconds between TCP and ACV probes of the cluster's servers. Also see the global parameters probe interval, probe timeout, probe delay, and strikeout threshold under "Modifying Global Parameters" on page 89.</p>
server agent port	<p>The port used to contact server agents. The default port is 1510. See Appendix A, "Server Agent Probes" on page 273 for more information.</p>
agent probe	<p>An optional string that is sent to an agent when an agent probe occurs. See Appendix A, "Server Agent Probes" on page 273 for more information.</p>
agent type	<p>server agent -- Equalizer uses a server agent to 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. See Appendix A, "Server Agent Probes" on page 273 for more information about configuring server agents.</p> <p>VLB -- Equalizer uses the VMware Infrastructure Management API to retrieve real-time virtual server performance information from a VMware vCenter (Virtual Center) console or from a single ESX Server. Before selecting this option, see Appendix F, "Equalizer VLB" on page 311.</p> <p>none -- No server agent is used.</p>

Layer 7 Persistence Tab

Please see “Enabling Persistent Server Connections” on page 139 for a discussion of server persistence on Equalizer.

cookie age	The 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 deletes the cookie and any subsequent client requests will be handled by Equalizer’s load-balancing algorithms.
cookie scheme	<p>Specifies the format of the cookie to be used for the cluster as an integer between 0 and 2 (default is 2)</p> <p>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.</p> <p>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.</p> <p>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.</p>
cookie generation	A value added to cookies when the cookie scheme is 2. In order for cookies to be valid, the specified cookie generation must match the equivalent number embedded in the cookie. Conversely, if you need to invalidate old cookies, increment this number.
cookie domain	If a cookie domain is specified, then Equalizer will honor cookies in client requests only if the server’s host name is within the specified domain. For example, if the cookie domain is <code>coyotepoint.com</code> , then Equalizer will only present the cookie to servers in the <code>coyotepoint.com</code> domain (for example, <code>www.coyotepoint.com</code> or <code>my.coyotepoint.com</code>). Wildcards are not supported in the cookie domain.
cookie path	If a cookie path is specified, Equalizer honors cookies in a client requests only when the path component of the request URI has the same prefix as that of the specified cookie path . For example, if the cookie path is <code>/store/</code> , Equalizer presents the cookie to the server only if the request URI includes a path such as <code>/store/mypage.html</code> .
persist	When enabled (the default), Equalizer uses cookies to maintain a persistent session between a client and a particular server. 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 , above).
always	By default, Equalizer inserts a persistence cookie into a server response only if it finds a cookie from the server in the response. If always and persist are enabled, Equalizer includes a cookie in the response regardless of whether the server sent a cookie. Note: this option is listed as ‘ persist always ’ on match rule Configuration tabs.

LB Policy Tab

On this tab, choose a load balancing **policy** and **responsiveness** for the cluster:

cluster parameters

policy

responsiveness

delay weight

active connections weight

agent weight

policy	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 , server agent , and custom . For more information, refer to “Configuring a Cluster’s Load-Balancing Options” on page 137.
responsiveness	responsiveness sets the load-balancing response setting for this cluster. For more information, refer to “Configuring a Cluster’s Load-Balancing Options” on page 137.

The parameter slider bars are static for all policies other than **custom**, and their positions indicate the relative weight that the selected policy uses for each parameter. If **custom** is selected, you can adjust the load balancing policy parameters:

delay weight	The relative influence on the policy of the current response time between Equalizer and the server.
active connections weight	The relative influence on the policy of the number of active connections currently open to a server
agent weight	The relative influence on the policy of the return value of a server agent (if any) running on the servers in the cluster.
VM CPU	For servers that are associated with VMware Virtual Machines, the relative influence on the policy of the VM CPU usage status returned by VMware. Displayed only if VLB Advanced is licensed and VLB is enabled for this cluster (see “Equalizer VLB” on page 311).
VM RAM	For servers that are associated with VMware Virtual Machines, the relative influence on the policy of the VM RAM usage status returned by VMware. Displayed only if VLB Advanced is licensed and VLB is enabled for this cluster (see “Equalizer VLB” on page 311).

Layer 7 Networking Tab

The parameters in the **Networking** tab affect:

- the amount of memory Equalizer allocates for data buffers and HTTP headers
- the connections between clients and Equalizer
- the connections between Equalizer and the servers in virtual clusters

send buffer	The amount of memory in kilobytes reserved by each Layer 7 proxy process to store outgoing data before it is placed on the network interface. Default: 32. Minimum: 4. Maximum: 128. This global parameter applies to Layer 7 HTTP and HTTPS clusters only, and can also be set per cluster.
receive buffer	The amount of memory in kilobytes reserved by each Layer 7 proxy process to store data that has been received on a network interface before it is processed. Default: 16. Minimum: 4. Maximum: 128. This global parameter applies to Layer 7 HTTP and HTTPS clusters only, and can also be set per cluster.
request max	The maximum amount of memory in kilobytes reserved for HTTP request headers. Default: 32. Minimum: 4. Maximum: 64. This global parameter applies to Layer 7 HTTP and HTTPS clusters only.
response max	The maximum amount of memory in kilobytes reserved for HTTP response headers. Default: 32. Minimum: 4. Maximum: 64. This global parameter applies to Layer 7 HTTP and HTTPS clusters only.
custom header	A custom HTTP header that Equalizer inserts into all client requests before they are sent to the server. The format of the string is <i>text:text</i> . Also see the insert client IP flag, below, and the section “Specifying a Custom Header for HTTP/HTTPS Clusters” on page 146.
compress minimum	The minimum file size in bytes required for GZIP compression, if enabled (see the compress flag under “Layer 7 Required Tab” on page 124). Files smaller than the minimum specified are not compressed. Default: 1024 bytes.

compress mime-types	<p>Specifies the <i>mime-types</i> that will be compressed when the compress flag is enabled for the cluster (see “Layer 7 Required Tab” on page 124). The value of this parameter is a string (maximum length: 512 characters) with valid mime-type names separated by a colon (:). The default compress mime-types string specifies the following mime-types:</p> <pre> text/* application/msword application/postscript application/rtf application/x-csh application/x-javascript application/x-sh application/x-shar application/x-tar application/x-tcl application/xslt+xml audio/midi audio/32kadpcm audio/x-wav image/bmp image/tiff image/x-rgb </pre> <p>Lists of officially supported mime-types can be found at: http://www.iana.org/assignments/media-types/</p>
connect timeout	<p>The time in seconds that Equalizer waits for a server to respond to a connection request. The default is the global value. See “HTTP and HTTPS Connection Timeouts” on page 278.</p>
client timeout	<p>The time in seconds that Equalizer waits before closing an idle client connection. The default is the global value. See “HTTP and HTTPS Connection Timeouts” on page 278.</p>
server timeout	<p>The time in seconds that Equalizer waits before closing an idle server connection. The default is the global value. See “HTTP and HTTPS Connection Timeouts” on page 278.</p>
abort server	<p>By default, when a client closes a connection, Equalizer waits for a response from the server before closing the server connection. If this flag is enabled, Equalizer will not wait for a response before closing the connection to the server; instead it sends a TCP RST (reset) to the server when the client closes the connection. This option will typically reduce the number of server connections in the TIME_WAIT state, as shown by the netstat console command.</p>
once only	<p>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 off by default: meaning that Equalizer will parse the headers of every client request. See “Enabling the Once Only and Persist Options” on page 140.</p>
insert client IP	<p>When this flag is enabled, Equalizer inserts an X-forwarded-for: header with the client's IP address into all client requests before they are sent to the server. This flag is <i>disabled</i> by default for HTTP clusters and <i>enabled</i> by default for HTTPS clusters.</p>

Layer 7 Security > Certificates Tab (HTTPS only)

Use the **Security > Certificates** tab to:

- upload an SSL certificate that clients will use to validate a connection to an HTTPS cluster (a **cluster** certificate)
- upload an SSL certificate for Equalizer to use to validate clients that request connections to HTTPS clusters (a **client** certificate)

See “Using Certificates in HTTPS Clusters” on page 296 for more information.

Layer 7 Security > SSL Tab (HTTPS only)


The **Security > SSL** tab allows you to configure various options that are specific to HTTPS connections.

cipher suite	Lists the supported 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. Please see “Configuring Cipher Suites” on page 307.
session cache timeout	The number of seconds that Equalizer waits before disposing of an SSL session cache entry.
session cache kbytes	The maximum amount of memory in kilobytes allotted to an SSL session cache.
client verification depth	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 E, “Using Certificates in HTTPS Clusters” on page 295.
certify client	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 E, “Using Certificates in HTTPS Clusters” on page 295.
require certificate	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. See Appendix E, “Using Certificates in HTTPS Clusters” on page 295.
verify once	Indicates that the server will verify certificates only on the first client request, even if SSL is renegotiated. See Appendix E, “Using Certificates in HTTPS Clusters” on page 295.
ssl unclean shutdown	Should be enabled if you cannot access 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.

enable unsafe renegotiation	<p>SSL session renegotiation is disabled by default for HTTPS clusters to close the security vulnerability described at:</p> <p>http://web.nvd.nist.gov/view/vuln/detail?vulnId=CVE-2009-3355</p> <p>While there is usually no reason to use client-side renegotiation, it is typically used by some websites to allow different SSL certificates to be used for different parts of a website. Equalizer only supports this type of configuration when redirects are used. With redirects, renegotiation does not occur -- the client starts a new SSL session when redirected to a different part of the website that requires a new certificate.</p> <p>If the allow unsafe renegotiation option is enabled, all clients will be permitted to renegotiate SSL session IDs. <i>Enabling this option is not recommended by Coyote Point</i>, since it leaves your configuration open to session stealing and data injection.</p> <p>Note that if SSL processing is done in software (as on the E250GX and E350GX), then newer clients that contain the fix for CVE-2009-3355 will be able to renegotiate SSL sessions.</p>
no header rewrite	<p>When enabled, forces Equalizer to pass responses from an HTTPS cluster's servers without rewriting them. In the typical Equalizer setup, you configure 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 responses from the server so that they are HTTPS. You can direct Equalizer to pass responses from the server without rewriting them by enabling the no header rewrite flag.</p>

Adding a Layer 4 Virtual Cluster

To add a new Layer 4 virtual cluster, follow these steps:

1. Log into the Administrative Interface using a login that has **add/del** access for global parameters (see “Logging In” on page 52).
2. Right click on **Equalizer** (or the configure *Failover Peer Name* for this Equalizer) at the top of the left frame, and select **Add Cluster** from the menu that appears. The **Add New Cluster** dialog appears.
3. Select **Layer 4 TCP** or **Layer 4 UDP** and then click the **Next** icon .
4. Enter the following information:

Cluster Name	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, <i>CPIimages</i>).
Cluster IP Address	<p>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.</p> <p>Note that UDP clusters, unlike other cluster types, can use the same IP address/port combination as another TCP, HTTP, or HTTPS cluster. This is because UDP is a completely separate protocol from TCP (and the HTTP and HTTPS protocols built on top of TCP) and so does not conflict with TCP protocols running on the same IP address and port.</p>

Start Port End Port	<p>For L4 UDP and L4 TCP protocol clusters, a <i>port range</i> 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:</p> <ul style="list-style-type: none"> • Specific applications that require a range of ports. • The need to open up access to servers behind the Equalizer for all ports. <p>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.)</p> <p>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.</p> <p>The port defined for a <i>server</i> 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 “Adding a Server to a Cluster” on page 152.</p>
--------------------------------	---

Click the **Next** icon  .

5. A confirmation screen appears; click commit to create the cluster with the parameters shown.
6. The **Configuration** tab for the new cluster is opened. See the following section for an explanation of the Layer 4 cluster configuration tabs and parameters.

Modifying a Layer 4 Virtual Cluster

The configuration tabs for a cluster are displayed automatically when a cluster is added to the system, or by selecting the cluster name from the left frame Configuration Tree. TCP and UDP cluster parameters are divided among the following tabs:

- **Layer 4 Required Tab**
- **Layer 4 Probes Tab**
- **Layer 4 Persistence Tab**

These are described in the following sections. To update the settings on any tab, make changes and select the **commit** button to save them.

Layer 4 Required Tab

ip	<p>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.</p>
-----------	---

start port end port	<p>It is required to enter a non-zero value for the start port field, the port on which Equalizer will connect to the server. When the end port value is zero (the default), Equalizer uses the start port as the only server port on which to connect to the server.</p> <p>If a non-zero value for end port is entered, Equalizer will accept connections to the cluster IP address on any of the ports in the <i>port range</i> defined by the start port and end port fields. A port range allows Equalizer clusters to accept connections on multiple, contiguous ports. This is provided for applications that require multiple open ports. If end port is nonzero, the number specified must be greater than the number specified for start port.</p> <p>When a port range is defined on a cluster, the port defined for a <i>server</i> in the cluster is the start port of the server port range; it does not need to be the same as the start port of the cluster. See “Adding a Server to a Cluster” on page 152.</p>
idle timeout	<p>The time in seconds before reclaiming idle Layer 4 connection records. Applies to Layer 4 TCP clusters only. See “Layer 4 Connection Timeouts” on page 281 for a full description. If you enable direct server return (see below), you may also need to increase this value as explained in the section “Configuring Direct Server Return (DSR)” on page 188.</p>
disable	<p>When this check box is turned on, no incoming requests to the cluster IP address will be processed by Equalizer.</p>
spoof	<p>When the spoof option is enabled on a cluster (the default), Equalizer uses the client’s IP address as the source IP address in all packets sent to a server in that cluster. All server responses to client requests that came through the Equalizer cluster IP address must be routed by the server back to the client through Equalizer. In many cases, the easiest way to do this is to set the default gateway on the servers in the cluster to Equalizer’s IP address on the server VLAN. If this is not possible, you can establish static routes on the server to send responses to specific client IP addresses to Equalizer’s IP address on the VLAN.</p> <p>If you disable spoof, the server receiving the request will see Equalizer’s IP address as the client address because the TCP connection to the client is terminated when the request is routed. The server will therefore send its response back to Equalizer’s IP address. Disabling the spoof option enables Source Network Address Translation (SNAT) and also has the following effects on a Layer 4 cluster:</p> <ul style="list-style-type: none"> • A sticky record is maintained for each connection, regardless of whether sticky connections are enabled or not. See “Enabling Sticky Connections” on page 139.] • On GX model Equalizers, if there is more than one VLAN defined, the source IP address used in packets going to the servers when spoof is disabled is the Equalizer IP address on the first VLAN on which servers are configured (as indicated by the server icons on the VLAN Configuration tab in the GUI -- see “Configuring VLANs on Equalizer” on page 65). In other words, all Layer 4 clusters that have spoof disabled <i>must</i> be located on first VLAN on which servers are configured. <p>Note: On legacy ‘si’ model Equalizers, the source address selection behavior for Layer 4 clusters with spoof disabled is not changed: the source address used for outgoing packets to servers always uses the IP address of the Internal interface.</p>
direct server return	<p>When enabled, Equalizer forwards packets to the server in such a way that the server responds directly to the client, rather than through Equalizer. This option requires special configuration on the servers in the cluster; see “Configuring Direct Server Return (DSR)” on page 188 before enabling this option. The spoof option must also be enabled when DSR is enabled.</p>

Layer 4 Probes Tab

probe port	<p>The port on the Equalizer to be used to for all TCP and ACV server health check probes for this cluster. The port specified here becomes the default probe port used when a new server is added to the cluster.</p> <p>By default, the probe port field is set to zero and the Equalizer uses the Layer 4 start port value for the probe port when a new server is created. To change the default behavior, set probe port to a specific port number.</p> <p>A probe port value can also be set on individual servers as well; see Adding a Server to a Cluster.</p> <p>(Note that the server agent port remains a separate cluster port that is used only for server agent communication; see below.)</p>
ACV probe	<p>The optional active content verification (ACV) probe string. This string, if specified, is sent to the server as part of a TCP probe when ACV is enabled. The server must respond with the ACV response string (see below) or it is marked down. The ACV probe is only required when the service running on the probe port requires an input string in order to return the ACV response string. The ACV probe is limited to 99 characters, and must use only the printable ASCII characters (decimal 32 to 126). For more information, refer to "Using Active Content Verification (ACV)" on page 144.</p>
ACV response	<p>Specify an active content verification (ACV) response string to enable ACV. When ACV is enabled, Equalizer augments its high-level TCP probe mechanism by searching for the ACV response string in the first 1024 characters of the server's response to high-level TCP probes. If the ACV response string is not found, the server is marked down. An ACV probe (see above) can be specified if the service running on the probe port requires input in order to respond. For more information, refer to "Using Active Content Verification (ACV)" on page 144.</p>
probe delay	<p>The minimum number of seconds between TCP and ACV probes of the cluster's servers. Also see the global parameters probe interval, probe timeout, probe delay, and strikeout threshold under "Modifying Global Parameters" on page 89.</p>
server agent port	<p>The port used to contact server agents. The default port is 1510. See Appendix A, "Server Agent Probes" on page 273 for more information.</p>
agent probe	<p>An optional string that is sent to an agent when an agent probe occurs. See Appendix A, "Server Agent Probes" on page 273 for more information.</p>
probe ssl	<p>Equalizer uses SSL when it sends the ACV probe string. For more information, refer to "Using Active Content Verification (ACV)" on page 144. Note: Not supported for UDP clusters.</p>
agent type	<p>server agent -- Equalizer uses a server agent to 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. See Appendix A, "Server Agent Probes" on page 273 for more information about configuring server agents.</p> <p>VLB -- Equalizer uses the VMware Infrastructure Management API to retrieve real-time virtual server performance information from a VMware vCenter (Virtual Center) console or from a single ESX Server. Before selecting this option, see Appendix F, "Equalizer VLB" on page 311.</p> <p>none -- No server agent is used.</p>

Layer 4 Persistence Tab

sticky time	sticky time is the number of seconds that Equalizer should “remember” connections from clients. Valid values are from 0 (which disables sticky connections) to 1073741823 seconds (or over 34 years). For more information, refer to “Enabling Sticky Connections” on page 139.
inter-cluster sticky	inter-cluster sticky is a Layer 4 option that allows you to extend Layer 4 persistence across multiple server ports. For more information, refer to “Enabling Sticky Connections” on page 139.

LB Policy Tab

On this tab, choose a load balancing **policy** and **responsiveness** for the cluster:

policy	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 , server agent , and custom . For more information, refer to “Configuring a Cluster’s Load-Balancing Options” on page 137.
responsiveness	responsiveness sets the load-balancing response setting for this cluster. For more information, refer to “Configuring a Cluster’s Load-Balancing Options” on page 137.

The parameter slider bars are static for all policies other than **custom**, and their positions indicate the relative weight that the selected policy uses for each parameter. If custom is selected, you can adjust the following load balancing policy parameters::

delay weight	The relative influence on the policy of the current response time between Equalizer and the server.
active connections weight	The relative influence on the policy of the number of active connections currently open to a server
agent weight	The relative influence on the policy of the return value of a server agent (if any) running on the servers in the cluster.

Deleting a Virtual Cluster

Deleting a cluster with servers assigned to it also deletes the server definitions as well. To delete a cluster, follow these steps:

1. Log into the Administrative Interface using a login that has **add/del** access for global parameters (see “Logging In” on page 52).
2. In the left frame, right-click on the name of the cluster to be deleted and select **Delete Cluster** from the menu.
3. When prompted, click **delete** to confirm that you want to remove the cluster.

Copying an Existing Virtual Cluster

You can copy an existing cluster’s configuration to a new cluster by specifying a new name, type, IP address, and port. The cluster’s configuration is copied as follows:

- If the cluster type is the same for the new cluster and the cluster being copied, then all cluster parameter settings are also copied to the new cluster. Otherwise, default cluster parameter settings are used for the new cluster. For a review of cluster settings, see the following sections:
 - For HTTP and HTTPS clusters, see “Modifying a Layer 7 Virtual Cluster” on page 122.
 - For TCP and UDP clusters, see “Modifying a Layer 4 Virtual Cluster” on page 132.
 - All servers and server settings are copied to the new cluster. For a review of server settings, see:
 - “Modifying a Server” on page 154
1. Log into the Administrative Interface using a login that has **add/del** access for global parameters (see “Logging In” on page 52).
 2. In the left frame, right-click on Equalizer’s system name and select **Copy Cluster** from the popup menu. The **Copy Cluster** dialog appears:

3. Fill out the fields of the form as described in the table below:

copy cluster	Select the name of the cluster you want to copy from the drop-down box.
new cluster name	Type a unique name for the new cluster that begins with an alphabetical character (for example, <i>CPIimages1</i>) and is a maximum of 63 characters.
new cluster type	Select a cluster type for the new cluster. If the cluster type is the same for the new cluster and the cluster being copied, then all cluster parameter settings are also copied to the new cluster. Otherwise, default cluster parameter settings are used for the new cluster.

new cluster ip	Enter the ip address , which is the dotted decimal IP address of the cluster. The IP address of the cluster is the address (for example, 199.146.85.0) that clients use to connect to the cluster.
new cluster port	Enter the numeric port number on the Equalizer to be used for traffic between the clients and the cluster. For HTTP clusters, the cluster port is usually 80. For HTTPS clusters, the cluster port is usually 443.

4. Click **commit** to create the new cluster.

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 Initial Weight" on page 157.

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' initial 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 assigned initial weights. A server with a higher initial 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 (if any) 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. The fastest response policy also 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 the adaptive load balancing policy. For example, if a server's active connection count and server agent values are high,

Equalizer might not dispatch new requests to that server even if that server's response time is the fastest in the cluster.

- **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 running on all servers in the cluster. For more information about server agents, see "Configuring a Cluster to Use Server Agents" on page 138.

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 initial 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 initial 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.

Aggressive Load Balancing

After you fine-tune the initial 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 Initial Weight" on page 157.

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.

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 **require agent response** is set for the cluster, in which case Equalizer regards that server as down).

1. Log into the Administrative Interface using a login that has **add/del** access for the cluster (see “Logging In” on page 52).
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 the **Probes** tab 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.
6. If your agent needs to have a string sent to it before it will respond, 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 “Server Agent Probes” on page 273.

Enabling Persistent Server Connections

Equalizer provides several methods by which connections 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 server connections are enabled using the **sticky time** cluster parameter and (optionally) the **inter-cluster sticky** cluster flag. See “Enabling Sticky Connections” on page 139.

For Layer 7 clusters, persistent server connections are enabled using the **persist** and **always** cluster flags. See “Enabling Cookies for Persistent Connections” on page 140.

Enabling Sticky Connections

For Layer 4 TCP and UDP clusters, 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.

Sticky connections are managed on Equalizer using *sticky records* that record the IP address, port and other information for the client-server connection. When you enable sticky connections, the memory and CPU overhead for a connection increase. This overhead increases as the sticky time period increases.

Consequently, 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.

[Also see the description of the **sticky netmask** global parameter in the section “Global Networking Parameters” on page 91.]

With the **inter-cluster sticky** option, you can configure Equalizer to direct requests from a client to the same server on any available port that has a current persistent connection *in any cluster*.

When Equalizer receives a client request for a Layer 4 cluster with inter-cluster sticky enabled and the client does not have a sticky record for the cluster, then Equalizer will check other clusters that have inter-cluster sticky enabled for a sticky record for the same client and server -- but on a different server port than the one originally used in the client request.

If such a sticky record is found and the server IP/port in the sticky record is configured as a server in the current cluster, then the sticky record is used to send the client request to that server IP/port. Otherwise, the client request is load balanced across the available servers in the cluster.

In order for the inter-cluster sticky option to work:

- The two clusters must have the same cluster IP address and different ports.
- At least one server in each of the two clusters must be configured with the same IP address and different ports.

Inter-cluster stickiness is provided for the case where you have similar services running on the same server IP on two or more ports. Using *port ranges* for a cluster achieves essentially the same effect, without using another cluster IP address (see “Layer 4 Required Tab” on page 132). Using **inter-cluster sticky** is preferable in situations where you’d like the service available on multiple cluster IPs as well as multiple ports.

To enable sticky connections for a cluster, follow these steps:

1. Log into the Administrative Interface using a login that has **add/del** access for the cluster (see “Logging In” on page 52).
2. In the left frame, click the name of the Layer 4 TCP or UDP cluster to be configured. The cluster’s parameters appear in the right frame.
3. Select the **Persistence** tab in the right frame.
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. 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 Connections

For Layer 7 HTTP and HTTPS clusters, 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 inserts 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 “Modifying a Layer 7 Virtual Cluster” on page 122.

Enabling the Once Only and Persist Options

Since HTTP 1.0, web browsers and servers have been able to negotiate persistent connections over which multiple HTTP transactions could take place. 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.

There is a lot of overhead associated with opening and closing the TCP connections for each image. The way HTTP 1.0 optimizes this is to allow 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.

For HTTP/1.1, persistent connections are the default.

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 load balanced, but sent to the same server as the first request.

Enabling **once only** can be incompatible with persistence and Layer 7 HTTPS clusters (which rewrite HTTP to HTTPS links in server response headers), since in these cases we generally want to examine every request in a connection. However, in configurations where examining the headers in every transaction in a connection is not required, enabling **once only** can significantly improve performance.

Whether **once only** is enabled or not has a significant effect on how Equalizer routes requests, as summarized in the following table:

Requests in a single keep-alive connection	once only enabled	once only disabled
First Request		
persist enabled	<p>If request contains a cookie and there is no match rule hit, send request to the server in the cookie.</p> <p>If request contains a cookie and there is a match rule hit, send the request to the server in the cookie <i>only if it is in the list of servers selected in the match rule definition</i>. Otherwise, ignore the cookie.</p> <p>If there is no cookie, load balance the request and send to the server chosen.</p>	<p>If request contains a cookie and there is no match rule hit, send request to the server in the cookie.</p> <p>If request contains a cookie and there is a match rule hit, send the request to the server in the cookie <i>only if it is in the list of servers selected in the match rule definition</i>. Otherwise, ignore the cookie.</p> <p>If there is no cookie, load balance the request and send to the server chosen.</p>
persist disabled	Load balance the request and send to the server chosen.	Load balance the request and send to the server chosen.
match rule hit	Send to the server chosen by the match rule.	Send to the server chosen by the match rule.
Subsequent Requests		
persist enabled	Send to same server as <i>first</i> request (any cookie in request is ignored).	<p>If request contains a cookie, send request to the server in the cookie.</p> <p>If there is no cookie, load balance request and send to server chosen by policy.</p>
persist disabled	Send to same server as <i>first</i> request.	Load balance the request and send to the server chosen.
match rule hit	Send to same server as <i>first</i> request.	Send to the server chosen by the match rule.

For example, let's look at how Equalizer processes HTTPS requests. For an HTTPS cluster, Equalizer offloads SSL processing from the servers in the cluster; that is, Equalizer does all the SSL related processing itself, and then forwards the request 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 146). 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. Most servers that support SSL offloading require that every request contain the special headers -- therefore, in most cases like this you need to disable the **once only** flag for the cluster if you want to be able to parse for these headers in every request on the server end.

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 load balancing decisions need to be made for every request or where any of the above effects are undesirable, **once only** should be disabled.

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.

Enabling Both the Once Only and Always Options

The **always** flag influences when Equalizer inserts cookies into server responses; it in turn is affected by the setting of the **once only** flag, as shown in the following table:

	once only enabled	once only disabled
always enabled	<p>Equalizer always inserts a cookie into the <i>first</i> set of response headers on a connection <i>only</i>. The cookie is inserted regardless of whether the server included one in the response.</p> <p>Subsequent responses on the same connection are forwarded to the client <i>unchanged</i> by Equalizer.</p>	<p>Equalizer inserts its own cookie into <i>all</i> server responses on a connection. The cookie is inserted regardless of whether the server included one in the response.</p>
always disabled	<p>If the <i>first</i> server response on a connection already has a server cookie in it, Equalizer inserts its own cookie into the <i>first</i> set of response headers on the connection. If the response has no cookie in it, Equalizer does <i>not</i> insert one of its own.</p> <p>Subsequent responses on the same connection are forwarded to the client <i>unchanged</i> by Equalizer.</p>	<p>If the <i>first</i> server response on a connection already has a server cookie in it, Equalizer inserts its own cookie into the <i>first</i> set of response headers on the connection.</p> <p>Equalizer will insert a cookie into subsequent responses on the same connection if:</p> <ul style="list-style-type: none"> • they do not contain a valid cookie • the cookie generation has changed • the server in the cookie has the quiesce flag enabled

Note that the cluster parameters **cookie path**, **cookie age**, **cookie generation**, and **cookie domain** specify cookie content for the cluster (see “Layer 7 Persistence Tab” on page 126). If any of these parameters are updated, this changes the information used in the cookies that Equalizer inserts into server responses.

Enabling Once Only and No Header Rewrite for HTTPS

In a Layer 7 HTTPS cluster, clients connect to the cluster IP using HTTPS connections. Equalizer terminates the HTTPS connection and communicates with the servers in the cluster using the HTTP protocol. By default, Equalizer examines server responses for `http://` URLs and rewrites them as `https://` URLs, so that these URLs work

properly on the client. If, for example, a server sends an HTTP redirect using the `Location:` header, this URL most likely will include the `http://` protocol. Equalizer rewrites this response so that the URL uses `https://`.

For server connections that contain multiple server responses, the setting of the **once only** flag determines whether `Location:` headers in all server responses are rewritten. This is shown in the table below.

Note that the Administrative Interface does not permit you to *enable once only* and *disable no header rewrite* -- this option combination would rewrite the `Location:` header in only the first response in the connection, and not rewrite the headers in subsequent responses in the same connection. Doing so would produce errors on the client.

Of course, you can also direct Equalizer to pass responses from the server *without* rewriting URLs by enabling the **no header rewrite** flag on the cluster.

	once only enabled	once only disabled
no header rewrite disabled	<i>Not supported.</i>	The <code>Location:</code> headers of <i>every</i> response in a connection are rewritten.
no header rewrite enabled	No headers are rewritten.	No headers are rewritten.

The compress option is not available on E250, E350, or E450 model Equalizers

Enabling Once Only and Compression

Enabling both the **once only** and **compress** options is not allowed by the Administrative Interface. These two options are not compatible, since setting them both would mean that only the first response in a connection would be compressed and not the remainder of the responses, which would likely cause client errors.

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. Note, however, that you cannot use ACV with Layer 4 UDP clusters.

Using ACV

ACV checking is performed as part of the high-level TCP probes that Equalizer sends to every server by default. To enable ACV, you specify an **ACV response** string for a cluster. Equalizer will then search for the **ACV response** string in the first 1024 characters of the server's response to the high-level TCP probes. If the ACV response string is not found, the server is marked down. An ACV probe (see above) can be specified if the service running on the server's **probe port** requires input in order to respond.

How ACV works is best explained using a simple example. The HTTP protocol enables you to establish a connection to a server, request a file, and read the result. Figure 24 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.

```

Figure 24 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.

Specifying an *ACV probe string* and an *ACV response string* basically automates the above exchange. 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 24, 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.

For more information on probing, see "Server Health Check Probes and Timeouts" on page 283.

Enabling ACV

To enable ACV in an HTTP, HTTPS, or TCP cluster, follow these steps:

1. Log into the Administrative Interface using a login that has **add/del** access for the cluster (see "Logging In" on page 52).
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 the **Probes** tab in the right frame.
4. In the **ACV probe** field, specify the string Equalizer will send to the server's probe port; this string should cause the application on the server's probe port to respond with a string that contains the ACV response. Many protocols require a string to be sent to the server before a response is received. Some protocols, such as SSH, do not require a probe string; for such protocols, the ACV probe can be left blank.

5. 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.

6. In the **ACV response** field, specify a case-sensitive string. An **ACV response** string must be supplied or ACV probing will not be enabled. 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).
7. 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: (client'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 HTTP/HTTPS Clusters

Some applications require specific headers in incoming client requests, and Equalizer provides the custom header field in HTTP and HTTPS clusters to allow you to inject a custom header into the client request before it is sent to a server behind Equalizer.

An example is the Exchange 2003 version of Microsoft Outlook Web Access (OWA). OWA 2003 normally requires that all incoming client requests use the Secure Sockets Layer (SSL) protocol. This means that all client requests must have the `https://` protocol in the URI. If, however, OWA is running on a server in an Equalizer Layer 7 HTTPS 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 2003 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 2003 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)

The following procedure shows you how to add a custom header to an existing HTTPS cluster definition, using the header required for an OWA 2003 server as an example.

1. Log into the Administrative Interface using a login that has **add/del** access for the cluster (see “Logging In” on page 52).
2. In the left frame, click the name of the cluster to be configured.
3. In the right frame, select the **Networking** tab.
4. Type the following in the **custom header** field:


```
Front-End-Https: on
```
5. Select **commit** to modify the cluster.

Performance Considerations for HTTPS Clusters

Layer 7 HTTPS clusters have several options that can have a significant impact on the performance and behavior of the cluster:

- The injection of a **custom header** to provide transaction-specific information to the server. For example, to tell the server that Equalizer terminated the HTTPS connection and performed SSL processing on the incoming request (see the previous section, above).
- The translation of HTTP redirects to HTTPS redirects (see the description of the **no header rewrite** flag under “Modifying a Layer 7 Virtual Cluster.”).
- 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. In order to inject custom headers and rewrite headers in every transaction in a connection, turning off **once only** is required.

HTTPS Performance and Xcel SSL Acceleration

The E650GX and E450GX include the Xcel SSL Accelerator Card. Equalizer models without Xcel (E250GX and E350GX) perform all SSL processing in software using the system CPU. Equalizers with Xcel perform all SSL processing using the dedicated processor on the Xcel card. This allows the system CPU to concentrate on non-SSL traffic. For most applications, Xcel will process several hundred HTTPS transactions per second with no noticeable degradation in performance either for the HTTPS cluster or for Equalizer as a whole.

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 accelerator, not a bulk data encryption device. 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 the Xcel I and Xcel II cards do 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 Xcel I card on older 'si' models also does not support "AES" ciphers.

The default configuration for HTTPS clusters on Equalizers with an Xcel card will not include ciphers that are unsupported by the Xcel card, as described above. If, however, the cluster's **cipher suite** string is modified to include them, it is possible that they may be negotiated with clients. This will not lead to incorrect operation of the system, but encryption for these cipher suites will occur in software instead of taking advantage of the improved performance provided by the Xcel hardware.

Providing FTP Services on a Virtual Cluster

The FTP protocol dates from the 1970s, and was designed to be used in an environment where:

- the network topology is simple
- the FTP server and client communicate directly with one another
- the addresses used by the client and server for active FTP data connections can be negotiated over the FTP control connection
- the FTP server is able to make connections back to the FTP client

These operational characteristics of FTP require special configuration for load balancers (as well as firewalls and NAT devices) that pass traffic between FTP servers and FTP clients:

- NAT devices and routers (including load balancers like Equalizer) on the client and server sides must be configured to monitor FTP transactions and provide appropriate address translation and packet rewriting.
- Firewalls on the client and server sides must be configured to let traffic on the ports used for FTP through the firewall.

Consult the documentation for the firewalls and NAT devices used at your site to determine how to set up those devices appropriately for FTP transfers. See the next section for how to configure an Equalizer cluster for responding to FTP requests from clients.

FTP Cluster Configuration

When configuring an FTP cluster on Equalizer, the following guidelines must be followed:

1. The **protocol** for the cluster must be **Layer 4 TCP**.
2. The **start port** parameter for the cluster must be set to port **21**. (Note that port 20 is also used, but you do not specify it when adding the cluster.)
3. The **spoof** flag must be enabled for the cluster.
4. If your servers are on a network the outside world cannot reach, consider enabling Equalizer's **passive FTP translation** global flag. 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. Note that if you select this option, clients will only be able to connect to the cluster in passive (PASV) mode.

Also observe the following notes and limitations:

- Port redirection cannot be used with an FTP cluster; that is, the port range defined for the cluster and the port ranges defined for the servers in the cluster must be identical.
- Defining a port range that includes but does not start at port 21 does *not* define an FTP cluster. The port range *must* begin at port 21. In other words, specifying a **start_port** of 19 and an **end_port** of 50 does *not* define an FTP cluster; Equalizer will assume that services other than FTP will be running on these ports.
- FTP data connections are automatically configured (internally) with a **sticky time** of one second. This is necessary to support the passive mode FTP data connection that most web browsers use. This means that

there will be one sticky record kept for each FTP data connection. For an explanation of sticky records, see “Enabling Sticky Connections” on page 139.

- FTP clusters occupy two internal virtual cluster slots, even though only one appears in the interface. This permits Equalizer's NAT subsystem to rewrite server-originated FTP data connections as they are forwarded to the external network.
- You cannot enable the **direct server return** option on an FTP cluster.

Managing Servers

The following sections discuss viewing, adding, and deleting servers, as well as server configuration options:

- The Server Table
- Server Software Configuration
- Adding a Server to a Cluster
- Modifying a Server
- Configuring Outbound NAT
- Adjusting a Server's Initial Weight
- Setting Maximum Connections per Server
- Shutting Down a Server Gracefully
- Deleting a Server

The Server Table










Every cluster has a **Servers** tab that lists all of the currently defined servers in the cluster, and provides basic configuration and status information for each server. To display the server table for a cluster, click on the cluster name in the left frame and then click on the **Servers** tab in the right frame:

Server Status: up/down quiesced/disabled hot-spare

Name	IP Address	Port	Status	Actions
sw00	10.0.0.19	80		
sw01	10.0.0.120	80		
sw02	10.0.0.122	80		
sw03	10.0.0.102	80		
sw04	10.0.0.121	80		

Figure 25 The server table

Name	The server's name.
IP Address	The server's IP address.
Port	The server port.

Status	Status indicators for each server in the cluster:	
		The server is responding to probes and is ready to receive traffic.
		The server is not responding to probes and no traffic is being routed to it.
		The server is responding to probes, and is either disabled (the server's initial weight is set to 0) or the quiesce option is enabled.
		The server is not responding to probes, and is either disabled (the server's initial weight is set to 0) or the quiesce option is enabled.
		The server's dont probe option is enabled, so the probing status is unknown.
		The server's hot spare option is enabled. This icon appears after one of the above icons.
Actions		Delete the server from the cluster.
		Display and modify the server's configuration and option settings.
		Add a new server to the cluster.

Server Software Configuration

Please observe the following guidelines and restrictions when configuring the software 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. In most cases, this means that each server in a cluster should be configured to use Equalizer as its default gateway, so that 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. 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 to a virtual cluster, follow these steps:

1. Log into the Administrative Interface using a login that has **add/del** access for the cluster (see “Logging In” on page 52).
2. In the left frame, right-click the name of the cluster to be configured and select the **Add Server** command from the menu.
3. Enter the following information:

Server Name	The logical name for the server, or accept Equalizer’s default. Each server must have a unique name in the cluster that begins with an alphabetical character (for example, <i>CPIimages</i>).
Server IP Address	Enter the dotted decimal IP address of the server. This is the address Equalizer uses to communicate with the server.
Server Port	<p>Enter the numeric port number on the Equalizer to be used for traffic between Equalizer and the server. For Layer 7 clusters, the default server port is 80.</p> <p>[Note that in <i>Layer 7 HTTPS</i> clusters, Equalizer performs all the SSL encryption and decryption and forwards traffic to the servers using the HTTP protocol. This is why when you add servers to an HTTPS cluster, the default server port is 80 (and should always be a port other than port 443).]</p> <p>For L4 UDP and L4 TCP protocol clusters, a cluster <i>port range</i> can be defined. 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. The Server Port defined for a <i>server</i> 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 below).</p>
Associate with Virtual Machine	When enabled, this option leads you through the process of associating this server with a VMware Virtual Machine. See “Associating a Server with a Virtual Machine” on page 317. This option is disabled by default.
Quiesce on creation	When enabled (the default), this option prevents any traffic from being routed to the server by enabling the quiesce option on the server. Once the server is functional, you can disable the quiesce option on the server’s Configuration tab. If the server is already configured for operation when you add it to Equalizer, you can disable this option.

Unless you want to set up port redirection, you can accept the default value; to redirect to a port other than the cluster port, enter the appropriate value for **Server Port**.

If a *port range* has been defined for the Layer 4 cluster to which the server is being added, the **Server Port** field refers to the first port on which to start servicing the cluster’s port range. For example:


Cluster Port Range	Server Port	Port Mapping (cluster to server)
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 as if the client was connecting to the server directly.

Note – For any server in a Layer 4 TCP or UDP cluster, port 20 can only be used for FTP traffic in an FTP cluster (see “Providing FTP Services on a Virtual Cluster” on page 148). Port 20 *can* be used as part of a cluster port range starting below port 20 and ending above port 21, but port 20 *will not* work -- that is, a service cannot be defined on port 20.

Click the **Next** icon  .

- The following screen allows you to associate this server definition with a VMware virtual machine, if Equalizer VLB is licensed and enabled. See “Associating a Server with a Virtual Machine” on page 317.

If you are not using Equalizer VLB on VMware virtual machines as servers, click the **Next** icon  .

- A confirmation screen appears; click **commit** to create the server with the parameters shown.
- The **Configuration** tab for the new server is opened. See the following section for an explanation of the server configuration parameters.

Modifying a Server

The configuration tabs for a server are displayed automatically when a server is added to the system, or by selecting the server name from the left frame Configuration Tree.

1. Log into the Administrative Interface using a login that has at least **write** access for the cluster that contains the server (see “Logging In” on page 52).
2. In the left frame, select the name of the server to modify. The server **Configuration** tab opens in the right frame:

Server parameters

ip	10.0.10.121
port	80
probe port	0
max connections	0
weight	100
hot spare	<input type="checkbox"/>
quiesce	<input type="checkbox"/>
dont probe	<input type="checkbox"/>
dont persist	<input type="checkbox"/>

commit
show defaults
reset

See the table below for an explanation of the server **Configuration** tab parameters:

ip	The dotted decimal IP address of the server. This is the address Equalizer uses to communicate with the server.
port	<p>Enter the numeric port number on the Equalizer to be used for traffic between Equalizer and the server. The default is port 80. (Note that in <i>Layer 7 HTTPS</i> clusters, the server port should be set to something other than 443 since Equalizer communicates with servers in an HTTPS cluster via HTTP.)</p> <p>For L4 UDP and L4 TCP protocol clusters, a cluster <i>port range</i> can be defined. 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. The port defined for a <i>server</i> 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 “Adding a Server to a Cluster” on page 152 for more information on defining a port range and port redirection.</p>
probe port	<p>By default, the server probe port field is set to zero and the Equalizer uses the server’s port field value for all TCP and ACV probes. If probe port is not zero, Equalizer uses the value specified as the port for all TCP and ACV probes.</p> <p>Note: For servers in <i>Layer 7 HTTPS</i> clusters, set probe port to something other than 443, since Equalizer communicates with the servers via HTTP. In many configurations, it is set to the server port.</p> <p>(Note that the server agent port, set on the cluster, remains a separate port that is used only for server agent communication.)</p>

<p>max connections</p>	<p>Sets the maximum number of permitted open connections for the server. Once this limit is reached, no more traffic is routed to the server until the number of open connections falls below this limit. This limit is set by default to 0, which means that there is no maximum connections limit on the server. See “Setting Maximum Connections per Server” on page 158 for more information.</p>
<p>initial weight</p>	<p>A number between 0 and 200 that indicates a server’s processing power relative to the other servers in the cluster. The default is 100. A value of 0 disables the server (no traffic will be routed to the server). For information about selecting an appropriate initial weight, refer to “Adjusting a Server’s Initial Weight” on page 157.</p>
<p>strict max cx</p>	<p>This flag allows you to customize the behavior of the max connections parameter (see above).</p> <p>When strict max cx is <i>enabled</i> (the default), the max connections parameter is interpreted as a strict maximum and is never overridden. If a client attempts to connect to a server that has a number of connections equal to the max connections setting, then the connection is refused.</p> <p>When strict max cx is <i>disabled</i>, the max connections setting will be overridden in any of the following circumstances:</p> <ul style="list-style-type: none"> • a client attempts to connect to a server with the hot spare flag enabled -- this allows hot spares to service more than the max connections setting of connections • a client attempting to connect to a Layer 7 cluster has a persistence cookie and the server identified in the cookie has already reached its max connections limit • a client attempting to connect to a Layer 4 cluster has an existing sticky persistence connection to a server and that server has already reached its max connections limit
<p>hot spare</p>	<p>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 <i>all</i> the other servers in the cluster are down. You should only configure <i>one</i> server in a cluster as a hot spare.</p> <p>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.</p>
<p>quiesce</p>	<p>When enabled, Equalizer avoids sending new requests to the server. This is usually used in preparation for shutting down an HTTP or HTTPS server, and is sometimes also called “server draining”. Please see “Shutting Down a Server Gracefully” on page 160. Note: if a cluster receives a new client request when all non-quiesced servers (including hot spares) are unavailable, then the request will be load balanced across all the quiesced servers in the cluster.</p>
<p>dont probe</p>	<p>Disables High Level Probes (TCP and ACV) for the server. This is usually used to disable probe checks for a particular server without changing the probe settings for the entire cluster.</p>

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 “Maximum Connections Limits, Responders, and Hot Spares” on page 159.
---------------------	--

3. If you made any changes to the default configuration values, click the **commit** button to save your changes.

Configuring Outbound NAT

The **enable outbound NAT** global parameter allows servers on a non-routable network to communicate with hosts on the Internet by mapping the server’s IP address to another IP address that is routable on the Internet. It is disabled by default. Enabling outbound NAT has a performance impact since Equalizer needs to modify every server response. This parameter should only be enabled when the system is configured in dual network mode, and is incompatible with single network mode operation.

In the default outbound NAT configuration, the Network Address Translation (NAT) daemon maps internal server IP addresses to Equalizer’s Default VLAN IP address (or the external interface IP address on the E250GX and legacy ‘si’ systems). You can also configure outbound NAT for individual servers, so server responses appear as if they came from the cluster IP address, instead of Equalizer’s external interface IP address. The address used for outbound NAT can be adjusted on a server by server basis, by clicking on the server in the left frame and opening the **Outbound NAT** tab. Outbound NAT must also be enabled in the global parameters, as shown below.

Enabling Outbound NAT

To enable outbound NAT:

1. Log into the Administrative Interface using a login that has **add/del** access for global parameters (see “Logging In” on page 52).
2. Click on **Equalizer** (or the configured *Failover Peer Name* for this Equalizer) in the left frame, and open the **Networking** tab in the right frame.
3. Enable the check box next to **enable outbound NAT**.
4. Select the **commit** button.

Configuring Outbound NAT for a Server

Each server defined on Equalizer can have a specifically assigned outbound NAT address that overrides the default (Equalizer’s external IP address). Note that outbound NAT must be enabled globally as described in the previous section for server specific outbound NAT settings to take effect.

To configure an outbound NAT address for a server:

1. Log into the Administrative Interface using a login that has **add/del** access on the cluster to which the server belongs (see “Logging In” on page 52).
2. Click on the name of the server in the left frame, and then open the **Outbound NAT** tab in the right frame.

3. Choose one of the following options:

Default address

This option will set the outbound NAT address to the Default setting. This means that if this server's outbound NAT option is not set explicitly in another cluster, the Equalizer's address will be used. If it is set explicitly in another cluster, that address (either Failover or Cluster) will be used.

Equalizer address

This option will set the outbound NAT address for this server (in this and all other clusters) to use the Equalizer's administrative IP address as the source address for outbound traffic.

Failover address

This option will set the outbound NAT address for this server (in this and all other clusters) to use the Failover IP address as the source address for outbound traffic.

Cluster L7-HTTP's address

This option will set the outbound NAT address for this server (in this and all other clusters) to use the IP address of this cluster. This means that if this server (a server with the same IP address) exists in another cluster, that server's Outbound NAT option will be reset to use this address.

commit

cancel

4. Click **commit** to save your selection, which takes affect immeidately.

Using Outbound NAT on a Server IP in Multiple Clusters

Servers are identified in the NAT daemon configuration file by their IP addresses. If a server IP address is listed more than once in the file, it is the *last* NAT setting listed in the file that takes effect for that server IP.

This means that:

- If the same server IP address is used in more than one cluster, then changing the outbound NAT setting for one of the server instances to anything other than **Default address** changes the NAT IP address for all the instances of that server in all clusters.
- If the outbound NAT setting for one of the instances of a server is subsequently changed to something other than the default, then the new setting takes effect for all instances of the server in all clusters.

Adjusting a Server's Initial Weight

Equalizer uses a server's initial weight as the starting point for determining the percentage of requests to route to that server. As Equalizer gathers information about the actual performance of a server against client requests, it adjusts the server's current weight so that servers that are performing well receive a higher percentage of the cluster load than servers that are performing at a slower rate.

When you install servers, set each server's initial weight value in proportion to its "horsepower." All the initial 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 initial weights of these servers to 100 and 50 is equivalent to setting the initial weights to 180 and 90.

Values for server weights can be in the range 0-200, with 0 meaning that no new requests will be routed to the server, essentially disabling the server for subsequent requests. In general, you should use higher initial weights. When the load balancing policy is *not* set to round robin or static weight, using higher initial 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 initial 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 initial 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.

Setting initial Weights for Homogenous Clusters

If all the servers in a cluster have the same hardware and software configurations, you should set their initial weights to the same value initially. We recommend that you use a initial 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 initial weights accordingly and again monitor their performance. You should adjust server weights by small increments; for example, you might set the initial 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 – A change to a server's initial weight is reflected in cluster performance only after Equalizer has load balanced a significant number of new client requests for up to 30 minutes against the cluster in which the servers reside. When testing initial weights, it is most useful to use a load-generating tool to run typical client requests against the cluster to determine appropriate server initial weights.

Setting initial Weights for Mixed Clusters

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

After you assign relative initial 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.

Setting Maximum Connections per Server

By default, the **max connections** server option is set to 0, which means that Equalizer will route traffic to the server whenever the server is selected by the current load balancing settings. If **max connections** is set to a value greater than 0, then Equalizer limits the total number of simultaneously open connections to the server to that value. This restriction applies regardless of the persistence options set on the cluster.

The **max connections** option can be set in Layer 4 TCP, Layer 7 HTTP, and Layer 7 HTTPS clusters. When a server reaches the specified 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 server

To set **max connections** on a server:

1. Log into the Administrative Interface using a login that has **add/del** access for the cluster that contains the server (see “Logging In” on page 52).
2. Do **one** of the following:
 - a. Create a new server: right-click a cluster name in the left frame and select **Add Server**. After you enter and **commit** the basic information, you’ll be taken to the server **Configuration** tab, where you can set max connections as shown in Step 2.
 - b. Modify an existing server: click on the server name in the left frame to display the server’s **Configuration** tab in the right frame.
3. Set **max connections** to a positive integer between 0 and 65535. A zero (the default) means that no connection limit is set for this server. (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.)
4. Select **commit** to save your changes to the server configuration.

Maximum Connections Limits, Responders, and Hot Spares

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

In general, a Responder is easier to configure than setting up a separate server as a hot spare, since the responder runs on Equalizer. However, while Responders are capable of returning only a single HTML page, a hot spare can be configured to return multiple HTML pages and images. See the section “Automatic Cluster Responders” on page 162 for information on configuring a responder.

To use a hot spare, you would usually configure it on Equalizer as follows:

- Set **max connections** 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 connections** limit (or are down) will be forwarded to the hot spare server.
- Enable the **dont persist** flag so that connections made to the hot spare don’t 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 connections** limit.

Interaction of Server Options and Connection Processing

Server option settings have a direct influence on connection and request processing, particularly Layer 4 and Layer 7 persistence. (Note that persistence is set at the cluster level, but can be disabled for individual servers using the **dont persist** option.) The hierarchy of server option settings is shown in the table below:

server disabled (initial weight = 0)	An initial weight of 0 tells Equalizer that no traffic should be sent to the server, disabling the server. This option setting takes precedence over all other options (including persistence, hot spare, etc.).
max connections > 0	If set to a non-zero value, Equalizer limits the total number of simultaneously open connections to the server to that value. This limit is not overridden if the hot spare option is enabled on a server, and is not overridden by a Layer 4 sticky record or Layer 7 persistence cookie for the server in an incoming request.
quiesce enabled	The server is <i>not</i> included in load balancing decisions, so that no <i>new</i> connections will be made to this server. If a request in an incoming connection has an existing Layer 4 sticky record or Layer 7 cookie for a server, however, the request will be sent to that server even when quiesce is enabled. [Note that if dont persist is also enabled on the server, the sticky record or cookie is ignored.]
hot spare enabled	The server is <i>not</i> included in load balancing decisions, so that traffic is sent to this server <i>only</i> when no other server in the cluster is available to accept client connections. If a request in an incoming connection has an existing Layer 4 sticky record or Layer 7 cookie for a server, however, the request will be sent to that server even when hot spare is enabled. [Note that if dont persist is also enabled on the server, the sticky record or cookie is ignored.]

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 initial 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 initial 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. Check the **quiesce** checkbox; then click **commit** to save your changes.

3. Click on **Equalizer > Status > Cluster Summary** and click the cluster name in the table. Watch the quiescing server's number of **active connections**. Once there are no active connections shown, click the server name in the left frame and set the server's weight to zero; click **commit** to save the change.
4. Click on the server name in the left frame and open the **Reporting** tab. 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, you can increment the cluster's **cookie generation** parameter. Click on the cluster name in the left frame and open the **Persistence** tab. Increment the **cookie generation** parameter by 1; then click **commit**.

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. Set the server's weight to zero; click **commit** to save the change. This action prevents Equalizer from routing new connections to the server.
3. Click on **Equalizer > Status > Cluster Summary** and click on the cluster name in the table. 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.

Deleting a Server

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

1. Log into the Administrative Interface using a login that has **add/del** access for the cluster that contains the server (see "Logging In" on page 52).
2. If necessary, shut the server down gracefully before taking it out of service, as shown in the section "Shutting Down a Server Gracefully" on page 160. This is particularly important if the server is in a Layer 4 cluster and may have active connections; see Step 4.
3. In the left frame, right-click the name of the server to be removed and select the **Delete Server** command from the menu.
4. When prompted, click **delete** to confirm that you want to remove the server from the cluster. Clicking **delete** removes the server from the configuration immediately. To cancel the deletion, click **cancel**. If you attempt to delete a server that has active connections:
 - If the server is being deleted from a Layer 4 cluster, clicking delete removes the server from the configuration and immediately terminates all active connections for that cluster IP and server.
 - If the server is being deleted from a Layer 7 cluster, clicking delete removes the server from the configuration, but does not terminate any active connections. Active connections for that cluster IP and server will remain open until they are completed or reach the appropriate timeout.

Automatic Cluster Responders

Responders are not supported on E250GX model Equalizers

A Responder is a server-like object that can be associated with a Match Rule. If an incoming request matches a Match Rule expression and all of the servers specified in the Match Rule are down, a Responder definition in the Match Rule (if present) tells Equalizer to send one of two automatic responses to the client:

- **A customized HTML “sorry page”** that can, for example, ask the client to retry later or go to another URL.
- **A standard HTTP Redirect response** that specifies a return code and redirect URL. When the client receives this page, it is automatically redirected to the redirect URL. Redirect pages can be configured to use parts of the request URL in the HTTP Redirect response (using a regular expression).

Note – In previous releases, HTTP Redirects and “sorry pages” for clusters were configured by defining a separate server as a hot spare -- this server would return a customized page to the client when no other servers are available. Responders allow you to do this without configuring a separate server, and avoid the problem of the hot spare itself becoming unavailable. Please see “Responders and Hot Spares” on page 170 for a discussion of when each is appropriate to use in your configuration.

Note – Responder definitions are not automatically transferred between failover pairs when the **Use SSL Only** failover flag is disabled (see “Enabling Failover Using the Failover Tabs” on page 100). Responders are also not synchronized automatically amongst Envoy sites. In either of these cases, responders need to be created and maintained separately on each Equalizer.

Managing Responders

To display a list of all currently defined Responders, click **Responders** in the left frame to open the **Responders** tab. The table lists all the existing Responders and their configuration:





Name	Type	Status	URL	Actions
resp00	redirect	307 Temporary Redirect	http://www.mycompany.com/sorry.html	 
resp01	sorry			 

Figure 26 The Responders tab

To add a Responder, you can either click the **Add** icon at the bottom of the **Actions** column, or click the Responder name in the left frame (click the plus sign next to **Responders** to display a list of Responder names).

To edit a Responder’s configuration, you can either click the **Edit** icon in one of the rows in the table above, or click the Responder name in the left frame (click the plus sign next to **Responders** to display a list of Responder names).

To delete a Responder, you can either click the **Delete** icon in one of the rows in the table above, or right-click the Responder name in the left frame and select **Delete Responder** from the menu (click the plus sign next to **Responders** to display a list of Responder names). A Responder cannot be deleted if it is currently used in a match rule definition.

Adding a Responder

Responders are a “global” resource: once created, they can be individually assigned to one or more match rules in one or more clusters. Up to 8192 Responders can be created.

1. To create a Responder, you can either:

- Right-click on **Responders** in the left frame and then select **Add New Responder** from the menu.

- Click on **Responders** in the left frame and then select the **Add** icon in the table in the right frame.

The **Add New Responder** dialog appears. By default, the form for creating a **Redirect Responder** is displayed:

- Type a **Name** for the Responder or leave the default name provided.
- Do one of the following:
 - Create a custom HTML page by selecting **Sorry Server**. The dialog changes to a text entry box, into which you can type the HTML that Equalizer will return to clients. The text size limit is 4096 bytes.
 - Create a standard **Redirect** page by supplying the following information in the popup screen:

Status	<p>The HTTP status code to return to the client. The default return code is 307 (Temporary Redirect). Use the drop-down box to choose a different return code:</p> <p>301 (Moved Permanently) 302 (Found) 303 (See Other)</p>
URL	<p>The HTTP Redirect URL: the full URL of the page to which the client will be redirected, as in the following example:</p> <p><code>http://www.coyotepoint.com/redirect/redirect.html</code></p> <p>If a Regular Expression is used to split the client URL into string variables, any variables appearing in the URL are replaced with strings from the request URL. The following is an example of a Redirect URL with named variables:</p> <p><code>http://\$1.\$2.net\$3\$4</code></p> <p>See the section "Using Regular Expressions in Redirect Responders" on page 164.</p>
Regular Expression	<p>An optional POSIX-style regular expression that splits the incoming request URL into variables that can be used for string replacement in the HTTP Redirect URL (see above). See the section "Using Regular Expressions in Redirect Responders" on page 164.</p>

When you are done, click the Next icon (>) at the top of the dialog.

4. In the screen that follows, you can optionally test your responder. Do one of the following:
 - For a **Sorry Server** responder, click the **test** button to see a preview of the page. Click the **close** button to close the preview.
 - For a **Redirect** responder, enter a **Test URL** (or use the default) and click the **test** button to see how the regular expression breaks the test URL into variables for re-use in the URL you supplied in the previous step.
 - Click the Next icon (>) at the top of the dialog to skip testing.
5. On the next screen, do one of the following:
 - Click the Back icon (>) at the top of the screen to review the responder configuration.
 - For a **Sorry Server**, click **commit** to add this responder or **cancel** to close the dialog without adding the responder.
 - For a **Redirect** responder, this screen displays the responder **Redirect URL** and the **Regular Expression** (if supplied).

If you clicked the **test** button on the previous screen, the **Match Components** and **Resulting Redirect** produced by matching the **Test URL** against the **Regular Expression** are also displayed (any variables appearing in the **Redirect URL** are replaced with strings from the **Test URL**).

Click **commit** to add the **Redirect** responder or **cancel** to close the dialog without adding the responder.

Modifying a Responder

1. To modify the configuration of an existing Responder, you can either:
 - Click on the name of the Responder (under **Responders**) in the left frame.
 - Click **Responders** in the left frame and then click on the **Edit** icon in the **Action** column of the table, on the same row as the name of the Responder you want to modify.



The Responder's **Configuration** tab appears.

2. Update the Responder configuration as desired; see the previous section, "Adding a Responder" on page 162, for a description of all Responder parameters.
3. Click **commit** to save your changes.

Plotting Responder Statistics

See the section "Plotting Responder Performance History" on page 208.

Using Regular Expressions in Redirect Responders

In some cases, it may be desirable to examine the URL of an incoming request and re-use parts of it in the URL returned to the client by a Redirect Responder. This is the purpose of the **Regex** field: specify a custom regular expression that is used to:

- parse the URL of an incoming request
- break it down into separate strings (based on the positions of literal characters in the expression)
- assign each string to a named variable

These named variables can then be used in the URL field of the Redirect Responder. When the Responder replies to a client, it performs string substitution on the URL.

Because the purpose of using regular expressions to perform string substitution in Redirect URLs is to parse request URLs into strings, constructing an appropriate regular expression requires an exact knowledge of the format of the request URLs that will typically be coming in to the cluster IP.

Equalizer supports POSIX-style extended regular expressions, as described in Appendix D, "Regular Expression Format" on page 291.

See the examples that follow below to help you understand how regular expressions are constructed and interpreted by Responders.

Example 1 -- HTTPS Redirect

The simplest form of HTTPS redirect involves simply referring the user to the top level of the https:// site, regardless of the path information that may have been included in the original request URL. For example, we could direct all requests for:

```
http://www.example.com/<path>
```

to:

```
https://www.example.com
```

But, this forces the client to re-specify the <path> after the redirect. It would be better to redirect to a URL that includes the path information:

```
https://www.example.com/<path>
```

The following regular expression:

```
^(([^ :/?#]+):)?//(.*)
```

breaks a request URL into the following named variables:

```
$0    http://www.example.com/<path>
$1    http
$2    http:
$3    www.example.com/<path>
```

We can then use these variables in the URL field as shown in the following Responder configuration screen:

Add New Responder ? X

Name:

Type: Redirect Sorry Server

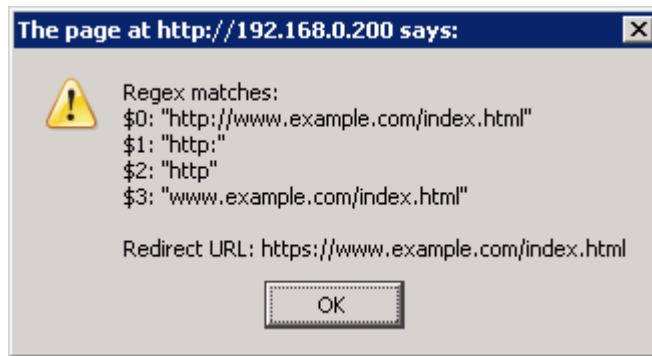
Status: 307 (Temporary Redirect) ▼

URL:

Regex:

Test:

Clicking the **test** button displays a popup that shows the effect of applying the **Regex** to the **Test URL**:



This Responder can be used in any cluster where a Redirect to an HTTPS cluster is desired.

Example 2 -- Multi-Hostname Redirect

Let's assume that we have a set of '.com' hostnames, all of which resolve to the same cluster IP, and we need a Responder that redirects requests to the same hostname prefixes with a '.net' suffix. We also want to include the rest of the URL exactly as specified by the client. For example, we want requests to URLs in these formats:

```
http://www.example.com/<path>
http://www.example2.com/<path>
http://www.example3.com/<path>
```

to be redirected to the following URLs:

```
http://www.example.net/<path>
http://www.example2.net/<path>
http://www.example3.net/<path>
```

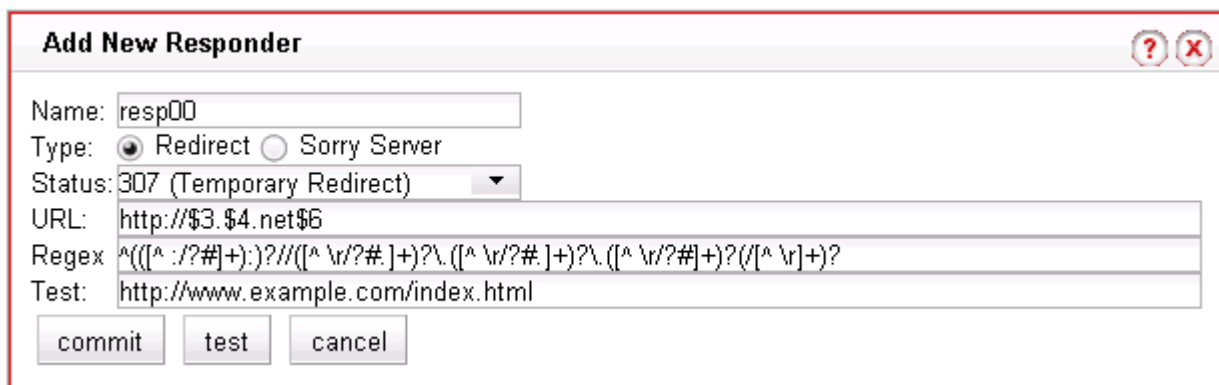
The following regular expression:

```
^(([^ :/?#]+):)?//(([^ \r/?#.]+)?\.[(^\ \r/?#.]+)?\.[(^\ \r/?#]+)?(/^[^ \r]+)?
```

breaks the request URL into the following named variables:

```
$0 http://www.example.com/<path>
$1 http:
$2 http
$3 www
$4 example
$5 com
$6 /<path>
```

We can then use these variables in the URL field as shown in the following Responder configuration screen:



Clicking the **test** button displays a popup that shows the effect of applying the **Regex** to the **Test URL**:



It should be noted that this example will not work for requests with destination URLs specified with an IP address for a hostname (e.g., '12.34.56.78' instead of 'www.example.com'). Providing support for IP addresses in URLs as well as DNS hostnames would involve either: a more complex regular expression that matches both; or, an additional Responder with a regular expression that matches IP addresses, as well as two match rules to match the two types of hostnames (so that the appropriate Responder replies to the client).

Example 3 -- Directory Redirect

The next example involves redirecting requests that include a particular directory to a different domain, omitting the directory from the redirect URL's path. Let's say we want all requests for:

```
http://www.example.com/images/<path>
```

to be redirected to:

```
http://images.example.com/<path>
```

The following regular expression:

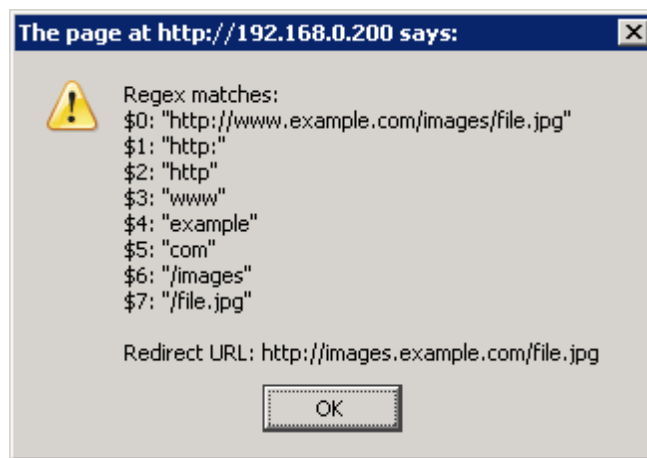
```
(([^\s:/?#]+):)?//([^\s/r/?#.]+)?\.[^\s/r/?#.]+\.([^\s/r/?#]+)?(/([^\s/r]+)?/([^\s/r]+)
```

breaks the request URL into the following named variables:

```
$0      http://www.example.com/images/<path>
$1      http
$2      http:
$3      www
$4      example
$5      com
$6      /images
$7      /<path>
```

We can then use these variables in the URL field as shown in the following Responder configuration screen:

Clicking the **test** button displays a popup that shows the effect of applying the **Regex** to the **Test URL**:



This Responder can be used in a Match Rule in any cluster where a similar directory name based redirect is required.

Using Responders in Match Rules

Once a responder is created, it can be associated with a cluster using a match rule (see “Using Match Rules” on page 221). When adding a responder to a match Rule, the way the match rule is configured has a direct effect on the conditions under which the responder is used:

- **expression:** The default match rule **expression** [`any()`] matches all incoming requests. If you want the responder to be used only for specific requests, then create an appropriate match rule expression to match those requests; see “Using Match Rules” on page 221.
- **server selection:** By default, no servers are selected in a match rule. This means that any incoming request URL that matches the match rule expression will be handled by the responder specified in the match rule. If you want the responder to be used only if no servers (or particular servers) are available, select all (or some) of the the **servers** listed in the match rule configuration screen.

Once a responder is created, it can then be selected in a match rule’s **response** list. The following sections show some common match rule and Responder configurations.

Creating a Match Rule for a “Sorry Page”

The most common use of a responder is to change the default match rule behavior when no servers are available in a cluster. By default, every HTTP and HTTPS cluster is created with a **Default** match rule that does not specify a

Responder -- thus, if all the servers in the **Default** match rule are down, Equalizer drops the client connection to the cluster.

In order to change the default behavior and supply a “sorry page” or redirect for a cluster, you need to add a new match rule that:

- matches any incoming request
- selects all servers in the cluster
- has a **Sorry Server** Responder selected

For example, let’s say you have two Responders defined as in Figure 26, and there is an existing cluster that you would like to redirect to `http://www.example.com` when no servers in the cluster are available. To accomplish this, we need to create a new Responder and then add a match rule to the cluster:

1. Right-click on **Responders** in the left frame and select **Add New Responder** from the popup menu.
2. Type **SorryExample** into the **Name** field and select **Sorry Server**.
3. Type the HTML content for the page to display into the text box that appears, as shown in the following example:

Name:

Type: Redirect Sorry Server

Enter HTML for Sorry page here:

```
We are sorry for the inconvenience, but all of our servers at the selected
location are currently busy. Please go to the following website to select
a different location: <a
href="http://www.example.com">www.example.com</a>.
```

4. Click **commit** to save the new Responder.
5. Right-click on the name of the cluster for which you want to display the sorry page in the left frame and select **Add Match Rule** from the menu.
6. If more than one match rule already exists in the cluster, select the appropriate position for the rule from the immediately before drop-down box.
7. Specify a **Name** for the match rule or accept the default.
8. Leave the match rule **expression** set to the default [`any()`] -- the rule will match all incoming requests.
9. Select *all* the servers in the **servers** field; *the Responder will only be used if none of these servers is available*.
10. Select **SorryExample** in the **response** drop-down box.
11. Click **commit** to save the match rule.

Creating a Match Rule to Redirect All Traffic for a Specific URL

Another common cluster configuration requirement is to be able to automatically redirect all traffic that uses a specific URL. To do this, you need to add a new match rule that:

- matches any incoming request
- selects *none* of the servers in the cluster
- has a **Redirect** Responder selected

For example, let's say that we want all traffic to a cluster that uses the URL `http://cluster/special/` to be redirected to `https://www.example.com/special/`. The following procedure shows you how to add the appropriate Responder and Match Rule:

1. Right-click on **Responders** in the left frame and select **Add New Responder** from the popup menu.
2. Type **RedirectExample** into the **Name** field and select **Redirect**.
3. Type `https://www.example.com/special/` into the **URL** field.
4. Click **commit** to save the new Responder.
5. Right-click on the name of the cluster for which you want to display the sorry page in the left frame and select **Add Match Rule** from the menu.
6. Specify a **Name** for the match rule or accept the default.
7. Leave the match rule **expression** set to the default [`any()`] -- the rule will match all incoming requests.
8. Do *not* select any servers in the **servers** box. *This means that if this match rule's expression selects a request, the Responder we select will respond to the selected request regardless of the status of the servers in the cluster.*
9. Click **commit** to create the match rule.
10. If more than one match rule already exists in the cluster, select the appropriate position for the rule from the **immediately before** drop-down box. Our example redirect rule should be immediately before the *first* existing match rule in the list.
11. Select **RedirectExample** in the **response** drop-down box.
12. Click **commit** to save the match rule.

After completing the above procedure, all client requests to `http://cluster/special/` will be redirected to `https://www.example.com/special/`, even when all the servers in the cluster are available.

More Responder Examples

More examples of using Responders and Match Rules can be found on the **Coyote Point Support Portal**, in the **Device Manuals** section.

<http://support.coyotepoint.com>

Responders and Hot Spares

Responders provide the ability to return either a simple HTML page or an HTTP redirect when no servers in a cluster are available to process a client request. This essentially automates the most basic functions of a hot spare server, and offloads them onto Equalizer. It is therefore tempting, from a cost and efficiency perspective, to use the responder feature exclusively instead of dedicating precious hardware resources to providing hot spare functionality. In some configurations, there are good reasons to consider using a mix of hot spares and responders:

Functionality -- Using a real hot spare server provides the ability to do more than provide a simple HTML page or redirect, which depending on the application may be important in terms of customer satisfaction.

Performance -- The resources Equalizer uses to service client requests using responders are resources potentially taken away from processing other client requests. Responders can have an effect on performance if all the servers in one or more clusters are down during periods of peak usage. In other words, if the system is already using a large percentage of resources (such as CPU and memory) to service client requests, then a sudden burst of responder

activity (because, for example, all the servers in several clusters have gone down) could significantly reduce performance for the entire configuration.

Smart Events are not supported on E250GX model Equalizers

Configuring Smart Events

Equalizer's **Smart Control** feature allows administrators to define **Smart Events** that automate common administrative functions based on pre-set threshold values for system parameters and statistics. For example, you could specify that when the number of active servers in a particular cluster falls below a certain number, then a currently quiesced server is made active.

Equalizer provides a basic set of functions that can be used to define a Smart Event for any cluster. Additional functions are available for clusters containing VMware virtual servers running under Equalizer VLB. To enable Equalizer to communicate with a VMware vCenter (Virtual Center) or single ESX server, see Appendix F, "Equalizer VLB" on page 311. This Appendix also shows you how to configure a cluster for VLB Agents.

In releases previous to Version 8.5, Equalizer VLB (Virtualization Load Balancing) "Basic" adjusts the dynamic weight of servers in a VLB enabled cluster based on the performance data retrieved from VMware. The Equalizer VLB "Advanced" functionality enhances this feature-set, and also adds the capability to control the behavior of virtual machines, rather than just the behavior of the traffic routed to those virtual machines.

Note – VLB-related Smart Event functions require that the VMTools software is installed on your Virtual Machine servers. This is usually added after a Virtual Machine is created. See the VMware documentation for instructions.

Smart Events Components

All clusters include a **Smart Events** tab that lists the currently defined Smart Events for the cluster. A newly created cluster has no Smart Events defined.

A Smart Event consists of:

- a **Trigger** expression
- an **Action** expression

The Trigger expressions of all Smart Events for a cluster are evaluated periodically at an interval set by the **smart timer** parameter on a cluster's **Probe** tab (the default is 15 seconds). If a Trigger expression evaluates to 'true', then the associated Action expression is evaluated, and the action specified by the expression result is performed.

Trigger and Action expressions are logical constructs that use Smart Event functions and operators to specify the conditions under which specific actions are to be performed. A simple example of a Trigger expression is:

```
active_servers < 2
```

This expression evaluates to 'true' when less than 2 servers in the cluster are active (i.e., are not down, quiesced, or designated a hot spare), and causes the associated Action expression to be evaluated. The following example of a simple Action expression prints a message to the system log:

```
log("Cluster c100 has fewer than 2 servers active.")
```

A Smart Event constructed using these Trigger and Action expressions would print a log message any time the number of active servers for the cluster is less than 2.

Smart Event Trigger Expressions

The basic Smart Control model is that a specific trigger expression is set on a cluster -- if this expression evaluates to 'true', then the action expression associated with that trigger is evaluated and an action performed. Trigger and

action expressions are built using a set of Smart Event functions and variables. The Expression Editor displays the appropriate set of functions and variables for trigger and action expressions, as shown in the following table:

Figure 27 Smart Event Trigger Functions and Variables

Trigger Functions & Variables	Description	All or VM only
active_servers	A variable whose value is the current number of active servers for a cluster.	All
connection_server (server)	Returns the number of active connections for the <i>server</i> selected from the drop-down box.	All
event_waiting (event)	Query to see if a event_wait timer is currently in effect for the specified <i>event</i> . Returns 'true' if the specified <i>event</i> is blocked; 'false' if not.	All
false()	Returns a logical 'false' value.	All
is_geosite_active (geocluster, geosite)	Returns true if the specified GeoSite in the specified GeoCluster is currently reported 'up' by Envoy; false otherwise. The <i>geocluster</i> and <i>geosite</i> names must be typed into the text box.	All
numeric (value)	A variable containing a numeric value. Click the drop arrow to enter the value.	All
pick_server (heuristic)	Sets the value of the any server according to the specified heuristic (an integer). The currently supported values for heuristic : 1 = lowest dynamic weight 2 = highest dynamic weight 3 = unquiesced server with lowest current weight 4 = unquiesced server with highest current weight	All
quiesced (server)	Query to see if a server is currently quiesced. Returns 'true' if the specified server's quiesce flag is enabled; returns 'false' if quiesce is disabled.	All
server_probability (server)	Returns the percentage of cluster connections being directed to the <i>server</i> selected from the drop-down box. For example: if a cluster has four servers and uses the round robin load balancing policy, this function will return 25 for any of the servers.	All
server_waiting (server)	Query to see if a server_wait timer is currently in effect for the specified <i>server</i> . Returns 'true' if the specified <i>server</i> is blocked; 'false' if not.	All
true ()	Returns a logical 'true' value.	All
weight_server (server)	Returns the current dynamic weight of the <i>server</i> selected from the drop-down box; between 0 and 200. 0 means that no new requests are being routed to the server, essentially disabling the server. See "Adjusting a Server's Initial Weight" on page 157.	All

Trigger Functions & Variables	Description	All or VM only
cpu_load_server	A variable whose value is the CPU load percentage for the specified <i>server</i> name, which must be associated with a VMware virtual machine. Equalizer queries VMware for the current load on the virtual machine. VMware returns the percentage of CPU resources currently in use.	VM only
powered (server)	Queries VMware and returns 'true' if the specified <i>server</i> name is associated with a virtual machine that is currently powered on -- regardless of whether the guest operating system for the virtual machine is running or not. Otherwise, returns 'false'.	VM only
running(server)	Queries VMware and returns 'true' if the specified <i>server</i> name is associated with a virtual machine whose guest operating system is currently running; otherwise, returns 'false'. For example, the powered() function would return 'true' and the running() function would return 'false' if the virtual machine used as an argument to both had been powered on and went directly into the system BIOS (this is a configurable option in VMware).	VM only

Smart Event Action Functions and Variables

The following table lists the Smart Event Action Functions:

Figure 28 Smart Event Action Functions and Variables

Action Functions & Variables	Description	All or VM only
email (address, message)	Send the specified <i>message</i> to the specified email <i>address</i> . The <i>address</i> must be in <i>user@domain</i> format (for example, joe@example.com). There is no character limit on either parameter. Returns true if successful, false otherwise. Email notification must be enabled in order for email to be sent. See "Configuring Email Notification" on page 214.	All
event_wait (event, seconds)	Sets the event_wait timer for the specified <i>event</i> , which blocks the <i>event</i> from being evaluated for the given number of <i>seconds</i> . This function always returns 'true'.	All
false ()	Returns a logical 'false' value.	All
ipmi_poweroff (BMC IP, BMC username, BMC password, lan)	Sends a power off command to a server with a Baseboard Management Controller (BMC) and Intelligent Platform Management Interface (IPMI) driver installed and configured. See the section "Using IPMI to Power Servers On/Off" on page 177.	All

Action Functions & Variables	Description	All or VM only
ipmi_poweron (<i>BMC IP, BMC username, BMC password, lan</i>)	Sends a power on command to a server with a Baseboard Management Controller (BMC) and Intelligent Platform Management Interface (IPMI) driver installed and configured. See the section “Using IPMI to Power Servers On/Off” on page 177.	All
log (“ <i>message</i> ”)	Print the specified message to the system log. Always returns ‘true’.	All
max_connection (<i>server,max</i>)	Sets the dynamic maximum number of concurrent connections for the specified <i>server</i> to <i>max</i> . This does not affect the max_connections setting in the server Configuration tab. Any subsequent configuration change that writes a new configuration file will reset the dynamic maximum number of concurrent connections to the max_connections setting.	All
numeric (<i>value</i>)	A variable containing a numeric value. Click the drop arrow to enter the value.	All
pick_server (<i>heuristic</i>)	Sets the value of the any server according to the specified heuristic (an integer). The currently supported values for heuristic : 1 = lowest current weight 2 = highest current weight 3 = unquiesced server with lowest current weight 4 = unquiesced server with highest current weight	All
quiesce (<i>server</i>)	Enable the quiesce option for the indicated <i>server</i> . Returns ‘true’ if the quiesce option is set by this function, ‘false’ if not (for example, if the option is already set).	All
reboot_equalizer()	The reboot_equalizer() function supports failover between two Equalizers that use <i>different gateways</i> . It is intended to be used in a Smart Event whose Trigger either performs a health check or minimum active server check on a gateway IP, and reboots Equalizer when the check fails. If the unit rebooted is the primary Equalizer in the failover pair, the backup Equalizer (which uses another gateway IP) will become primary when the reboot occurs.	All
server_wait (<i>server, seconds</i>)	Sets the server_wait timer for the specified <i>server</i> , which blocks any function from operating on the <i>server</i> for the given number of <i>seconds</i> . This function always returns ‘true’. Events called on blocked servers return ‘false’.	All
set_weight (<i>server, weight</i>)	Sets the dynamic (current) weight of the specified <i>server</i> to the specified <i>weight</i> (0-200). [See “Adjusting a Server’s Initial Weight” on page 157 for a description of how Equalizer uses server weight values.] This does not affect the initial weight setting in the server Configuration tab. Any subsequent configuration change that writes a new configuration file will reset the dynamic weight of the server to the initial weight setting.	All
true ()	Returns a logical ‘true’ value.	All

Action Functions & Variables	Description	All or VM only
unquiesce (server)	Disable the quiesce option for the indicated <i>server</i> . Returns 'true' if the option is successfully disabled, 'false' if not.	All
power_off (server)	Power off a VMware virtual server without first shutting down the server. Should usually be preceded by quiesce_server() and an event_wait() to make sure that the shutdown occurs when there are no more live connections on the server. Returns 'true' if the server was powered off; 'false' otherwise.	VM only
power_on (server)	Power on a VMware virtual server. Returns 'true' if the server was powered on; 'false' otherwise.	VM only
shutdown (server)	Equalizer VLB asks VMware to shut down a virtual server. Should usually be preceded by quiesce_server() to make sure that the shutdown occurs when there are no more live connections on the server. Returns 'true' if the server was shutdown; 'false' otherwise.	VM only

Smart Event Operators

The functions in the tables above can be combined using the operators shown below:

Operators	Description
== , > , <	numeric equals, greater than, less than
 , && , !	logical OR, AND, NOT
()	group two or more functions and operators
X	remove the selected function, variable, or operator from the expression

Figure 29 Smart Event Action and Trigger Operators

Smart Event Configuration Parameters

A Smart Event is created with default parameter settings, as discussed in the table below. Click the name of the Smart Event in the left frame and open the **Configuration > Parameters** tab in the right frame to view the current parameter settings for the event:

window timer	This parameter is used by the weight_server() trigger function <i>only</i> , and specifies the number of samples used for the CPU weight calculation. The default value is 5, which means that weight_server() will look at the last 5 server dynamic weight values recorded in server statistics and average them. The result is the dynamic weight value returned by the function.
---------------------	--

event timer	Specifies an event-specific timer frequency for executing an event, in seconds. The default is 15 seconds, which is the same as the default global event interval (see “Global Probe Parameters” on page 90). An event that uses the default values for event interval and event timer will be processed by Equalizer every 15 seconds. The event timer parameter has no effect on event execution unless it is set to a higher value than the global event interval . See “Setting Event Timing Parameters” on page 176.
event priority	Used to establish a priority order for processing Smart Events. The default is 0, which is the <i>highest</i> priority. A new event is given a priority of 0 and is added to the end of the list of priority 0 events in the left frame. Events are evaluated/executed in the order in which they appear in the left frame. Changing an event’s priority will change its position in the left frame event list.

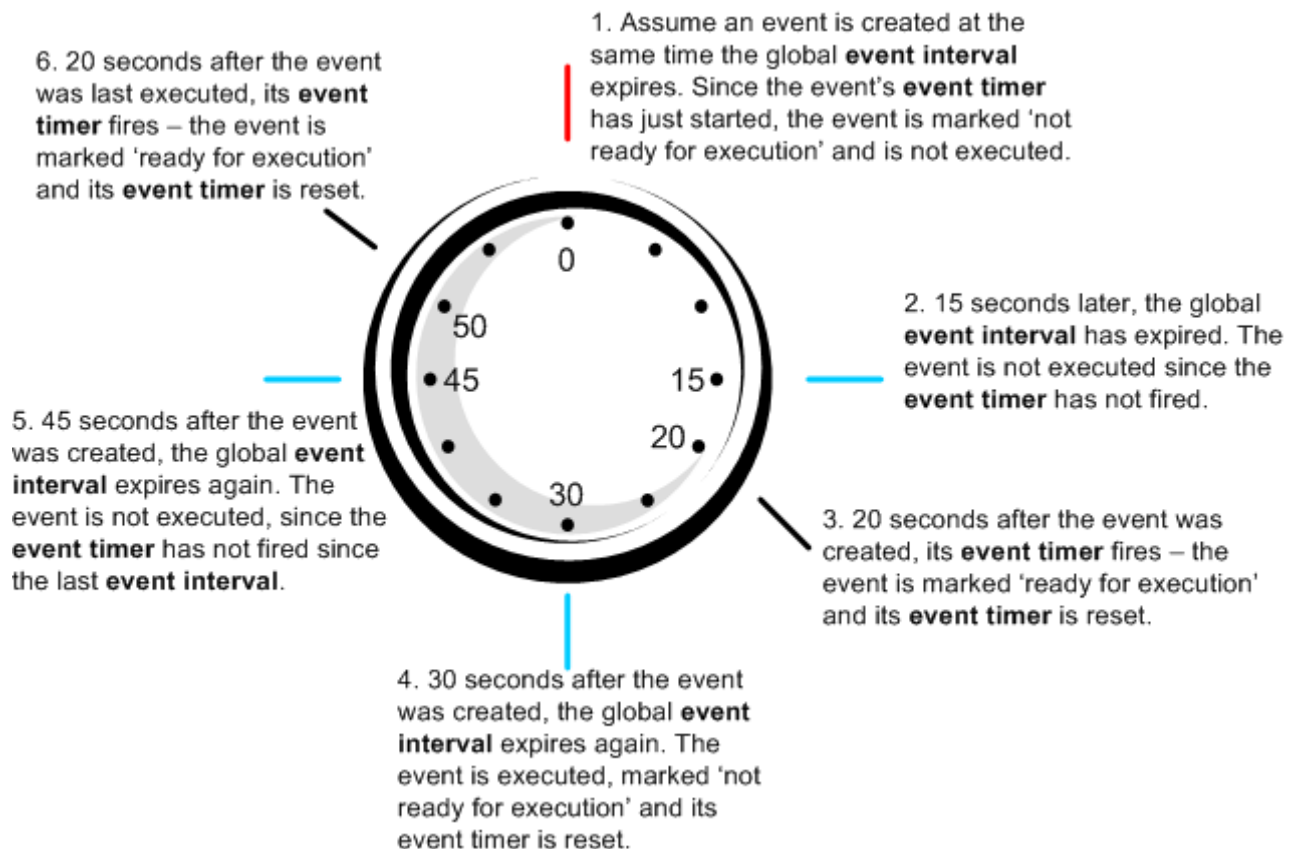
Setting Event Timing Parameters

The global **event interval** determines when events are processed by Equalizer for possible execution. The per-event **event timer** specifies when an individual event is marked ready for execution. The relationship between the event timer and event interval determines when the event will fire as follows:

- If **event timer** \leq **event interval**, then the event will be executed each time the **event interval** expires.
- If **event timer** $>$ **event interval**, then the event will be executed at every n^{th} **event interval** period, where n is determined by dividing the **event timer** value by the **event interval**, and rounding up:

$$n = \text{ceiling} (\text{event timer} / \text{event interval})$$

For example, if the global **event interval** is 15 and you set the **event timer** for an event to 20, then the event will be executed after every 2 event intervals, or about every 30 seconds. This is illustrated in the figure below.



Using IPMI to Power Servers On/Off

The Intelligent Platform Management Interface (IPMI) is an open standard for software-based control of hardware functions, such as powering the system on and off. IPMI is implemented by a driver (OpenIPMI) and a set of software tools (IPMITools) that communicate with the driver either from the local machine or over a LAN connection. Using IPMI, it is possible to power systems on and off using Smart Events on Equalizer (see the `ipmi_*` functions in the section “Smart Event Action Functions and Variables” on page 173).

In order to use an IPMI function to control a server, the server must have a Baseboard Management Controller (BMC), a separate network interface that provides IPMI services. The BMC is usually enabled and configured via the system BIOS, which must be accessed when the system boots. If the installed operating system on the server has an IPMI driver installed and configured, you may also be able to configure the BMC from the command line or using graphical utilities. The tools used to configure BMC controllers and IPMI drivers are specific to a server’s hardware and OS platform. See the hardware and operating system documentation for your servers for specific BMC and IPMI configuration instructions.

The BMC needs to be configured with the following information:

- The **IP address** on which to listen for IPMI requests.
- A **username** and **password**.

The **IP address**, **username**, and **password** specified when configuring the IPMI driver on the server must be provided as arguments to the IPMI functions used in your Smart Event on Equalizer, so that Equalizer can log into the IPMI subsystem on the server. The final argument to the IPMI functions is always “lan”. An example of a valid IPMI function call is:

```
ipmi_poweron("10.0.0.92","bmcroot","bmcpasswrd","lan")
```

For an example of using IPMI functions in Smart Events, see the section “Using IPMI to Conserve Server Resources” on page 183.

Complex Smart Event Expressions

The Smart Control language is flexible and allows you to combine functions, variables, and operators to create complex expressions. For example, the order of processing in a Smart Event expression honors ‘short-circuiting’ rules; this includes ‘chained’ events which use the logical OR and AND operators to decide whether expressions on the right side of the operator are evaluated.

Essentially, this means that the right hand side of an expression is not evaluated if the evaluation of the left side of the expression determines the outcome. For example: if the left side of a two-operand expression that uses the `||` operator evaluates to ‘true’, the right side of the expression is not evaluated. Similarly, if the left side of a two-operand expression that uses the `&&` operator evaluates to ‘false’, the right side of the expression is not evaluated.

Consider an event with the following **trigger** expression:






```
active_servers < 5 && server_wait(1000)
```

This trigger expression essentially overrides the default **smart timer** value (the time interval between event evaluation) if there are less than 5 servers active in the cluster: if there are less than 5 active servers, a timer is set to 1000 seconds, and after this timer expires the **action** expression for the Smart Event is processed. Therefore, this event will not be evaluated again for at least 1000 seconds. If there *are* 5 or more active servers, the event will be evaluated again after **smart timer** seconds.

Note that ‘short-circuiting’ in logical expressions is a standard feature of many programming languages. More information on constructing logical expressions can be found in programming texts and on the Internet.

Managing Smart Events

Smart Events are a per-cluster resource, and so are listed under the cluster name in the left frame object tree, after the servers in the cluster. Click on an existing Smart Event name to edit the event as discussed below. Click on any cluster name in the left frame and then open the **Smart Events** tab in the right frame to manage all the Smart Events for that cluster. The **Smart Events** tab lists all the currently defined Smart Events for the cluster in a table; initially, it is empty as shown below:

Name	Status	Actions
low_servers	ready	 
manage_spare	blocked	 
		

The **Name** column lists the Smart Event name (supplied when the event is created). The **Status** column can be one of the following: **ready** (the event is ready to be executed) or **blocked** (the event is currently blocked by a `wait_event()` function call). The buttons in the Action column allow you to **Add**, **Edit**, and **Delete** Smart Events.

Adding a Smart Event

- To add a Smart Event to a cluster, do *one* of the following:
 - Right-click the cluster name in the left frame and select **Add Event** from the menu.
 - Click the cluster name in the left frame, open the **Smart Events** tab, and then click on the **Add** icon in the **Action** column of the table.
- The **Add New Event** dialog is displayed. Enter a unique event name and click Next (>).
- Enter the Trigger expression using the expression editor. See “Using the Smart Event Expression Editor” on page 179. When you are done, click Next (>).
- Enter the Action expression using the expression editor. See “Using the Smart Event Expression Editor” on page 179. When you are done, click Next (>).
- The trigger and action you have entered are displayed for confirmation; click **commit** to save the new event.



Editing a Smart Event

- To view and edit a Smart Event, do *one* of the following:
 - Click on the Smart Event name in the left frame. (Use the expand control (plus sign) next to a cluster name to see all the Smart Events defined for the cluster).
 - Click the cluster name in the left frame, open the **Smart Events** tab, and then click on the **Edit** icon in the **Action** column of the table.
- The **Configuration > Trigger** tab for the new event is opened. See “Using the Smart Event Expression Editor” on page 179. If you edit the expression, click **commit** to save your changes.
- Click on the **Action** tab to edit the action expression for the Smart Event. See “Using the Smart Event Expression Editor” on page 179. If you edit the expression, click **commit** to save your changes.



Deleting a Smart Event

- To delete a Smart Event, do *one* of the following:
 - Right-click the Smart Event name in the left frame and select **Delete Event** from the menu. (Use the expand control (plus sign) next to a cluster name to see all the Smart Events defined for the cluster.)
 - Click the cluster name in the left frame, and open the **Smart Events** tab. Click on the name of the Smart Event you want to delete in the table and then click the **Delete** icon in the **Action** column of the table.
- A confirmation dialog is displayed. Click **delete** to delete the Smart Event.



Delete icon

Displaying Smart Event Statistics

- To display statistics for a Smart Event, do *one* of the following:
 - Click on the Smart Event name in the left frame. (Use the expand control (plus sign) next to a cluster name to see all the Smart Events defined for the cluster).
 - Click the cluster name in the left frame, open the **Smart Events** tab, and then click on the **Edit** icon in the **Action** column of the table.
- Open the **Reporting** tab. The following statistics are displayed:



Edit icon

number of times this event was processed	A Smart Event is processed, by default, every 15 seconds (see the event interval parameter under “Global Probe Parameters” on page 90).
number of times this event was fired	The number of times a Smart Event’s trigger expression evaluated to <i>true</i> .
number of times this event was blocked	The number of times a Smart Event was blocked by an event_wait() function call.

Using the Smart Event Expression Editor

Smart Control Events can be entered using a graphical editor in the Equalizer GUI. When a new event is created, you must enter a unique name for the event on the first wizard screen (a name is defaulted for you if you would not like to enter your own). The following two screens allow you to enter a **trigger** and **action** for this event.

The Expression Editor lets you choose the elements of a trigger or action expression from a menu displayed at the top of the editor screen, and manipulate them easily with the mouse. To use the editor:

- Click the button with the name of a function, variable, or operator. See Figure 27 and Figure 29 on page 175 for a list of the expression elements supported.
- The name of the function or variable appears in the edit box below; if you chose the **variable** button (used to enter data values), a blank element appears.
- If the function you chose requires a parameter, click the down arrow next to the function name in the edit box to supply the argument. Click **accept** to save the value, which now appears in the edit box.
- Click on an expression element in the edit box and then click an element in the menu to add the menu element to the expression after the element selected in the edit box.
- Click on an element and hold the mouse button to drag the selected element to a new position in the expression; release the mouse button to drop the element into its new position.
- Click on an element and select the **X** menu item to remove the selected element from the expression.

When your expression is complete, click the **commit** button to save the expression.

Smart Event Examples

Several examples of Smart Events are presented in this section, using functions that are available to all clusters. (For examples of Smart Events that use Equalizer VLB-only functions, see the section “Smart Control Event Examples Using VLB” on page 318.)

Logging a Message When Server Count is Low

Let’s say we want to create a Smart Event for a cluster that prints a message to the Equalizer log any time there are fewer than 2 servers active in the cluster. To create this event:

1. Right-click on a cluster name in the left frame and select **Add Event** from the popup menu.
2. Type a **Name** for the event (or accept the default) and click the Next icon (>). This opens the **Event Trigger** expression editor:

Event Trigger

3. In the **operators** field, click on the following controls:
 - a. **active_servers**
 - b. the ‘less than’ operator (<).
 - c. **numeric**

The **expression workbench** field should now look like this:

4. In the **expression workbench** field, click on the drop-down arrow shown in the blank parameter. Type ‘2’ into the **numeric value** text box and then click **accept**.
5. At the top of the **Add New Event** popup window, click the Next icon (>). This displays the **Event Action** editor.
6. In the **functions** field, click on **log**.
7. In the **expression workbench** field, click on the drop-down arrow next to **log()**. Type the following into the message text box that appears:

There are fewer than two servers active.

When you finish typing the message, click **accept**. The **expression workbench** field should now look like this:

8. At the top of the **Add New Event** popup window, click the Next icon (>).
9. A confirmation screen appears that summarizes the new event. Click **commit** to save the new event. The new event is now active, and will be evaluated at the next **smart timer** interval. Equalizer opens the event's configuration tabs so you can inspect the event definition and make further changes, if necessary.

Unquiescing a Server When Server Count is Low

Let's say we have a cluster that has three servers, **sv00**, **sv01**, and **sv02**:

- We want **sv00** and **sv01** to actively serve traffic, and **sv02** to have the **quiesce** option enabled while the other two are running.
- If one (or both) of the active servers becomes unavailable, we want to disable the **quiesce** option on **sv02** so it starts serving traffic.
- Once the original two active servers are available again, we want to quiesce the third server.

The above can be accomplished by constructing two Smart Events:

Event 1: If there are less than two servers active, unquiesce **sv02** and print a log message.

Event 2: If there are three servers active, quiesce **sv02** and print a log message.

To create **Event 1**, do the following:

1. Right-click on the cluster name in the left frame and select **Add Event** from the popup menu.
2. Type a **Name** for the event, such as **unquiesce-sv02**, or accept the default. Click the Next icon (>) to open the **Event Trigger** expression editor.
3. In the **operators** field, click on the following controls:
 - a. **active_servers**
 - b. the 'less than' operator (<).
 - c. **numeric**

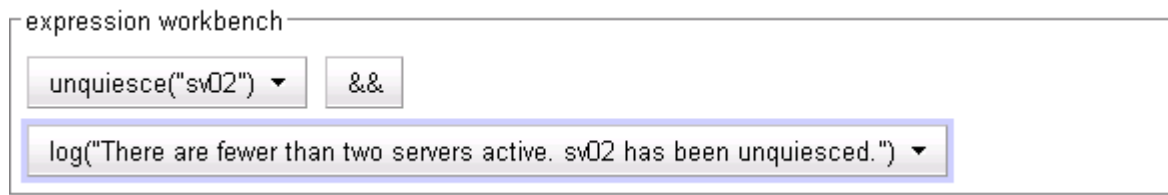
The **expression workbench** field should now look like this:



4. In the **expression workbench** field, click on the drop-down arrow shown in the blank parameter. Type '2' into the **numeric value** text box and then click **accept**.
5. At the top of the **Add New Event** popup window, click the Next icon (>). This displays the **Event Action** editor:
6. In the **functions** field, click on **unquiesce**.
7. In the **expression workbench** field, click on the drop-down arrow next to **unquiesce**. Select **sv02** in the popup dialog and click **accept**.
8. In the **operators** field, click on **&&**.
9. In the **functions** field, click on **log**.
10. In the **expression workbench** field, click on the drop-down arrow next to **log()**. Type the following into the message text box that appears:

```
There are fewer than two servers active. sv02 has been unquiesced.
```

When you finish typing the message, click **accept**. The **expression workbench** field should now look like this:



11. At the top of the **Add New Event** popup window, click the Next icon (>).
12. A confirmation screen appears that summarizes the new event. Click **commit** to save the new event.

To create **Event 2**, do the following:

1. Right-click on the cluster name in the left frame and select **Add Event** from the popup menu.
2. Type a **Name** for the event, such as **quiesce-sv02**, or accept the default. Click the Next icon (>) to open the **Event Trigger** expression editor.
3. In the **operators** field, click on the following controls:
 - a. **active_servers**
 - b. the 'equals' operator (==).
 - c. **numeric**

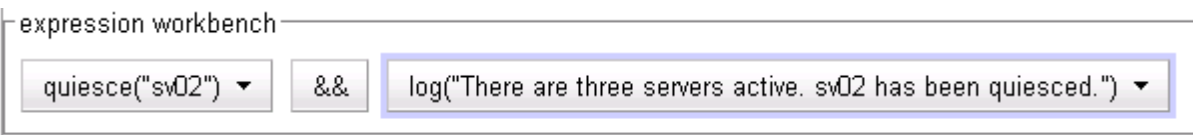
In the **expression workbench** field, click on the drop-down arrow shown in the blank parameter. Type '3' into the **numeric value** text box and then click **accept**. The **expression workbench** field should now look like this:



4. At the top of the **Add New Event** popup window, click the Next icon (>) to open the **Event Action** editor.
5. In the **functions** field, click on **quiesce**.
6. In the **expression workbench** field, click on the drop-down arrow next to **quiesce**. Select **sv02** in the popup dialog and click **accept**.
7. In the **operators** field, click on **&&**.
8. In the **functions** field, click on **log**.
9. In the **expression workbench** field, click on the drop-down arrow next to **log(“”)**. Type the following into the message text box that appears:

There are three servers active. sv02 has been quiesced.

When you finish typing the message, click **accept**. The **expression workbench** field should now look like this:



10. At the top of the **Add New Event** popup window, click the Next icon (>).
11. A confirmation screen appears that summarizes the new event. Click **commit** to save the new event.

Using IPMI to Conserve Server Resources

Smart Events with IPMI functions can be used for power management of server resources. For example, let's say we have a cluster whose traffic can be handled during non-peak hours by two non-IPMI enabled servers -- these servers are always powered on. During peak operating periods, one additional (IPMI-enabled) server is required to handle increased traffic, and is always powered on with the **hot spare** server option enabled.

Using Smart Events, the additional server can be brought online during peak traffic periods and powered down when the traffic is being handled by the two non-IPMI servers. Smart Rules will also provide a dynamic hot spare: should one or both of the non-IPMI servers become unavailable, the IPMI-enabled server will be powered up -- and powered down once the server that went down is available again.

If we assume a cluster with two non-IPMI servers (**sv01** and **sv02**) and one IPMI-enabled server (**ipmi01**), the logic of the events is shown in the table below:

<p>Event peak-on-ipmi01:</p> <p>If the number active connections to sv01 and sv02 are both above some threshold:</p> <ul style="list-style-type: none"> • Power on ipmi01 and log a message. • Set a wait timer on server ipmi01, so that no events will affect the server for the duration of the timer. At a minimum, set the timer to the time it takes ipmi01 to boot and be available to handle traffic. • Set the event timer on peak-off-ipmi01 to 0 to enable it. • Set a long event timer on this event (such as 864000, or 10 days), so that it does not continually fire while the trigger conditions are true. 	<p>Event peak-off-ipmi01:</p> <p>If the number active connections to sv01 and sv02 are both below some threshold:</p> <ul style="list-style-type: none"> • Power off ipmi01 and log a message. • Set a wait timer on server ipmi01, so that no events will affect the server for the duration of the timer. At a minimum, set the timer to the time it takes ipmi01 to shut down completely. • Set the event timer on peak-on-ipmi01 to 0 to enable it. • Set a long event timer on this event (such as 864000, or 10 days), so that it does not continually fire while the trigger conditions are true.
<p>Event spare-on-ipmi01:</p> <p>If the dynamic weights of <i>either</i> sv01 or sv02 are below 1, then:</p> <ul style="list-style-type: none"> • Power on ipmi01 and log a message. • Set a wait timer on server ipmi01, so that no events will affect the server for the duration of the timer. At a minimum, set the timer to the time it takes ipmi01 to boot and be available to handle traffic. • Set the event timer on spare-off-ipmi01 to 0 to enable it. • Set a long event timer on this event (such as 864000, or 10 days), so that it does not continually fire while the trigger conditions are true. 	<p>Event spare-off-ipmi01:</p> <p>If the dynamic weights of <i>both</i> sv01 and sv02 are greater than 0 and there are no active connections to ipmi01, then:</p> <ul style="list-style-type: none"> • Power off ipmi01 and log a message. • Set a wait timer on server ipmi01, so that no events will affect the server for the duration of the timer. At a minimum, set the timer to the time it takes ipmi01 to shut down completely. • Set the event timer on spare-on-ipmi01 to 0 to enable it. • Set a long event timer on this event (such as 864000, or 10 days), so that it does not continually fire while the trigger conditions are true.

For our example, note the following:

- We'll use 1000 connections as the number of connections on **sv00** or **sv01** that tells us that they have reached a peak operating period.
- When they both go below 100 connections, the peak period has ended.

- We'll assume that **ipmi-server** can start serving traffic 180 seconds after it is powered on, and that it takes the same number of seconds to shut down completely, and use 180 seconds for the wait timer on **ipmi01** in the event actions. These times could be adjusted (i.e., increased) in a production environment to prevent **ipmi-server** from bouncing up and down during high traffic spikes. Low server wait times relative to the wait set on the event timers (see the next bullet) should be used, however, since in this example the server wait timer is not explicitly set to 0 by any event action.
- Each event will also set an event timer on itself so that it does not re-fire until its associated event has fired, and will clear the event timer on its associated event (see the last two bullet items in each event description, above).

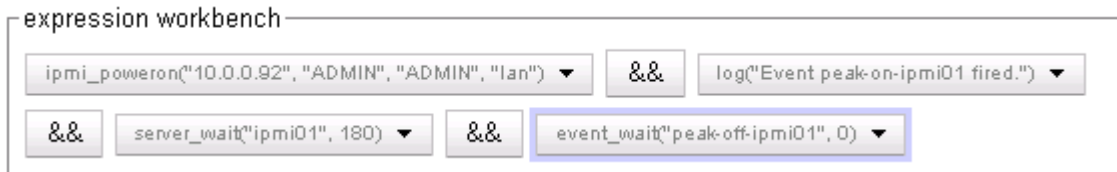
To create the **peak*** and **spare*** events shown in the table above, do the following:

1. Right-click on the cluster name, and select **Add Event** from the popup menu:
 - a. Type **peak-on-ipmi01** into the **Event Name** text box and click the next icon (>).
 - b. Construct the **Event Trigger** shown below using the expression editor controls:



Click the next icon (>).

- c. Construct the **Event Action** shown below using the expression editor controls:



Click the next icon (>).

- d. Click **commit** to create the **peak-on-ipmi01** event. The object tree at left refreshes to display the new event.
- e. Click on the new event name and open the **Action** tab in the right frame. Use the expression editor to add an event wait timer for this event, as shown below:



- f. Click **commit**.

2. Right-click on the cluster name, and select **Add Event** from the popup menu:
 - a. Type **peak-off-ipmi01** into the **Event Name** text box and click the next icon (>).
 - b. Construct the **Event Trigger** shown below using the expression editor controls:



Click the next icon (>).

- c. Construct the **Event Action** shown below using the expression editor controls:

expression workbench

ipmi_poweroff("10.0.0.92", "ADMIN", "ADMIN", "lan") && log("Event peak-off-ipmi01 fired.") && server_wait("ipmi01", 180) && event_wait("peak-on-ipmi01", 0)

Click the next icon (>).

- d. Click **commit** to create the **peak-off-ipmi01** event. The object tree at left refreshes to display the new event.
 e. Click on the new event name and open the **Action** tab in the right frame. Use the expression editor to add an event wait timer for this event, as shown below:

expression workbench

ipmi_poweroff("10.0.0.92", "ADMIN", "ADMIN", "lan") && log("Event peak-off-ipmi01 fired.") && server_wait("ipmi01", 180) && event_wait("peak-on-ipmi01", 0) && event_wait("peak-off-ipmi01", 864000)

- f. Click **commit**.

3. Right-click on the cluster name, and select **Add Event** from the popup menu:

- a. Type **spare-on-ipmi01** into the **Event Name** text box and click the next icon (>).
 b. Construct the **Event Trigger** shown below using the expression editor controls:

expression workbench

"weight_server_sv00" < 1 || "weight_server_sv01" < 1

Click the next icon (>).

- c. Construct the **Event Action** shown below using the expression editor controls:

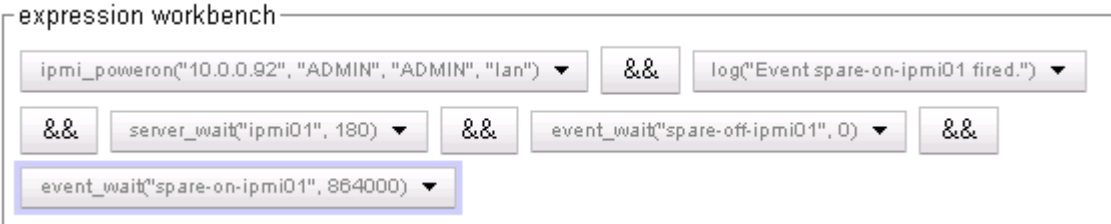
expression workbench

ipmi_poweron("10.0.0.92", "ADMIN", "ADMIN", "lan") && log("Event spare-on-ipmi01 fired.") && server_wait("ipmi01", 180) && event_wait("spare-off-ipmi01", 0)

Click the next icon (>).

- d. Click **commit** to create the **spare-on-ipmi01** event. The object tree at left refreshes to display the new event.

- e. Click on the new event name and open the **Action** tab in the right frame. Use the expression editor to add an event wait timer for this event, as shown below:

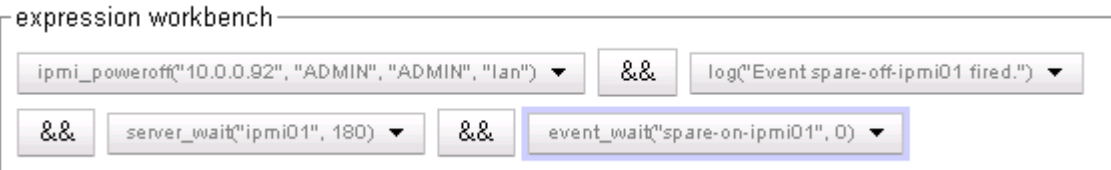


- f. Click **commit**.
4. Right-click on the cluster name, and select **Add Event** from the popup menu:
 - a. Type **spare-off-ipmi01** into the **Event Name** text box and click the next icon (>).
 - b. Construct the **Event Trigger** shown below using the expression editor controls:



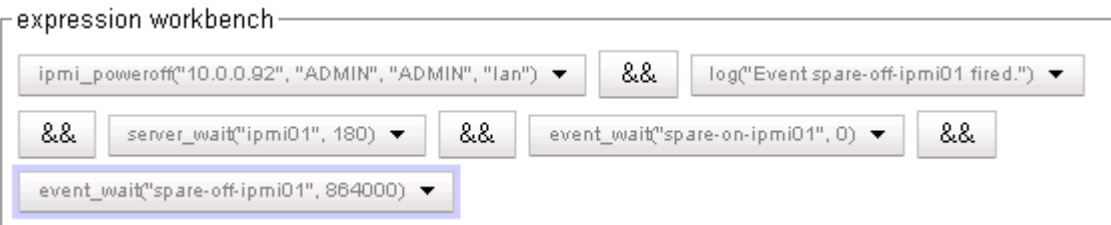
Click the next icon (>).

- c. Construct the **Event Action** shown below using the expression editor controls:



Click the next icon (>).

- d. Click **commit** to create the **spare-off-ipmi01** event. The object tree at left refreshes to display the new event.
- e. Click on the new event name and open the **Action** tab in the right frame. Use the expression editor to add an event wait timer for this event, as shown below:



- f. Click **commit**.
5. Once **ipmi-server** is available and ready for use, you can enable all the events. To do this:
 - a. In the left frame, click on the name of one of the events you just created.
 - b. Open the **Required** tab.
 - c. Uncheck the **disable** check box.
 - d. Click **commit**.

- e. Do the above for each event in the cluster. When you are done, click on the cluster name in the left frame and open the **Smart Events** tab in the right; the **Status** column in the table should display “ready” for all the events.

The events you just created will now be processed by the system at the interval specified by the **event timer** on the event's **Required** tab (default: 15 seconds). If your system is in a test environment, use a load generation tool to generate client connections and test the operation of the events (you may need to adjust the number of connections used in the triggers for the **peak*** events to meet the capabilities of your load generation tool). As load is applied and the events fire:

- You can watch the processing counters increase on each event's **Reporting** tab.
- Click your Equalizer's name in the left frame and open the **Status > Event Log** tab in the right to monitor the system log for the log messages configured into each event.

Configuring Direct Server Return (DSR)

In a typical load balancing scenario, server responses to client requests are routed through Equalizer on their way back to the client. Equalizer examines the headers of each response and may insert a cookie, before sending the server response on to the client.

In a Direct Server Return (DSR) configuration, the server receiving a client request responds directly to the client IP, bypassing Equalizer. Because Equalizer only processes incoming requests, cluster performance is dramatically improved when using DSR in high bandwidth applications, especially those that deliver a significant amount of streaming content. In such applications, it is not necessary for Equalizer to receive and examine the server's responses: the client makes a request and the server simply streams a large amount of data to the client.

DSR is supported on Layer 4 TCP and UDP clusters only, and is not supported for FTP clusters (Layer 4 TCP clusters with a start port of 21). Port translation or port mapping (using a different port, or range of ports, on a cluster and the servers in the cluster) is not supported in DSR configurations.

DSR configurations are usually configured in single network mode, where the cluster IP and the server IPs are all on the internal interface. An example single network mode DSR configuration is shown below:

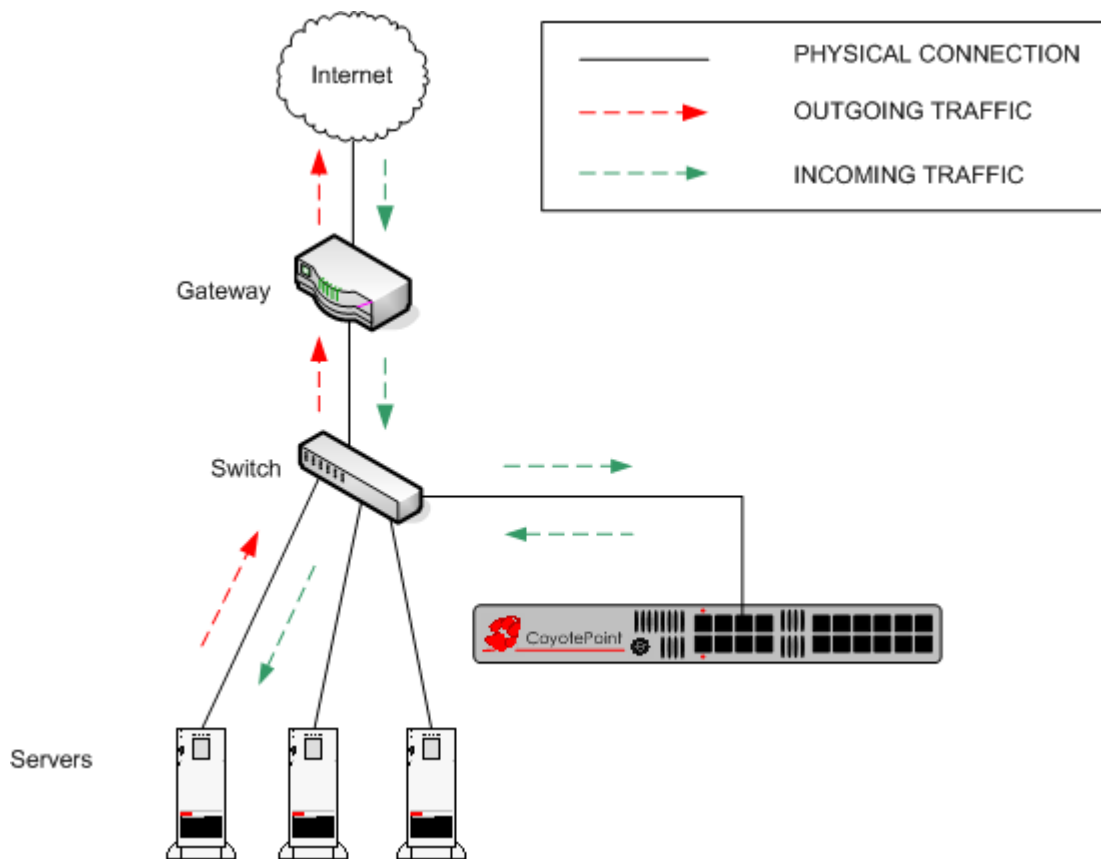


Figure 30 Example of a DSR Single Network Mode Configuration

DSR can also be used in dual network mode, although this is a less common configuration than single network mode. Cluster IPs are on the external interface, and server IPs are on the internal interface. An example of a dual network mode DSR configuration is shown below.

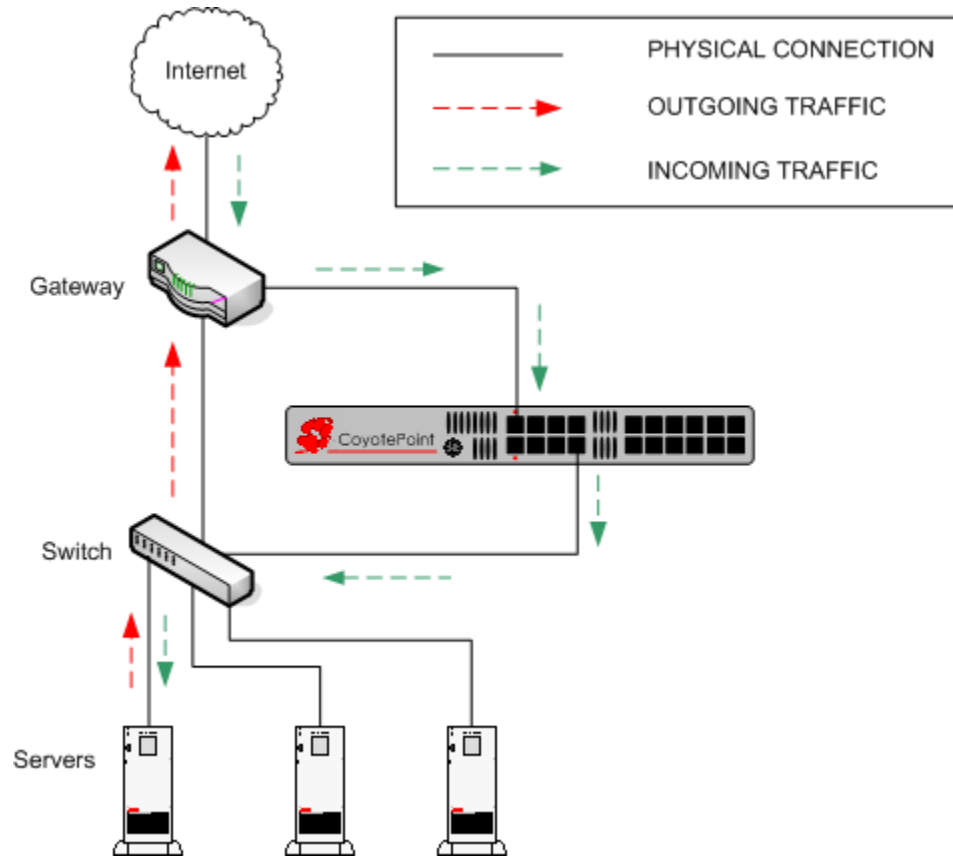


Figure 31 Example of a DSR Dual Network Mode Configuration

Note in both configurations that the incoming client traffic is assumed to originate on the other side of the gateway device for the subnets on which Equalizer and the servers reside. The servers will usually have their default gateway set to something other than Equalizer so that they can respond directly to client requests.

Configuring a Cluster for Direct Server Return

The cluster parameters **direct server return**, **spoof**, and **idle timeout** are directly related to direct server return connections:

direct server return	Enables Direct Server Return. All requests to this cluster IP will be forwarded to the server with the client IP as the source IP, and the cluster IP as the destination IP. The loopback interface of the server must be configured with the cluster IP to receive the requests. See “Configuring Servers for Direct Server Return” on page 190.
spoof	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 <i>client</i> is sent to the server, not the IP address of the Equalizer. This flag must be enabled for DSR.

idle timeout	The time in seconds before reclaiming idle Layer 4 connection records. Applies to Layer 4 TCP clusters only. (See “Layer 4 Connection Timeouts” on page 281 for a full description.) For DSR, idle timeout must be set to a non-zero value, or Equalizer will never reclaim connection records for connections terminated by the server. The cluster’s idle timeout should be set to the longest period within your application that you would like Equalizer to wait for consecutive messages from the client (since the Equalizer does not see server packets on DSR connections). For example, if the longest expected server response time and the longest expected delay between client responses on active connections are both 60 seconds, then set the idle timeout to 120 seconds.
---------------------	--

To create a new cluster or modify an existing one for DSR, do the following:

1. Log into the Administrative Interface using a login that has **add/del** access for the cluster (see “Logging In” on page 52).
2. Do **one** of the following:
 - a. Create a new Layer 4 TCP or UDP cluster: right-click **Equalizer** in the left frame and select **Add Cluster**. After you enter and **commit** the basic information, you’ll be taken to the server **Configuration** tab.
 - b. Modify an existing Layer 4 TCP or UDP cluster: click on the cluster name in the left frame to display the cluster’s **Configuration** tab in the right frame.
3. Enable the **direct server return** and **spoof** check boxes.
4. If the cluster is a Layer 4 TCP cluster and the **idle timeout** parameter is set to **0**, increase it as described in the table above. Skip this step for Layer 4 UDP clusters.
5. Select **commit** to save your changes to the cluster configuration.
6. If there are existing servers in the cluster, ensure that port translation is *not* enabled for the cluster. Do the following for each existing server:
 - a. Click on the cluster and record the value of the **start port** parameter.
 - b. Click on the server name in the left frame and verify that the server **port** is set to the same value as the cluster **start port**. If it is not, set the **port** to the same value as the **start port** parameter and click **commit**.
7. If you need to add servers to the cluster, add them by doing the following:
 - a. Right-click the cluster name in the left frame and select **Add Server**.
 - b. For the server **port**, specify the same port number used for the cluster **start port**.
 - c. Fill in the remainder of the required information.
 - d. Click **commit**.
8. Perform the procedure in the following section on each server that you add to the cluster.

Configuring Servers for Direct Server Return

The following sections shows you how to configure server systems for DSR. These are the physical servers whose IP addresses you added to Equalizer as servers in a Layer 4 cluster with the **dsr** option enabled. Do the following to configure a server for DSR:

1. Add a *loopback* network interface on the server.
2. Configure the loopback interface with the IP address and port of the DSR cluster.
3. Edit the configuration of the application on the server to listen for connections on the cluster IP and port. (An HTTP server, for example, returns a `Bad Hostname` error to the client if there is an IP mismatch.)

4. Check the routing on your network to ensure that traffic is being routed as expected. For example, Equalizer is usually *not* going to be used as the default gateway on your servers, since we want the servers to respond to clients directly. In most DSR configurations, the default gateway used on servers is the gateway most appropriate for reaching the client network. If routes are also needed through Equalizer, they should be configured through static routes on the servers.
5. In DSR configurations where a client device resides on the same side of the gateway as the DSR servers, there is the possibility that the servers will receive the ARP (Address Resolution Protocol) request for the virtual cluster IP address. Since the cluster IP address is configured on the loopback interface of each server (see “Configuring Servers for Direct Server Return” on page 190), one or more may respond to the ARP request. The client, and possibly even the gateway, will then route requests for the cluster IP to servers directly without going through Equalizer. If this occurs, you need to reconfigure the servers so that they do not respond to ARP requests for the cluster IP addresses configured on the loopback interface. The procedure to follow to do this is specific to the operating system running on the servers, so please consult the documentation for your server operating system.

The following sections show examples of configuring the loopback adapter and an HTTP server on Windows and Linux platforms for DSR:

Configuring Windows Server 2003 and IIS for DSR

The basic procedure below also applies to Windows XP and other versions of Windows.

1. Open **Start > Control Panel** and double-click **Network Connections**.
2. Select **View > Tiles**. If a **Microsoft Loopback Adapter** is already listed, proceed to the next step. Otherwise, to install the loopback interface as follows:
 - a. Open **Start > Control Panel > Add Hardware**, and then click **Next**.
 - b. Click **Yes, I have already connected the hardware**, and then click **Next**.
 - c. At the bottom of the list, click **Add a new hardware device**, and then click **Next**.
 - d. Click **Install the hardware that I manually select from a list**, and then click **Next**.
 - e. Click **Network adapters**, and then click **Next**.
 - f. In the **Manufacturer** box, click **Microsoft**.
 - g. In the **Network Adapter** box, click **Microsoft Loopback Adapter**, and then click **Next**.
 - h. Click **Finish**.
3. To configure the loopback interface for DSR:
 - a. In **Network Connections**, right click on the **Microsoft Loopback Adapter** and select **Properties**.
 - b. In the **General** tab, double-click on **Internet Protocol (TCP/IP)** in the scroll box.
 - c. Select **Use the following IP address**, and enter the **IP address** and **Subnet mask** for the Layer 4 cluster, as configured on Equalizer. Click **OK**.
 - d. Click **OK** to return to **Network Connections**.
4. To configure the IIS HTTP server for DSR:
 - a. Open **Start > Administrative Tools > Internet Information Service (IIS) Manager**.
 - b. In the left frame, expand the **local computer** and then **Web Sites** to display a list of the web sites running on the server.
 - c. Right-click on the web site you want to configure for DSR and select **Properties**.
 - d. On the **Web Site** tab, next to **IP address**, select the **Advanced** button.
 - e. Select the **Add...** button under the top list box.
 - f. Enter the **IP address** and the **TCP port** for the Layer 4 cluster, as configured on Equalizer. Click **OK**.
 - g. Click **OK** twice to return to the **Internet Information Service (IIS) Manager**.

You should now be able to send client requests to the cluster IP and port, and get responses directly from the IIS HTTP server running on Windows 2003. Remember that static routes on your servers may be necessary, depending on your network configuration.

Configuring a Linux System running Apache for DSR

This is an example of how to configure a typical Linux system running Apache 2.0 for DSR:

1. Log into the Linux server as *root*, and enter the following command to configure a loopback interface:

```
# ifconfig lo:dsr inet cluster-ip netmask 255.255.255.255
```

Substitute the IP address of the DSR-enabled cluster on Equalizer for *cluster-ip* in the command above.

Note that in most Linux distributions, you are configuring an alias for the loopback interface and should specify a netmask of *255.255.255.255* instead of the netmask used to configure the cluster on Equalizer.

2. Enter the following command to verify that the loopback alias was created:

```
# ifconfig lo:dsr
```

The output should look like this:

```
lo:dsr Link encap:Local Loopback
inet addr:cluster-ip Mask:255.255.255.255
UP LOOPBACK RUNNING MTU:16436 Metric:1
```

3. To configure an Apache 2.0 server for DSR, edit the server configuration file to add a `Listen` directive for the cluster IP (on many systems, the configuration file is found at */usr/local/etc/apache/httpd.conf*). Look for the first line beginning with the `Listen` directive, and add another line that looks like this:

```
Listen cluster-ip
```

Where *cluster-ip* is the DSR-enabled cluster IP. Save your changes to the file.

4. Reboot the Apache server:

```
# apachectl restart
```

You should now be able to send client requests to the cluster IP and port, and get responses directly from the Apache server running on Linux. Remember that static routes on your servers may be necessary, depending on your network configuration.

Configuring a Loopback Interface on Other Systems for DSR

The commands and interfaces used to configure a loopback interface vary slightly between operating systems, and sometimes between versions of the same operating system. Check the documentation for your server operating system for instructions on how to configure a loopback interface. For example, on some BSD systems, the command used in Step 1 in the previous section would be slightly different, as shown below:

```
# ifconfig lo0 cluster-ip netmask cluster-netmask alias
```

Notice that in this case, the netmask used matches the netmask used to configure the cluster on Equalizer, instead of *255.255.255.255* as in the Linux system example.

Loopback Interface Responds to ARP Requests

On some versions of Linux, once you configure the loopback interface using the **ifconfig** command, the system will by default respond to ARP requests for the cluster IP address from remote systems on any network. You can see if this is the case by disabling the cluster on Equalizer and then trying to access the cluster IP address -- if you get a response, then a Linux server behind Equalizer may be responding. You can verify this by examining packet captures on the servers.

This behavior must be disabled when configuring servers for DSR, or there will be network issues.

To configure a server to only respond to the local host, follow the documentation for the server operating system that you are using to set the appropriate ARP parameters.

For example, for a Red Hat 2.6 kernel with two interfaces (*em0* and *em1*), you would set the following parameters to the indicated values in the file */etc/sysctl.conf* and reboot the system:

```
net.ipv4.conf.all.arp_ignore=1
net.ipv4.conf.eth0.arp_ignore=1
net.ipv4.conf.eth1.arp_ignore=1
net.ipv4.conf.all.arp_announce=2
net.ipv4.conf.eth0.arp_announce=2
net.ipv4.conf.eth1.arp_announce=2
```

In the above:

- 1 = reply only if the target IP address is a local address configured on the incoming interface.
- 2 = avoid local addresses that are not in the target's subnet for this interface.

For more information, refer to your server's operating system documentation for the ARP protocol.

Weak and Strong Host Models and DSR

Network interfaces on non-routing systems use either the “weak host” or “strong host” models for packet transmission and reception (these models are defined in RFC1122). In the “strong host” model, a system that is not acting as a router cannot send or receive any packets on a given interface unless the destination/source IP in the packet is assigned to the interface. In the “weak host” model, this restriction does not apply.

In order for DSR to work, the “weak host” model must be enabled on the server's loopback interface, as well as the interface on which requests are received from Equalizer.

Most Linux and Unix systems default to the “weak host” model on all network interfaces, so no additional configuration is usually necessary. For example, on FreeBSD and NetBSD, this behavior is controlled by the setting of `sysctl net.inet.ip.check_interface`, which by default is set to 0 (“weak host”).

Windows XP and Windows 2003 use the “weak host” model on all IPv4 interfaces and the “strong host” model on all IPv6 interfaces, and this is not configurable.

Windows Vista and Windows 2008 support “strong host” by default on all interfaces, but this is configurable for individual interfaces. Use the following command to list interface status:

```
netsh interface [ ipv4 | ipv6 ] show interface
```

The following three command are an example of changing the mode to “weak host” for the LAN and loopback interfaces:

```
netsh interface ipv4 set interface "Local Area Connection" weakhostreceive=enabled
netsh interface ipv4 set interface "Loopback" weakhostreceive=enabled
netsh interface ipv4 set interface "Loopback" weakhostsend=enabled
```

The interface names used in quotes above must match the interface names that appear in the Windows **Network Connections** folder.

Testing Virtual Cluster Configuration

1. After you have configured a virtual cluster and added servers, use a web browser (or just use telnet) to connect to each of the virtual clusters configured on the Equalizer from a system on your network. When you connect to a virtual cluster from the external test machine, Equalizer should send the request to one of the servers configured in the cluster, and you should see the output for that server.
2. From a client machine on the Internet, connect to each virtual cluster using a Web browser.

3. Try to reach Equalizer's Administrative Interface via the internal, external (if configured), and failover (if configured) IP addresses.

For help in resolving configuration problems, see Appendix G, "Troubleshooting" on page 325. Also visit the **Coyote Point Support Portal** (<http://www.coyotepoint.com/support.php>) for more help.

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. Ping Equalizer's external address (if configured) from a host on the external network interface address.
2. Ping Equalizer's internal address from a host on the internal network interface address.
3. If DNS is configured, ping a host on the Internet (e.g., www.coyotepoint.com) from Equalizer to ensure that DNS and the network gateway are functioning properly.
4. 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.
5. 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.
6. 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.)
7. 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.
8. 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 `tracert` (UNIX) or `tracert` (Windows) tools to perform this test.
9. Log into the Administrative Interface on either the external (if configured) or internal interfaces, as described in "Logging In and Navigating the Administrative Interface" on page 52.

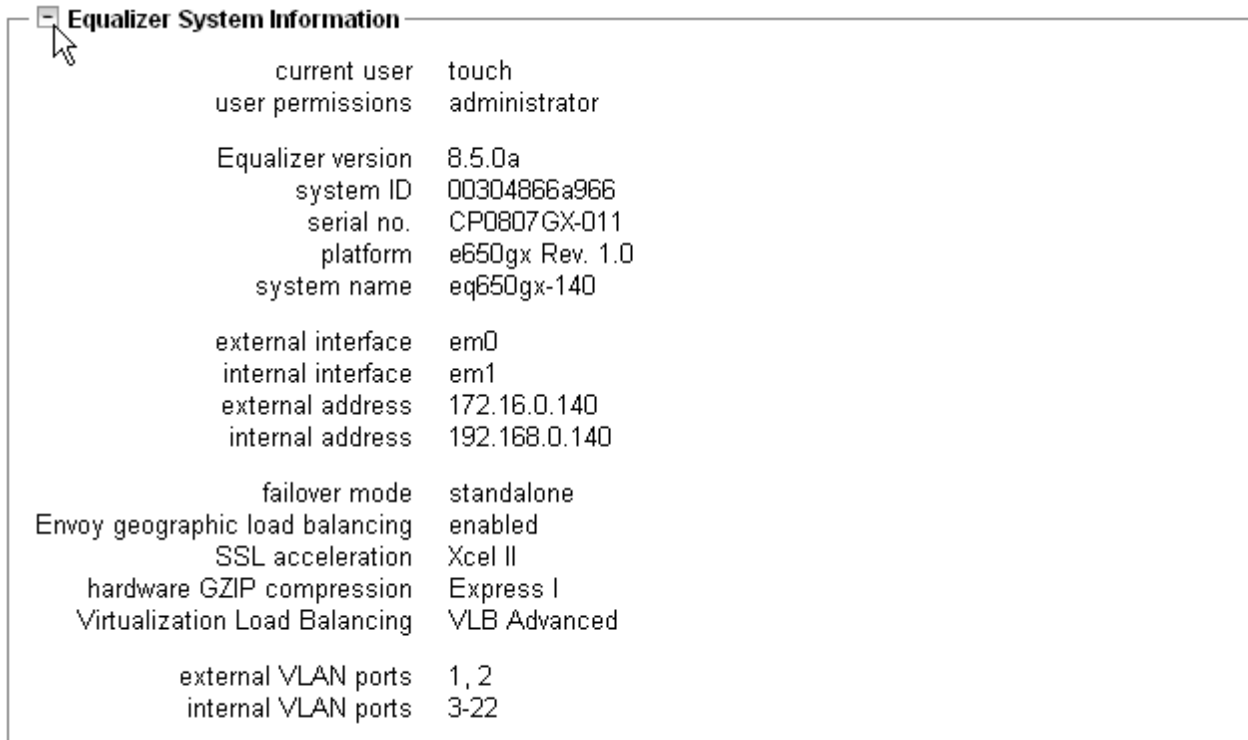


System status information and performance statistics can be gathered and displayed from within the Equalizer Administrative Interface. Equalizer models E350GX and above can also be monitored using standard Simple Network Management Protocol (SNMP) utilities:

Displaying Equalizer System Information	196
Displaying General Cluster Status	197
Displaying the System Event Log	198
Displaying the Virtual Cluster Summary	199
Displaying Global Connection Statistics	201
Displaying Cluster Statistics	203
Displaying Server Statistics	203
Displaying Envoy Statistics	203
Displaying Site Statistics	204
Plotting Global Performance History	205
Plotting Cluster Performance History	205
Plotting Server Performance History	206
Plotting Match Rule Performance History	208
Plotting Responder Performance History	208
Plotting GeoCluster Performance History	209
Plotting Site Performance History	209
Exporting Usage Statistics	210
Configuring Custom Event Handling	213
Forwarding Equalizer Log Information	213
Specifying a Command to Run on an Event	213
Configuring Email Notification	214
Disabling Email Notification	215
Browsing Equalizer Configurations using SNMP	216
Enabling the SNMP Agent	217
Setting Up an SNMP Management Station	218
MIB Description	218
Siblings	219
Configuration and Status	219
Clusters	219
Servers	219
Events	219

Displaying Equalizer System Information

The Equalizer Status screen is displayed when you log into the Administrative interface, and anytime by selecting **Help > About**:



Equalizer System Information	
current user	touch
user permissions	administrator
Equalizer version	8.5.0a
system ID	00304866a966
serial no.	CP0807GX-011
platform	e650gx Rev. 1.0
system name	eq650gx-140
external interface	em0
internal interface	em1
external address	172.16.0.140
internal address	192.168.0.140
failover mode	standalone
Envoy geographic load balancing	enabled
SSL acceleration	Xcel II
hardware GZIP compression	Express I
Virtualization Load Balancing	VLB Advanced
external VLAN ports	1, 2
internal VLAN ports	3-22

Figure 32 Equalizer system information

The Equalizer status screen displays information about Equalizer's operation mode and overall status:

current user	The login name of the currently logged in user. See "Managing Multiple Interface Users" on page 56.
user permissions	The permissions level of the current user . See "Objects and Permissions" on page 57.
Equalizer version	The currently running version of the Equalizer software.
system ID	The unique identifier for the Equalizer unit. [Note: in previous releases, this was shown with a colon (:) separating each pair of numbers.]
serial no.	The hardware serial number. This is the same as the serial number on the tag on the back of Equalizer's metal housing.
platform	The model number and hardware revision of Equalizer.
system name	The hostname assigned to Equalizer (default: equalizer).
external interface	The name of the external interface (as used, for example, in the eqadmin interface).
internal interface	The name of the internal interface.
external address	The IP address assigned to Equalizer's external interface.

internal address	The IP address assigned to Equalizer's internal interface.
failover mode	The current failover mode: standalone (no failover); initializing (the failover subsystem is coming up); primary (the system is the primary failover peer); or, backup (the system is the backup failover peer).
Envoy geographic load balancing	Envoy status: enabled (licensed) or disabled (not licensed).
SSL acceleration	Xcel™ SSL Hardware Acceleration status: enabled or disabled .
hardware GZIP compression	Express™ GZIP Hardware Compression status: enabled or disabled .
Virtualization Load Balancing	Equalizer VLB status: enabled (licensed) or disabled (not licensed).
internal VLAN	List the port numbers of the front panel switch ports currently assigned to the two supported VLANs. See "Managing Interface Ports" on page 75.
external VLAN	

Displaying General Cluster Status

To display a quick view of the status of all clusters and servers defined on Equalizer, click the second item from the top of the left frame object tree; this is either **Equalizer** or, if failover is enabled, the failover peer name of the Equalizer. An example of the **General Cluster Status** table is shown below:

Clusters
Status
Monitoring
Permissions
Maintenance

General
Probes
Networking

Use the icons in the **Actions** column below to add, delete, and modify clusters.
Set global parameters on the **Probes** and **Networking** tabs above.

Server Status:
↑ Up
↓ Down
⏸ Quiesced
🔥 Hot Spare

Name	Type	IP Address	Port	Servers	Actions
tcp-test	tcp_l4	192.168.1.171	80	1 ↑ 0 ↓ 0 ⏸ 0 🔥	
http-test	http	192.168.1.172	80	3 ↑ 0 ↓ 0 ⏸ 0 🔥	
udp-test	udp_l4	192.168.1.173	53	0 ↑ 1 ↓ 0 ⏸ 0 🔥	
https-test	https	192.168.1.174	443	2 ↑ 0 ↓ 0 ⏸ 0 🔥	
http_test_2	http	192.168.1.177	80	5 ↑ 0 ↓ 0 ⏸ 0 🔥	

Figure 33 The general cluster status table

Name	The cluster name.
-------------	-------------------

Type	The cluster type: one of tcp_l4 (Layer 4 TCP), udp_l4 (Layer 4 UDP), http (Layer 7 HTTP), https (Layer 7 HTTPS).
IP Address	The cluster IP address.
Port	The cluster port.
Servers	Status indicators for all servers in the cluster. Shows the number of servers in the following states: Up (responding to health check probes), Down (not responding to health check probes), Quiesced (not accepting new connections), and Hot Spare (only responding to requests when no other server is up).
Actions	Delete or Modify the cluster in the same row as the icon chosen. The Add icon at the bottom of the column opens the Add New Cluster dialog.
reset table width	The columns on the table can be resized. If you extend a column too far to the right so that other columns are no longer visible, this button returns the table to its default proportions.

Displaying the System Event Log

The System Event Log displays start-up, operating system, cluster, and server status messages. You can view the last 20, 50, 100, 200, 500, or 1000 entries in any available sub-type.

1. Select **Equalizer > Status > Event Log** to view the log:

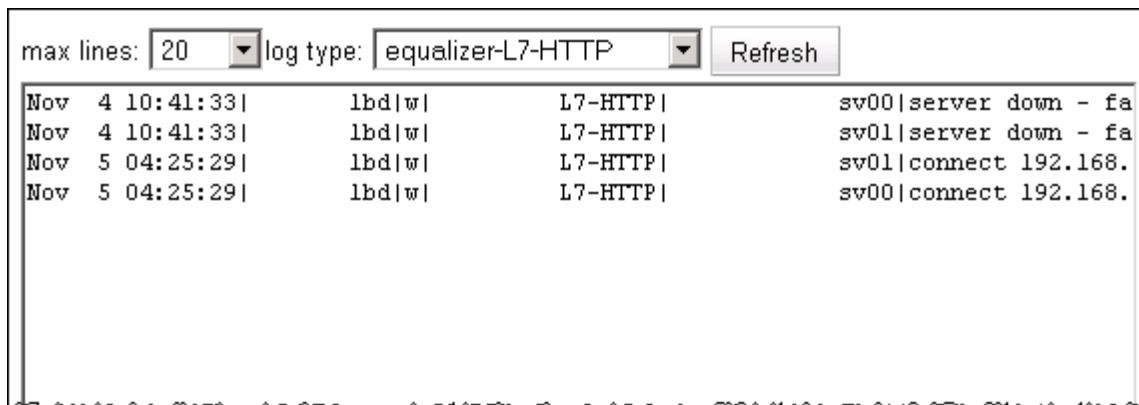


Figure 34 Viewing the system event log

Each log entry has the following general format:

```
time stamp | subsystem | type | cluster | server | event description
```

The type can be one of 'w' (warning), 'n' (notice), or 'e' (error). Use the scrollbar at the bottom of the log to see the entire event description.

2. Select the maximum number of lines to display from the bottom of the log in the **max lines** select box.
3. Select the type of messages to display in the **log type** select box: **equalizer** displays the Equalizer software log; **operating system** displays the log for Equalizer's host operating system; other entries display log entries for the appropriate cluster only.
4. Select the **Refresh** button to display the selected log entries.

To export the contents of a log, 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

Select **Equalizer > Status > Cluster Summary** to open the Virtual Cluster Summary. This table displays basic status and statistics for the currently configured virtual clusters, their associated servers, and Layer 7 match rules, as shown in the example below:

L7-HTTP-0					
Servers	Status	InitialWeight	CurrentWeight	Processed	Active
sv00	↑	100	60	454922	433
sv01	↑	100	60	454063	478
Match rules			Processed		
Default			9084823		
L7-HTTPS +					
L7-HTTP-AB +					
L7-HTTP-1 +					
L7-HTTP-2 +					
L7-HTTP-3 +					
L7-HTTP-4 +					

Figure 35 Viewing cluster summary information

Click on a cluster name to open the summary for that cluster. For each server in a cluster, the table displays the following information:

Servers	The server name.
Status	Status indicators for each server in the cluster: Up (responding to health check probes), Down (not responding to health check probes), Quiesced (not accepting new connections), and Hot Spare (only responding to requests when no server is marked Up).
Initial Weight	The initial server weight assigned to the server by the administrator. This weight is used by Equalizer as it starts to load balance requests amongst the servers in the cluster. For all load balancing policies other than static weight and round robin , Equalizer adjusts the server initial weights to reflect the relative performance of the servers in the cluster over time. The dynamically adjusted server weight is displayed as Current Weight in the table.
Current Weight	The dynamically adjusted server weight used by equalizer when load balancing incoming requests (see Initial Weight, above). If the load balancing policy for the cluster is static weight , then this column shows the Initial Weight value. If the policy is round robin , then “ Round Robin ” is displayed instead of a value.
Processed	The total number of connections that have been processed by the server since the system was rebooted.
Active	The total number of currently active connections to this server.

Sticky	(Layer 4 clusters with a non-zero sticky time only): The number of inactive “sticky records” currently held by Equalizer. This equals the number of sticky records minus the number of Active connections (see above). See “Enabling Sticky Connections” on page 139.
---------------	---

For each match rule in a Layer 7 cluster, the summary displays the following information:

Processed	The number of requests that have been <i>evaluated</i> by the Match Rule since the system was last rebooted. This number includes both requests that matched and did not match the match rule expression.
------------------	---

If Envoy is enabled, GeoCluster names are also listed in the Cluster Summary table. For each site in a GeoCluster, the summary displays the following information:

Site	The site name.
Status	Status indicators for each site in the GeoCluster: Up (responding to health check probes), Down (not responding to health check probes), Quiesced (not accepting new connections), and Hot Spare (only responding to requests when no site is marked Up).
Weight	The site weights determine the relative proportion of connection requests that Equalizer routes to each site. These weights are the current, dynamically-adjusted values, not the initial 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.

Displaying Global Connection Statistics

Click on the plus sign (+) next to **Connections** in the left frame to display the following statistics:

L4 processed	The total number of Layer 4 connections processed since the last reboot. These are connections that have been opened and data has passed over the connection.
L4 peak	The peak (highest) number of Layer 4 connections processed <i>per second</i> since the last reboot.
L4 timeouts	The total number of partially established Layer 4 connections that were closed because either the idle timeout or stale timeout timers expired.
L7 active	The total number of currently active Layer 7 connections. This is the number of established open connections, regardless of whether any data has passed over the connections.
L7 processed	The total number of Layer 7 connections processed since the last reboot. This is the number of established connections over which data has passed.
L7 peak	The peak (highest) number of Layer 7 connections processed <i>per second</i> since the last reboot.

Click on **Connections** in the left frame, or select **Equalizer > Status > Statistics**, to display the following global connection statistics. All statistics are reset when the system reboots.

Basic Statistics	
L4 total connections processed	The total number of Layer 4 connections processed since the last reboot. These are connections that have been opened and data has passed over the connection.
L4 peak connections processed	The peak number of Layer 4 connections processed (in connections per second).
L4 stale connections timed-out	The total number of partially-established Layer 4 connections that were closed because the stale timeout for the connection expired. See "Global Networking Parameters" on page 91.
L4 idle connections timed-out	The total number of partially-established Layer 4 connections that were closed because the idle timeout for the connection expired. See "Global Networking Parameters" on page 91.
L7 current active connections	The total number of currently active Layer 7 connections. Includes partially established connections (client connections that have not yet been load balanced to a server).
L7 total connections processed	The total number of Layer 7 connections processed.
L7 peak connections processed	The peak number of Layer 7 connections processed (in connections per second).
Advanced Statistics	
L7 client connections acceptable	The number of Layer 7 client connections that were initiated.

L7 connections timed out	The number of Layer 7 connections that timed out because one of the connection timers (client timeout , connect timeout , or server timeout) expired.
L7 request bytes from clients	The number of bytes received in client requests.
L7 response bytes to clients	The number of bytes received in server responses.
L7 complete requests	The number of Layer 7 client requests that were completed (i.e., all headers were received before client timeout expired).
L7 min. usec to complete request	The minimum number of microseconds required to receive a complete client request.
L7 max. usec to complete request	The maximum number of microseconds required to receive a complete client request.
L7 avg. usec to complete request	The average number of microseconds required to receive a complete client request.
L7 maximum headers exceeded by client	The number of times a request was received that contained more than the maximum of 64 headers supported by Equalizer (connections that exceed 64 headers are dropped by Equalizer).
L7 total client connections	The total number of Layer 7 clients connections received (not necessarily processed).
L7 current client connections	The number of currently active client connections.
L7 requests processed	The total number of Layer 7 clients requests processed.
L7 responses processed	The total number of Layer 7 server responses processed.
L7 server conx reused	The number of times a server connection was kept open and re-used by Equalizer.
L7 cookies stuffed	The number of times Equalizer inserted a cookie into a Layer 7 packet.
requests in error	Number of requests that caused an error.
L7 responses in error	Number of Layer 7 responses that caused an error.
L7 client request timeouts	Number of Layer 7 requests that were dropped because the client timeout expired.
L7 server connect timeouts	Number of Layer 7 requests that were dropped because the connect timeout expired.
server response timeouts	Number of Layer 7 requests that were dropped because the server timeout expired.
L7 avg. usec to connect to server	The average number of seconds that Equalizer had to wait for a connection to a server.
L7 http compressed response count¹	The total number of server responses compressed.
L7 http compressed current responses count¹	The number of server responses currently being compressed.

L7 http bytes selected for compression¹	The total number of input bytes from all server responses that were selected for compression.
L7 http compressed bytes output¹	The total number of compressed bytes output from all server responses.
L7 http compression ratio¹	The approximate current compression ratio (bytes selected for compression divided by the compressed bytes output).

¹ Note that compression statistics are only displayed if Express Hardware GZIP Compression is installed in Equalizer.

Displaying Cluster Statistics

To display statistics for a cluster, click on the cluster name in the left frame object tree, and then select the **Reporting > Statistics** tab in the right frame. The following statistics are displayed:

total number of servers	The number of servers defined for the cluster.
server active connections	The number of active (current) connections to this cluster.
total connections served	The total number of connections to this cluster since the last reboot.
time since last activity	The number of seconds since the last connection to this cluster.

Displaying Server Statistics

To display statistics for a server, click on the server name in the left frame object tree, and then select the **Reporting > Statistics** tab in the right frame. The following statistics are displayed:

server dynamic weight	The current dynamic weight for this server.
server active connections	The number of active (current) connections to this server.
total connections served	The total number of connections to this server since the last reboot.
time since last activity	The number of seconds since the last connection to this server.

Displaying Envoy Statistics

To display Envoy statistics, click on **Envoy** in the left frame object tree, and then open the **Status > Statistics** tab in the right frame. The following statistics are displayed:

DNS Requests received	The total number of DNS requests received.
Invalid DNS Requests received	The total number of Invalid DNS requests received.
Geocluster not found	The total number of DNS requests received that contained a GeoCluster name not defined on the local Equalizer.

Displaying Site Statistics

To display statistics for a Site in a GeoCluster, click on the Site name in the left frame object tree, and then select the **Reporting > Statistics** tab in the right frame. The following statistics are displayed:

total requests	The number of requests directed to this Site since the last reboot.
number queued	The number of requests queued for this site.
timed out	The number of agent-to-client triangulation probes that timed out before Equalizer received a response.
site had zero weight	The number of times the server was chosen but had a zero weight.
agent retries	The number of probes Equalizer re-sent to its agent.
agent misses	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.
agent errors	The number of Equalizer-to-agent probes that returned a resource-unavailable error -- that is, Envoy on the remote site determined that the requested resource is unavailable.
unavailable	The number of times the server was chosen but was unavailable.
site returned	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 default	The number of clients directed to the default site.
resource performance	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.

Plotting Global Performance History

1. Click on **Equalizer** (or the configured *Failover Peer Name* for this Equalizer) in the left frame, and open the **Status > Plots** tab in the right frame.
2. Select one or more of the following statistics to plot (all statistics are reset on reboot):

CPU Utilization	The average percent of non-idle CPU time over the selected time period.
Memory Utilization	The average percent of in-use memory over the selected time period.
L4 Connections Timed Out	The number of Layer 4 connections that were closed because a connection timeout expired (see Appendix B, "Timeout Configuration").
L7 Connections Timed Out	The number of Layer 7 connections that were closed because a connection timeout expired (see Appendix B, "Timeout Configuration").
L4 Total Connections Processed	The number of client connections to Layer 4 clusters processed.
L7 Total Connections Processed	The number of client connections to Layer 7 clusters processed.

3. Use the slider controls to select the following:

Refresh Rate	The amount of time between updates of the plot data.
Duration	The time interval displayed in the plot. Sets the horizontal time scale for the plot. For example, if 5 mins is selected, all the data collected over the last 5 minutes is displayed in the plot.

4. The plot display is updated automatically with your settings the next time the display is refreshed.

Plotting Cluster Performance History

To display a graphical representation of the performance history of a cluster:

1. Click on the cluster name in the left frame object tree, and then select the **Reporting > Plots** tab in the right frame.
2. Select one or more of the following statistics to plot (all statistics are reset on reboot):

Servers	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	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. Initialized to 0 when the system boots. By design, when a server with a non-zero service time transitions to having no active connections for some period of time, Equalizer stops adjusting the service time and continues using the last service time value in load balancing decisions. This is why sometimes the service time is not equal to 0 when there are no active connections to the server.
Active Connections	The total number of active connections on the servers in the cluster.
Hit Rate	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	The average of the server agent return values for all servers in the cluster. If you have not configured server agents, -2 is displayed.

3. Use the slider controls to select the following:

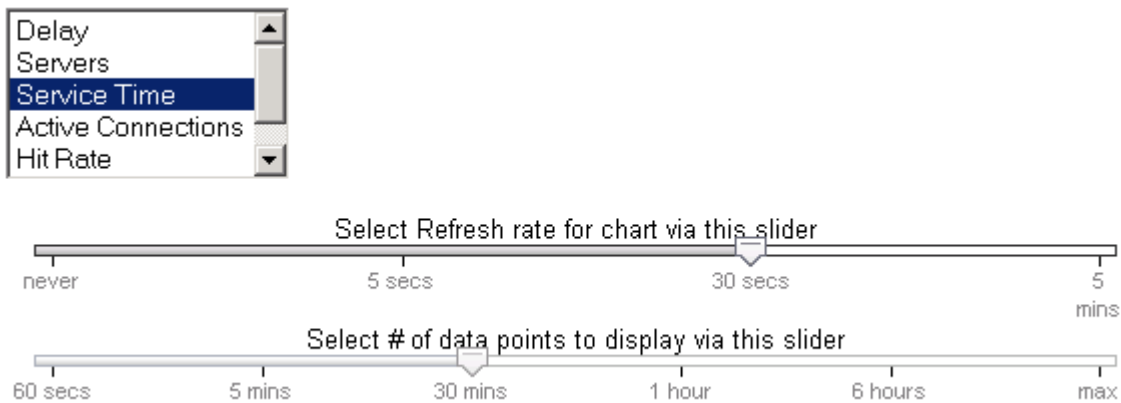
Refresh Rate	The amount of time between updates of the plot data.
Duration	The time interval displayed in the plot. Sets the horizontal time scale for the plot. For example, if 5 mins is selected, all the data collected over the last 5 minutes is displayed in the plot.

4. The plot display is updated automatically with your settings the next time the display is refreshed.

Plotting Server Performance History

To display a graphical representation of the performance history of a server:

1. Click on the server name in the left frame object tree, and then select the **Reporting > Plots** tab in the right frame.
2. Scroll down using the scrollbar at the right of the plot screen to display the plot controls:



3. In the drop down box, use the **Ctrl** or **Shift** keys and the left mouse button to select one or more of the following statistics to plot:

Active Connections	The number of active connections on the server. Equalizer “smooths” the connection count using a sliding-window smoothing algorithm before being plotted.
Service Time	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 (such as / cgi-bin pages), 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. Initialized to 0 when the system boots. By design, when a server with a non-zero service time transitions to having no active connections for some period of time, Equalizer stops adjusting the service time and continues using the last service time value in load balancing decisions. This is why sometimes the service time is not equal to 0 when there are no active connections to the server.
Computed Load	<p>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.</p> <p>Equalizer derives a server’s computed load value from its service time, number of active connections, and server agent value (if configured). It also takes into account the load balancing policy used by the cluster.</p> <p>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.</p>
Dynamic Weight	<p>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.</p> <p>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.</p> <p>If you are not using adaptive load balancing (for example, the load balancing policy is set to <i>round robin</i> or <i>static weight</i>), 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 137.</p>

Server Agent	<p>The value that the server agent daemon returns. When queried, the server agent returns a value in the range -2 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 displayed is -2.</p> <p>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.</p>
---------------------	---

- Use the slider controls to select the following:

Refresh Rate	The amount of time between updates of the plot data.
Duration	The time interval displayed in the plot. Sets the horizontal time scale for the plot. For example, if 5 mins is selected, all the data collected over the last 5 minutes is displayed in the plot.

- The plot display is updated automatically with your settings the next time the display is refreshed.

Plotting Match Rule Performance History

- Click on the server name in the left frame object tree, and then select the **Reporting > Plots** tab in the right frame. The number of **Processed Connections** is the number of connections selected by the conditions of the match rule.
- Use the slider controls to select the following:

Refresh Rate	The amount of time between updates of the plot data.
Duration	The time interval displayed in the plot. Sets the horizontal time scale for the plot. For example, if 5 mins is selected, all the data collected over the last 5 minutes is displayed in the plot.

- The plot display is updated automatically with your settings the next time the display is refreshed.

Plotting Responder Performance History

- Do one of the following:
 - Click on the Responder name in the left frame and open the **Reporting** tab in the right frame.
 - Click **Responders** in the left frame and then click on the **Edit** icon in the **Action** column of the table, on the same row as the name of the Responder whose statistics you want to view.
- The chart shows the number of **Processed Connections** for the Responder -- this is the number of times the Responder was executed by a match rule. This counter is incremented each time a Responder is executed by any match rule in any cluster.
- Use the slider controls to select the following:



Refresh Rate	The amount of time between updates of the plot data.
---------------------	--

Duration	The time interval displayed in the plot. Sets the horizontal time scale for the plot. For example, if 5 mins is selected, all the data collected over the last 5 minutes is displayed in the plot.
-----------------	---

- The plot display is updated automatically with your settings the next time the display is refreshed.

Plotting GeoCluster Performance History

If you have licensed Envoy on your Equalizer, you can use the Plot feature to view a graphical representation of the performance history for the selected GeoCluster. To plot the performance history for a geographic cluster, follow these steps:

- In the left frame, right-click the name of the geographic cluster whose history you want to view, and select **Plot GeoCluster** from the menu. The graphical history for the selected cluster appears in the right frame.
- To change the information being plotted, scroll down using the scrollbar at the right of the plot screen to display the plot controls.
- Choose the statistics to plot from the drop down box:

sites	The number of requests received for all sites in the GeoCluster since the last reboot.
network latency	The average ICMP triangulation time (if ICMP triangulation is enabled) when at least one site was able to respond. This value does not include clients for which the default site was selected.
global request rate	The number of requests received for the cluster per minute.

- Use the slider controls to select the following:

Refresh Rate	The amount of time between updates of the plot data.
Duration	The time interval displayed in the plot. Sets the horizontal time scale for the plot. For example, if 5 mins is selected, all the data collected over the last 5 minutes is displayed in the plot.

- The plot display is updated automatically with your settings the next time the display is refreshed.

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:

- In the left frame, right-click the name of the site whose history you want to view, and select **Plot Site** from the menu. The graphical history for the selected cluster appears in the right frame.
- To change the information being plotted, scroll down using the scrollbar at the right of the plot screen to display the plot controls.
- Choose the statistics to plot from the drop down box:

Site Chosen	The number of times that Equalizer returned this site in response to a client query.
--------------------	--

No Agent Response	The number of requests in which an agent failed to reply to Equalizer's probes.
Resource Down	The number of times that the target resource failed to respond during the period plotted.
Default Chosen	The number of times the default site was chosen in response to a client query.
Agent Errors	The number of ICMP ECHO requests that the agent at this site sent to clients and for which the agent received no response.
Resource Performance	The relative workload of this site during the plotted period.

4. Use the slider controls to select the following:

Refresh Rate	The amount of time between updates of the plot data.
Duration	The time interval displayed in the plot. Sets the horizontal time scale for the plot. For example, if 5 mins is selected, all the data collected over the last 5 minutes is displayed in the plot.

5. The plot display is updated automatically with your settings the next time the display is refreshed.

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:

1. Select **Equalizer > Monitoring > Export to CSV**.
2. Select the **time period** for which you want to export the data from the drop-down box.
3. Select **export** to download the file for saving via your browser.

The amount of data (and hence the size of the export file) is limited by the number of clusters, servers, and match rules in your configuration as well as available disk space. Each cluster, server, and match rule occupies one column for each of the statistics shown in the table below. If you have a large number of columns, this will limit the number of data samples (rows) that can be stored on disk for export. The following statistics are reported in the exported file, with one row for every five seconds in the selected **time period**:

For each Cluster:	
Delay	The average service time of all of the active servers in the cluster, in milliseconds. The service time is the time it takes a server to start sending reply packets once it receives a client request. Initialized to 0 when the system boots. By design, when a server with a non-zero delay transitions to having no active connections for some period of time, Equalizer stops adjusting the delay and continues using the last delay value in load balancing decisions. This is why sometimes the delay is not equal to 0 when there are no active connections to the server.

Agent	The average of the server agent values returned for all servers in the cluster.
Connections	The average number of active connections for all servers in the cluster.
Load	A relative value that can range between 0 and 100 times the number of servers in the cluster. The cluster load is calculated by adding together the current server load values for all servers in the cluster, and dividing by the number of currently active servers. It is initialized to 0 when the system boots and is updated once the cluster starts accepting client connections. It can be used only to gauge the relative current load of the cluster overall. For example, if the cluster load is 350 and there are 7 servers in the cluster, then the cluster is currently operating at roughly half its total capacity.
For each L7 Match Rule:	
Smoothed Processed Connections	The total number of incoming requests that were examined and matched the match rule expression.
For each Server:	
Delay	The average service time of the server in milliseconds. The service time is the time it takes a server to start sending reply packets once it receives a client request. Initialized to 0 when the system boots. By design, when a server with a non-zero delay transitions to having no active connections for some period of time, Equalizer stops adjusting the delay and continues using the last delay value in load balancing decisions. This is why sometimes the delay is not equal to 0 when there are no active connections to the server.
Agent	The average of the server agent values returned for the server.
Connections	The number of active connections for the server.

<p>Load</p>	<p>The computed load for the server. Server load as calculated by Equalizer is a measure of the request load on this server <i>relative to the other servers in the cluster</i>. The server load is a number between 0 and [100 times the number of servers in the cluster]. So, for example, if there are 7 servers in a cluster, each server can have a load value between 0 and 700.</p> <p>Server load is initialized to the server's initial weight when the system boots.</p> <p>Equalizer's server and cluster load calculations are primarily for internal use, indicate relative processing power, and are not designed to indicate an absolute load (i.e., they are not percentages).</p> <p>Server load values are used in load balancing decisions, such that a server with a load of 250 would be twice as likely to be chosen to receive a new incoming request as a server with a load value of 500.</p> <p>By design, when a server with a non-zero load transitions to having no active connections for some period of time, Equalizer stops adjusting the load and continues using the last load value in load balancing decisions. This is why sometimes the server load is not equal to 0 when there are no active connections to the server.</p>
<p>Total</p>	<p>The total number of connections processed by the server.</p>
<p>Time</p>	<p>The amount of time (in milliseconds) that the server spent processing client requests (since the last reboot).</p>
<p>Weight</p>	<p>The server's dynamic weight.</p>
<p>Global Statistics:</p>	
<p>Total Connections Processed</p>	<p>The total number of connections processed.</p>
<p>Peak Connections Processed</p>	<p>The peak number of connections per second processed.</p>
<p>Connections over last sec.</p>	<p>The number of connections over the last second.</p>
<p>Connections Timed Out</p>	<p>The number of connections that were dropped because one of the connection timeout counters expired.</p>
<p>CPU Utilization</p>	<p>A number indicating the percent of available CPU capacity being used.</p>

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 log entries from Equalizer’s System Event Log (see “Displaying the System Event Log” on page 198), to another machine that is running a system logging daemon. When this option is enabled, each system event message is sent to the remote system via a UDP datagram by the **syslogd** daemon running on Equalizer. To specify a remote system logging host, follow these steps:

1. Log into the Equalizer Administration Interface (see “Logging In” on page 52).
2. Select **Equalizer > Monitoring > Events**.

Figure 36 The Events tab - logging field

3. In the **logging** field, enable the **use remote syslog** checkbox.
4. In the **syslog host** text box, type the hostname or IP address of the machine to which you want to forward **syslog** messages. The system you specify must be running a system logging daemon (such as **syslogd**) that is configured as a system logging host; see the documentation for the operating system running on that system for more information.
5. Click the **commit** button.

Note – The remote **syslog** facility on Equalizer sends all **syslog** messages to the remote host, including low priority messages (e.g., information only messages) that are not displayed when viewing the operating system log via the Administrative Interface. You should edit the **syslog** configuration on the remote host to filter incoming messages so that only the priority levels in which you are interested are recorded on the remote system.

Specifying a Command to Run on an Event

You can configure Equalizer to run a command that you specify (such as running a custom shell script) whenever certain 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

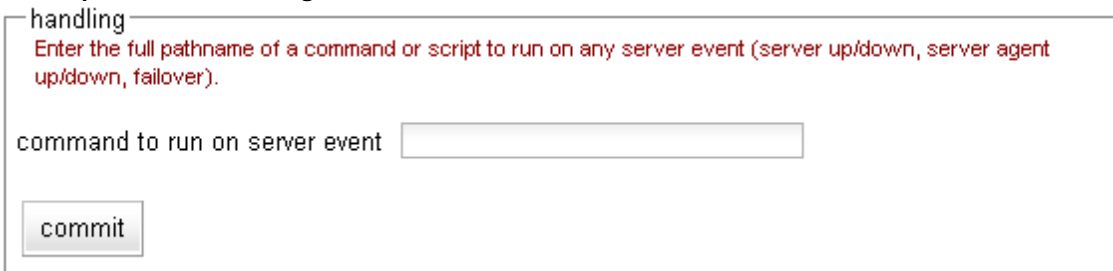
When an event command is configured and one of the above events occurs, the command is executed and a one-line message describing the event that occurred is sent to the standard input of the specified command. This message can then be read and examined by the command to which it is passed. It is the same message that is sent via email notification for such events.

For example, the following shell script will append the current date and the event message to a file:

```
#!/bin/sh
read MSG
echo `date`: $MSG >> /tmp/echomsgs.txt
```

Once the above shell script is installed as (for example) `/usr/bin/local/echomsgs` on Equalizer, you can then tell Equalizer to run the script by doing the following:

1. Log into the Equalizer Administration Interface (see “Logging In” on page 52).
2. Select **Equalizer > Monitoring > Events**



The screenshot shows a web form for configuring server event handling. At the top, the word "handling" is displayed. Below it, a red instruction reads: "Enter the full pathname of a command or script to run on any server event (server up/down, server agent up/down, failover)". A text input field is labeled "command to run on server event" and is currently empty. Below the input field is a "commit" button.

3. In the **handling** field, enter the command that you want Equalizer to run when it detects a server event. For our example above, you would enter:

`/usr/local/bin/echomsgs`

4. Click the **commit** button.

Note – Any program specified 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

You can configure Equalizer to send an email notification whenever a server event occurs, for the same list of events shown in the previous section. 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. The procedure below shows you how to use the **event notification** screen to configure, enable, and disable email notification.

1. Log into the Equalizer Administration Interface (see “Logging In” on page 52).
2. Select **Equalizer > Monitoring > Events**

email notification

Enter from and to addresses in "user@example.com" or "<user@example.com>" format. The SMTP server can be specified as an IP address or hostname. Enable the check box to send email on any server event.

enable email notification

from

to

SMTP server

3. In the **email 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** (for example: **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. At least one recipient address must be specified. Specify multiple email addresses by including them in angle brackets, separated by whitespace, as in the following example:


```
<recipient1@example.com> <recipient1@example.com> <recipient1@example.com>
```
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 did not check the **enable email notification** check box, you will not be able to commit the changes.)

Disabling Email Notification

To disable email notification:

1. Log into the Equalizer Administration Interface (see “Logging In” on page 52).
2. Select **Equalizer > Monitoring > Events**. On the **event configuration screen**, clear the **enable email notification** checkbox.
3. Select **commit**.

Browsing Equalizer Configurations using SNMP

SNMP
is not
supported
on E250GX
model
Equalizers

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 E350GX 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. Log into the Equalizer Administration Interface (see “Logging In” on page 52).
2. Select **Equalizer > Monitoring > SNMP:**

SNMP agent configuration

Set values below to be used by the SNMP agent, and enable the check box to run the agent. There are two MIB files to import into your MIB browser: the [Registrations MIB](#) and the [Equalizer MIB](#).

Enable SNMP Agent	<input checked="" type="checkbox"/>
system description	<input type="text" value="Equalizer"/>
system location	<input type="text" value="location"/>
system contact	<input type="text" value="contact"/>
system name	<input type="text" value="equalizer"/>
community string	<input type="text" value="public"/>
agent IP address	<input type="text"/>

Enable SNMP traps by setting an IP address and optional port (default 162) to receive the traps. Enable the check boxes next to the events that will generate traps.

trap IP address:port	<input type="text"/>
Enable server up/down events	<input checked="" type="checkbox"/>
Enable peer events	<input checked="" type="checkbox"/>
Enable failover events	<input checked="" type="checkbox"/>
Enable partition events	<input checked="" type="checkbox"/>

Figure 37 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. By default, the SNMP agent will listen on Equalizer’s VLAN IP address on all VLANs. It will, by default, always respond using the Default VLAN IP address as the source address. To configure the SNMP agent to listen and respond on a particular IP address, enter the address in the **agent IP address** text box. This feature would most commonly be used to have the agent listen and respond on a VLAN’s **Failover IP** address. [Note that the port used by the SNMP agent is always UDP port 161 and is not configurable.]
6. 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.

- 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, <code>cpsSysEqServerDownEv</code> and <code>cpsSysEqServerUpEv</code> . Equalizer triggers these traps when it detects either a server failure or a response from a failed server.
Enable peer events	This checkbox controls two traps, <code>cpsSysEqSiblingContactLostEv</code> and <code>cpsSysEqSiblingContactOkayEv</code> . Equalizer triggers these traps whenever it is configured as part of a failover pair and it either loses or regains contact (respectively) with its peer.
Enable failover events	This checkbox controls one trap, <code>cpsSysEqAssumedPrimaryRoleEv</code> . Equalizer sends this trap whenever it assumes primary status.
Enable partition events	This checkbox controls one trap, <code>cpsSysEqPartitionDetectedEv</code> . Equalizer sends this trap whenever it is in failover mode and detects that both Equalizers have assumed primary status.

- 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.
- 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.

Note that Equalizer's MIB does *not* contain MIB objects for **system**, **interface**, and many other "standard" MIB object trees common to many SNMP-enabled devices. As a result, any management station or other SNMP-based

software that queries for them will return an error. Only the objects defined in the *cpsreq.my* and *cpsequal.my* MIB definition files are supported by Equalizer.

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 its 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.



Match Rules
are not
supported
on E250GX
model
Equalizers

This chapter tells you all you need to know to create Layer 7 Match Rules that load balance requests based on the content in the payload of the requests, as well as the header information and other request characteristics.

Why Match Rules?	222
Match Rules Overview	222
Match Rule Processing	223
Match Rule Order	224
Match Rules, the Once Only Flag, and Cookies	225
General Match Expressions and Match Bodies	226
Match Expressions	226
Match Bodies	228
Match Rule Definitions	228
Managing Match Rules	229
The Match Rules Table	230
The Default Match Rule	230
Creating a New Match Rule	231
Modifying a Match Rule	235
Removing a Match Rule	235
Match Functions	235
Match Function Notes	239
Match Rule Behavior When Server Status is not 'Up'	239
Considering Case in String Comparisons	240
Regular Expressions	240
Supported Headers	240
HTTPS Protocol Matching	241
Supported Characters in URIs	241
Logical Operators and Constructs in the GUI	241
Using Responders in Match Rules	242
Example Match Rules	242
Parsing the URI Using Match Rules	243
Changing Persistence Settings Using Match Rules	244
Changing the Spoof (SNAT) Setting Using Match Rules	246
Selective SNAT Example	246
Server Selection Based on Content Type Using Match Rules	249
Using the Custom Load Balancing Policy with Match Rules	251

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 algorithms 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.

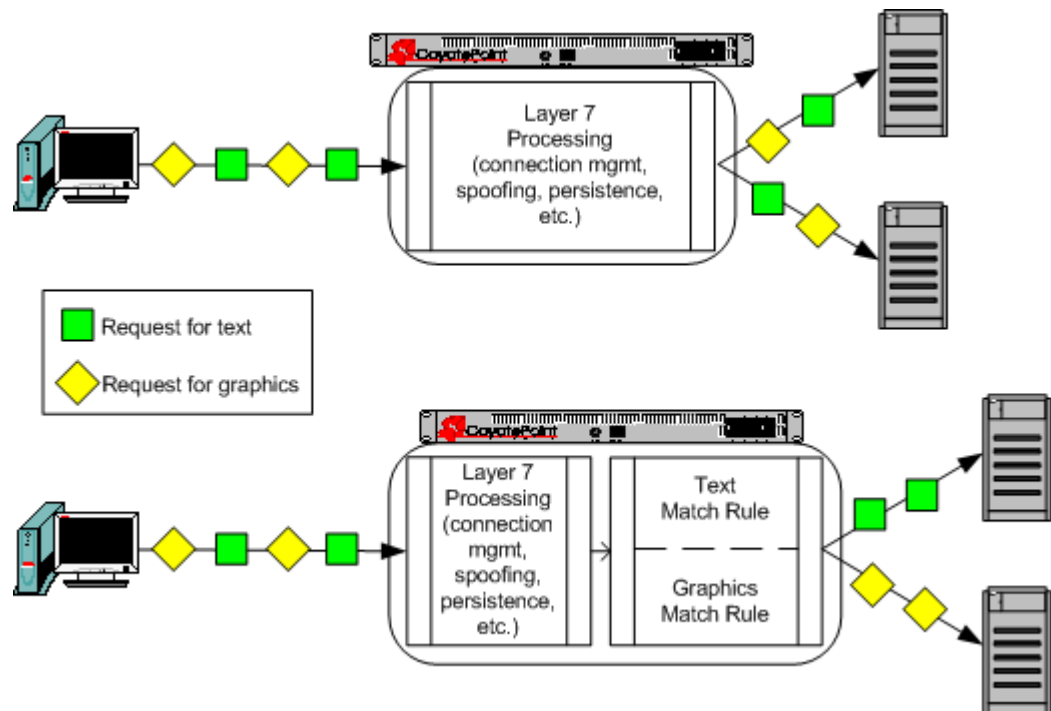


Figure 38 Conceptual Example of Match Rule Processing

Most client requests are a mix of requests for text and graphics. Layer 7 processing without Match Rules (top diagram in Figure 38) 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 38), 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 -- that is, the match rule expression evaluates to *true* -- no further match rules are checked against the request.

Equalizer makes a load balancing decision as follows:

1. If the request headers contain a cookie that specifies a server in the match rule's server list, Equalizer sends the request to the server in the cookie.
2. Otherwise, Equalizer sends the request to the server in the match rule's server list that is selected by the load balancing policy in effect for the match rule.

This process applies even if all the servers selected for the match rule are unavailable. In this case, when the match rule expression matches the request and all the servers in the match rule server list are unavailable, no reply is sent to the client. Eventually, the client sees a connection timeout.

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.

Match Rule Order

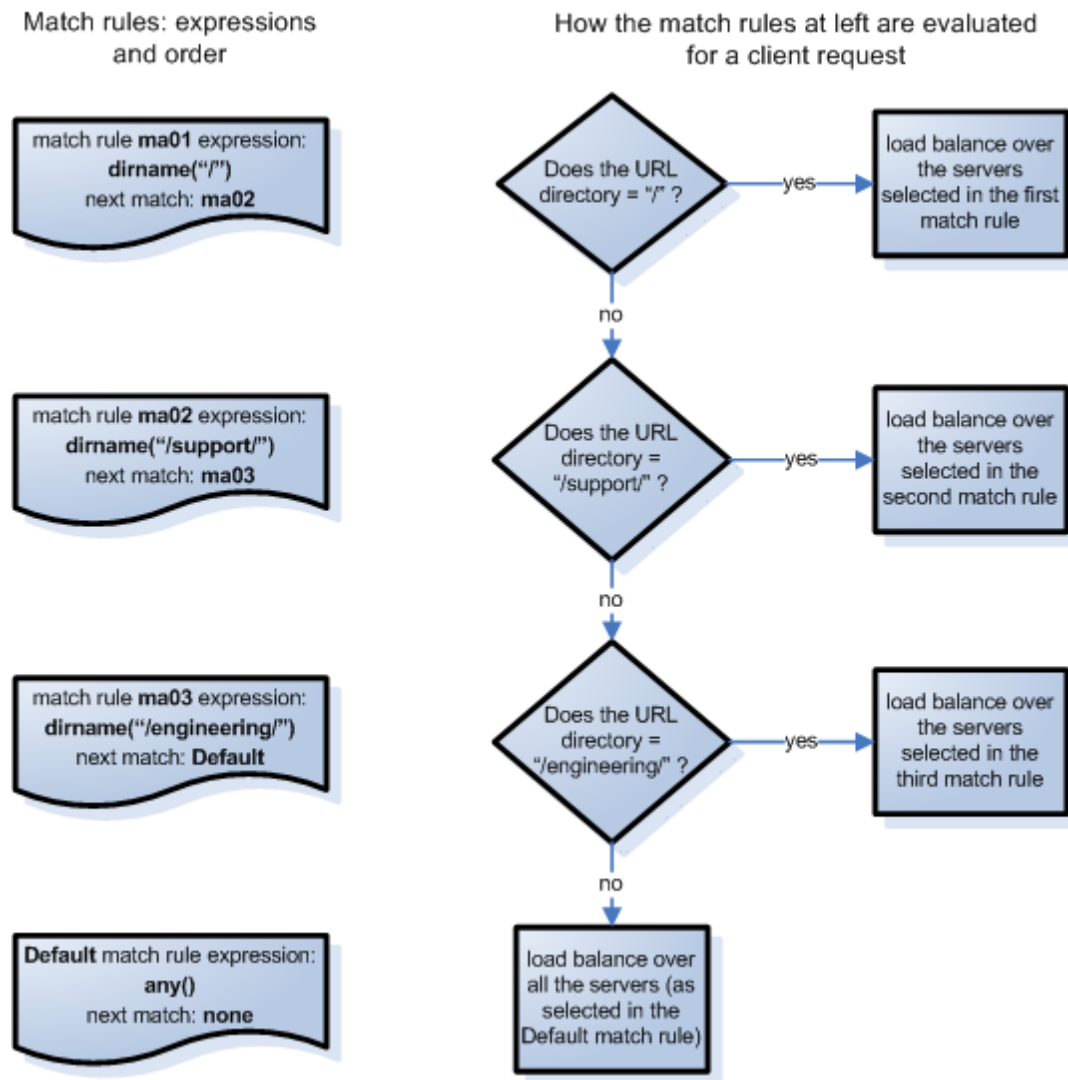
When you add more than one match rule to a cluster, the order in which the match rules are processed is important to system performance. Since processing a match rule requires system CPU and memory, the most efficient way of ordering match rules is from the most common case to the least common case. In this way, you ensure that the greatest number of client connections possible will process the first match rule and, if it matches the request, stop processing match rules for that request.

In other words, the goal is to load balance the highest possible number of requests according to the settings in the first match rule, which has the effect of reducing to a minimum the amount of match rule processing required for requests to that cluster.

This is best illustrated by an example. Let's say you want to construct a set of match rules that achieves these goals:

- Direct all requests whose URL contains one of two specific directories to specific servers. Assume these two directories are */support* and */engineering*.
- Of the two directories above, we expect more requests to contain */support*.
- Load balance requests whose URL does *not* contain a directory across all servers.
- We want to process requests that do *not* contain a directory the fastest, since we expect that 75% of requests to this cluster will NOT contain a directory in the URL.

The set of match rules that achieves this, their order, and how the match rules are evaluated, is described in the following figure.



At left in the figure above are the expressions for the three match rules, shown in the order in which they are configured in the cluster. At right, the decision tree describes how the match rules are evaluated for every client request that comes into this cluster.

As described previously, the first match rule (**ma01**) is meant to match any request that does not have a directory in it. Since this is our most common case, match rule evaluation will stop after the first match rule is evaluated for the majority of incoming requests.

The second and third rules, **ma02** and **ma03**, match for specific directory names. We match for the most common directory name first, then the less common directory name.

Finally, if all three of the match rule expressions for **ma01**, **ma02**, and **ma03** fail to match an incoming request, then that request is load balanced across all servers in the cluster using the options set on the cluster (and mirrored in the **Default** match rule).

Match Rules, the Once Only Flag, and Cookies

Since multiple client requests may be received on a single TCP/IP connection, Equalizer has a flag (**once only**) that specifies whether to check the headers in every request received on a connection, or to load balance based solely

upon the first set of headers received on a connection (and ignore the headers in subsequent requests on the same connection).

The **once only** flag is a cluster parameter on the **Networking** tab. When using Match Rules, it is usually desirable to turn *off* the **once only** flag for the cluster so that Equalizer matches against each individual request in a connection, not just the first one.

You can also enable or disable **once only** in a match rule, to override the setting on the cluster for any request that matches that rule. For example, if **once only** is enabled on a cluster and disabled on a match rule, any request that matches that match rule's expression will be load balanced as if **once only** were disabled on the cluster.

The following table shows how the setting of **once only** affects load balancing when a match rule hit occurs:

match rule hit on...	once only disabled	once only enabled
...the first request on a connection	<p>If the request headers contain a cookie specifying a server in the match rule's server list, send the request to the server in the cookie.</p> <p>Otherwise, send the request to the server in the match rule's server list that is selected by the load balancing policy in effect for the match rule.</p>	Same as at left.
...second and subsequent requests on the same connection	Same as above.	<p>If the request headers contain a cookie specifying a server in the match rule's server list, send the request to the server in the cookie.</p> <p>Otherwise, send the request to the server that was selected by the first request.</p>

Note that Equalizer always honors a cookie that specifies a server in the match rule's server list, regardless of the setting of the **once only** flag: the request is sent to the server specified by the cookie. If, however, the cookie specifies a server that is *not* in the match rule's server list, the cookie is ignored.

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 happy 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 expression¹. 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 expressions (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")
```

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 (unless a responder is selected in the match rule, in which case the responder sends a response to the client).

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;
  servers = sv_01;
}
```

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`, which are all sent to server `sv_01`. Additionally, this rule disables the

1. Regular expressions are specified according to IEEE Std 1003.2 (“POSIX.2”).

spoof flag (that is, when the connection is made to the server, the server sees a connection to the Equalizer, not to the client). This rule looks as follows in the Administrative Interface:

The screenshot displays the configuration for a match rule in the Administrative Interface. It is divided into several sections:

- order:** Set to "immediately before" with a "Default" dropdown menu.
- expression:** Contains the expression `client_ip("10.0.0.19")`, which is highlighted in yellow. An arrow points to this section with the text: "If the incoming client IP address matches this expression...".
- servers and options:**
 - servers:** A list box containing "sv01" and "sv02". An arrow points to this list with the text: "...then, load balance the request across the selected servers...".
 - policy:** Set to "custom".
 - options:**
 - cookie age: 0
 - cookie domain: (empty)
 - cookie path: (empty)
 - disable:
 - spoof: (inherit from cluster:)
 - once only: (inherit from cluster:)
 - abort server: (inherit from cluster:)
 - persist: (inherit from cluster:)
 - insert client IP: (inherit from cluster:)
 - persist always: (inherit from cluster:)
- responder:** Set to "response" with a "none" dropdown menu. An arrow points to the "spoof" option with the text: "...and disable the spoof option (leave all other options set to the cluster settings).".

Figure 39 Example match rule

The **expression** section of the screen shows the expression that is evaluated against the incoming request. If the expression evaluates to *true*, the **servers and options** section specifies the servers that will be used to satisfy the incoming request, as well as the options that will be set for the request. The next section of this document explains all the match rule settings in detail.

Managing Match Rules

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.

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 226.

The Match Rules Table

Click on a cluster name in the left frame and then click on the **Match Rules** tab to display a list of match rules defined for that cluster.

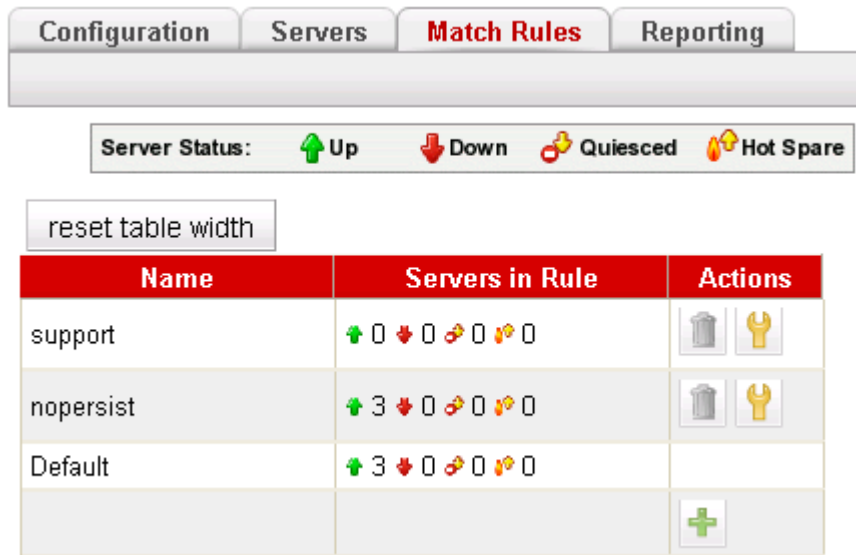


Figure 40 The match rules table

Name	The match rule name.
Server in Rule	Status indicators for all servers in the rule. Shows the number of servers in each of the following states: Up (responding to health check probes), Down (not responding to health check probes), Quiesced (not accepting new connections), and Hot Spare (only responding to requests when no other server is up).
Actions	Delete or Modify the match rule in the same row as the icon chosen. The Add icon at the bottom of the column opens the Add New Match Rule dialog.
reset table width	The columns on the table can be resized. If you extend a column too far to the right so that other columns are no longer visible, this button returns the table to its default proportions.

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 121). Figure 41 shows the default match rule for an HTTPS cluster on an E650GX with two servers.

order
immediately after

expression
any()

servers and options

servers
sv00
sv01

policy
round_robin

compress mime-types
text/*:application/msword:appli

cookie age
0

cookie domain

cookie path

disable

spooof

once only

abort server

persist

compress

insert client IP

certify_client

ssl unclean shutdown

no header rewrite

persist always

responder
response

Figure 41 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. Do this by creating a new rule *immediately before* the Default that uses any() as the matching expression, so that the Default match rule is never processed. This effectively creates a new default match rule that you can configure with the desired load balancing options. Also note that some options in the match rule displayed in Figure 41 are only displayed for an HTTPS cluster, or on an Equalizer with GZIP compression enabled.

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 Administrative Interface using a login that has **add/del** access for the cluster (see “Logging In” on page 52).
2. In the left frame, right-click the name of the Layer 7 cluster to which you want to add a match rule, and select **Add Match Rule**.

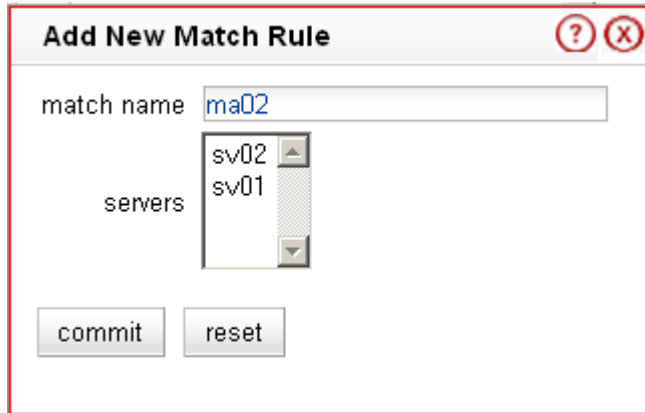


Figure 42 Example Add New Match Rule dialog box

3. Enter a name for the new rule in the **match name** field or accept the default. All match names within a cluster must be unique.
4. In the **servers** section, use the CTRL+left click and SHIFT+left click to select the names of the servers that you want to use to load balance requests that match the rule.

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.*

Select **commit** once you choose the servers for the match rule.

- The Match Rule **Configuration** tab is displayed.

The screenshot shows the Match Rule Configuration tab with the following settings:

- order:** immediately before **Default** (dropdown)
- expression:** any() (highlighted in yellow) with an undo button below it.
- servers and options:**
 - servers:** A list box containing sv00 and sv01, with sv00 selected.
 - policy:** round robin (dropdown)
 - compress mime-types:** text/*:application/msword:appli (text input)
 - cookie age:** 0 (text input)
 - cookie domain:** (empty text input)
 - cookie path:** (empty text input)
 - disable:**
 - spooof:** (inherit from cluster:)
 - once only:** (inherit from cluster:)
 - abort server:** (inherit from cluster:)
 - persist:** (inherit from cluster:)
 - compress:** (inherit from cluster:)
 - insert client IP:** (inherit from cluster:)
 - certify_client:** (inherit from cluster:)
 - ssl unclean shutdown:** (inherit from cluster:)
 - no header rewrite:** (inherit from cluster:)
 - persist always:** (inherit from cluster:)
- responder:** response **none** (dropdown)

Figure 43 Match rule **Configuration** tab

The **order** field displays the name of the rule before which the currently displayed rule is evaluated. By default, a new rule is placed immediately before the Default rule. Change the placement of the new rule by choosing a rule from the **immediately before** list box. The evaluation order of the rules in a cluster is shown in the left frame.

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, any() is one that will always evaluate to *true*, meaning that this match rule will always 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. Use the **disable** flag (see Step 9) to skip a match rule that is still being developed.

- Build your match rule expression in the **expression** section. To place or modify a match function, click the appropriate part of the expression. The part of the expression that the editor will directly affect is now displayed in a dialog box.

- a. From the drop-down list, select the match function and or expression with which you want to replace the selected part of the expression. Supply values for all arguments required by the function. To learn more about match functions, refer to “Match Functions” on page 235.

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. Some of the structural editing operators include the function you are replacing (for example, if you have selected the host() function, **replace with host AND any** will appear in the drop down box).

- b. Click the **continue** button. If there are any syntax errors, an error screen appears. This most likely occurs if there are missing arguments or syntax errors in the argument strings. Correct the error and click **continue**. If your changes are syntactically correct, Equalizer displays the new version of the match expression in the **Configuration** tab.
 - c. Repeat **a** and **b** until your expression is complete.
7. The **servers and options** section allows you to specify the following load balancing options for matching requests:

policy	Change these parameters to override the cluster setting. See “Modifying a Layer 7 Virtual Cluster” on page 122 for an explanation of these parameters.
compress mime-types (E650GX only)	
cookie age	
cookie domain	
cookie path	
disable	Enable this flag to disable this match rule without deleting it. This can be useful when testing new match rules.
spooF	The two columns of check boxes to the right of these flags 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. The right-hand check box for each flag, if set, indicates that the flag setting will be inherited from the cluster setting -- in the screen above in Step 7, the spooF setting on the cluster (enabled) will be overridden for this match rule. See “Modifying a Layer 7 Virtual Cluster” on page 122 for an explanation of these parameters.
once only	
abort server	
persist	
compress (E650GX only)	
insert client IP (HTTPS only)	
certify client (HTTPS only)	
ssl unclean shutdown (HTTPS only)	
no header rewrite (HTTPS only)	
persist always	

- 8. The **responder** field allows you to specify an automatic responder for client requests that match this rule when none of the servers selected in the rule are available. The responder must already be configured. For a description of responders as well as examples of using responders in match rules, see the section “Automatic Cluster Responders” on page 162.
- 9. Click **commit** to save the match rule definition.

Modifying a Match Rule

To edit a match rule, follow these steps:

1. Log into the Administrative Interface using a login that has **write access** for the cluster (see “Logging In” on page 52).
2. In the left frame, click the name of the match rule to be changed.
3. Make the desired changes to the match rule, as shown in the procedure in the previous section, starting at Step 5 on page 233.

Removing a Match Rule

To delete a match rule, follow these steps:

1. Log into the Administrative Interface using a login that has **add/del** access for the cluster (see “Logging In” on page 52).
2. In the left frame, right-click the name of the match rule to be deleted and select **Delete Match Rule** from the local menu.
3. Click **delete** to confirm that you want to delete 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 drop-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 241.

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:

non-URI Match Function	Description
any()	This function always evaluates to <i>true</i> .
client_ip(<i>string</i>)	<p>This function evaluates to <i>true</i> only if the IP address of the client machine making the connection matches the <i>string</i> argument.</p> <p>The <i>string</i> 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.</p>

non-URI Match Function	Description
debug_message(<i>string</i>)	This function always evaluates to <i>true</i> . It writes the <i>string</i> 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. <i>Use this function for testing and debugging only. Do not use it in production environments, since it has a negative impact on performance.</i>
ignore_case()	This function always evaluates to <i>true</i> , and is intended to be used to apply the ignore_case flag for comparisons when it is <i>not set</i> 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 <i>true</i> , and is intended to be used to override the ignore_case flag for comparisons when it is <i>set</i> 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 <i>true</i> 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(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> 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.
ssl2()	HTTPS only. This function evaluates to <i>true</i> if the client negotiated the encrypted connection using SSL version 2.0.
ssl3()	HTTPS only. This function evaluates to <i>true</i> if the client negotiated the encrypted connection using SSL version 3.0.
tls1()	HTTPS only. This function evaluates to <i>true</i> if the client negotiated the encrypted connection using TLS version 1.0.
header match functions	<i>See "Match Function Notes" on page 239, for the headers that can be specified in these functions.</i>
header_prefix(<i>header, string</i>)	This function evaluates to <i>true</i> if the selected <i>header</i> is present and if the string-valued argument <i>string</i> is a prefix of the associated header text.
header_suffix(<i>header, string</i>)	This function evaluates to <i>true</i> if the selected <i>header</i> is present and if the argument <i>string</i> is a suffix of the associated header text.
header_substr(<i>header, string</i>)	This function evaluates to <i>true</i> if the selected <i>header</i> is present and if the string-valued argument <i>string</i> is a sub-string of the associated header text.

non-URI Match Function	Description
header_regex(<i>header</i>, <i>string</i>)	This function evaluates to <i>true</i> if the selected <i>header</i> is present and if the string-valued argument <i>string</i> , interpreted as a regular expression, matches the associated header text.

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:

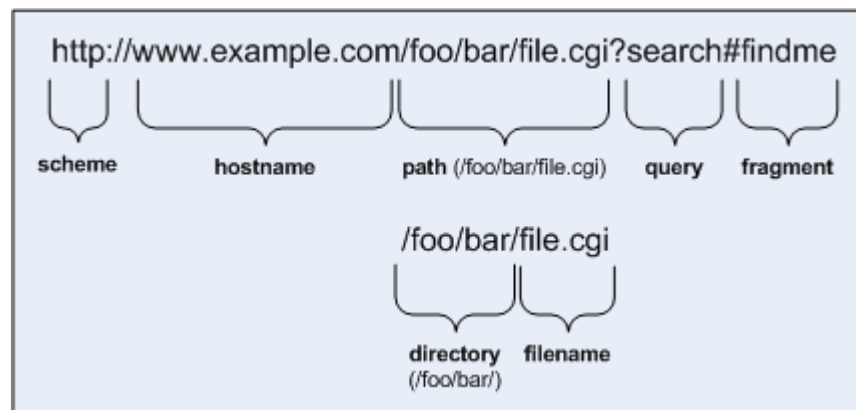


Figure 44 URI components

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 44.

URI Match Function	Description
host(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument exactly matches the hostname portion of the request. <i>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.</i>

URI Match Function	Description
host_prefix(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> 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").
host_suffix(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> 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").
pathname(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument exactly matches the path component of the request URI.
pathname_prefix(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument is a prefix of the path component of the request URI.
pathname_suffix(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument is a suffix of the path component of the request URI.
pathname_substr(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument is a substring of the path component of the request URI.
pathname_regex(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument, interpreted as a regular expression, matches the path component of the request URI.
dirname(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> 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(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument is a prefix of the directory portion of the path component of the request URI. The leading slash must be included in the <i>string</i> (for example, "/fo" is a prefix of "/foo/bar/file.html").
dirname_suffix(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument is a suffix of the directory portion of the path component of the request URI. The trailing slash must be included in the <i>string</i> (for example, "ar/" is a suffix of the directory portion of "/foo/bar/file.html").
dirname_substr(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument is a substring of the directory portion of the path component of the request URI.
dirname_regex(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument, interpreted as a regular expression, matches the directory portion of the path component of the request URI.
filename(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument exactly matches the filename portion of the URI path. <i>This portion includes only the text after the last trailing path component separator (/), as that is considered part of the directory</i> (for example, "file.html" is the filename portion of "/foo/bar/file.html").
filename_prefix(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument is a prefix of the filename portion of the URI path.

URI Match Function	Description
filename_suffix(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument is a suffix of the filename portion of the URI path.
filename_substr(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument is a substring of the filename portion of the URI path.
filename_regex(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument, interpreted as a regular expression, matches the filename portion of the URI path.
query(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> 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(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument is a prefix of the query portion of the URI path.
query_suffix(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument is a suffix of the query portion of the URI path.
query_substr(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument is a substring of the query portion of the URI path.
query_regex(<i>string</i>)	This function evaluates to <i>true</i> if the <i>string</i> argument, interpreted as a regular expression, matches the query portion of the URI path.

Match Function Notes

Please observe the notes in the following sections when constructing match rules.

Match Rule Behavior When Server Status is not 'Up'

When a match rule expression matches a client request, the request is load balanced using the servers, parameters, and flags specified in the match rule. The servers specified in the match rule may be in a number of "states" that affect the load balancing behavior: the server may be up or down, and may have one or both of the **quiesce** and **hot spare** options enabled.

server up	The request is routed to the selected server.
up/quiesce enabled	The request is routed to the selected server.
up/hot spare enabled	The request is routed to the selected server.
server down	If no Responder is selected in the match rule, then the request is sent to the selected server and, eventually, the client times out. If a Responder is selected, the Equalizer sends the configured response to the client.

The reason match rules behave as shown above is because the purpose of a match rule is to send a request that matches an expression to a particular server that can (presumably) better satisfy the request. In some cases, sending the request to a particular server may be required behavior for a particular configuration.

With this in mind, it does not make sense to skip a match rule because the server (or servers) named in the rule are down, hot spared, or quiesced -- rather, since the server in the rule is presumably critical to satisfying the request, it

makes sense to route the request to the (for example) down server, and have the client receive an appropriate error -- so that the request can be retried.

If we instead were to skip a match rule because, for example, the server selected by the match rule is down, the request would be evaluated by the next match rule -- or the default match rule. The request, therefore, could potentially be sent to a server in the cluster that does not have the requested content. This means that the client would receive a “not found” error, instead of an error indicating that the appropriate server is not currently available.

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"))
```

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") OR
dirname_substr("four") OR
dirname_substr("six") OR
dirname_substr("eight")
```

Equalizer supports POSIX regular expression syntax only. See Appendix D, “Regular Expression Format” for a description.

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 instance 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:

Accept	From	Referer
Accept-Charset	Host	TE

Accept-Encoding	If-Match	Trailer
Accept-Language	If-Modified-Since	Transfer-Encoding
Authorization	If-None-Match	Upgrade
Cache-Control	If-Range	User-Agent
Connection	If-Unmodified-Since	Via
Content-Length	Max-Forwards	Warning
Cookie	Pragma	X-Forwarded-For
Date	Proxy-Authorization	
Expect	Range	

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.

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. Note in particular 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.

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 <i>eq.conf</i> 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.
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 <i>function</i> with the "any()" function logically OR'ed with the current selection.
replace with function AND any	Replaces the currently selected <i>function</i> with the current selection logically AND'ed with the "any()" function.
replace with function OR any	Replaces the currently selected <i>function</i> 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 right side of the logical expression (e.g., "A AND B" becomes "B").

Using Responders in Match Rules

Responders are used to send automated responses to clients when all the servers in a match rule are down. See the section "Automatic Cluster Responders" on page 162 for a complete description of Responders as well as examples of using Responders in Match Rules.

Example Match Rules

This section shows you how to create a few of the most commonly used types of match rules:

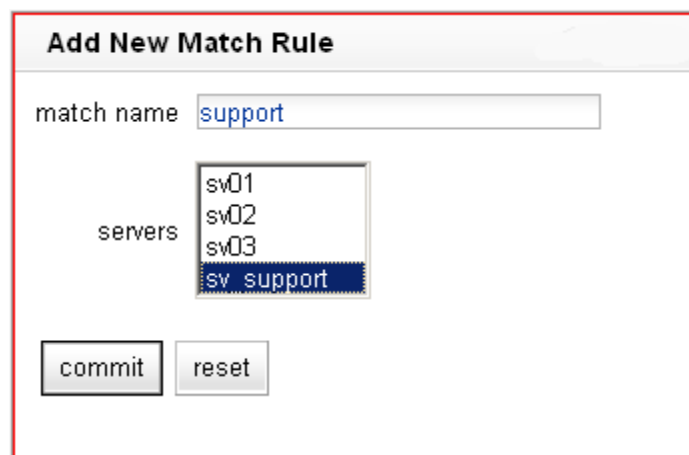
- "Parsing the URI Using Match Rules" on page 243
- "Changing Persistence Settings Using Match Rules" on page 244
- "Changing the Spoof (SNAT) Setting Using Match Rules" on page 246
- "Server Selection Based on Content Type Using Match Rules" on page 249

Parsing the URI Using Match Rules

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**, **sv01**, **sv02**, **sv03**) has already been defined.

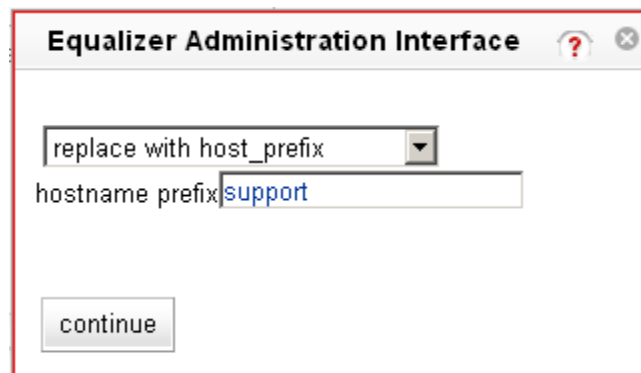
1. Log into the Administrative Interface using a login that has **add/del** access for the cluster (see “Logging In” on page 52).
2. In the left frame, right-click the name of the Layer 7 cluster to which you want to add the rule, and select Add Match Rule. The **Add Match Rule** dialog appears:
 - a. Type **support** into the **match name** text box.
 - b. Select **sv_support** in the servers list; make sure only this server is selected:



- c. Select **commit**.

The match rule is created, added to the object tree, and its **Configuration** tab is opened, as shown on the following page:

3. In the **expression** field, select **any()** to open the **Select function** dialog:
 - a. Select **replace with host_prefix** from the drop-down box.
 - b. Type “**support**” into the **hostname prefix** text box. The dialog should now look like this:



- c. Click **continue**.
4. Select the **commit** button to save your changes to the **support** rule.

Changing Persistence Settings Using Match Rules

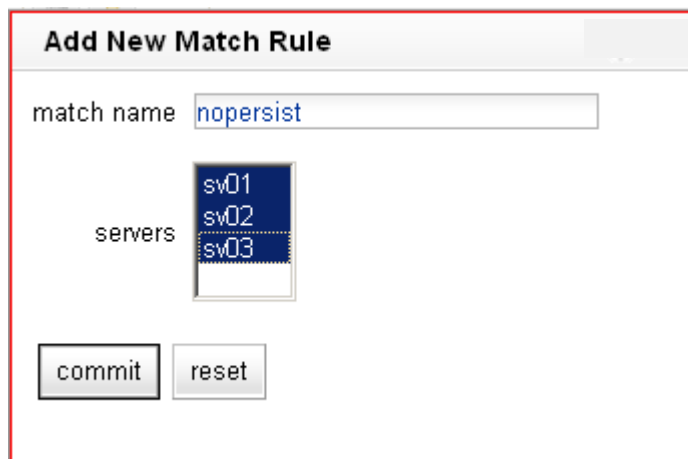
By default, a client request that matches a match rule expression is load balanced using the same load balancing parameters and options that are currently set on the cluster. This section shows you how to change load balancing parameters and flags in a match rule.

For example, persistent connections to servers are enabled by the **persist** cluster flag, which is enabled by default when you create a cluster. While you could select the **dont persist** option for a server, this means that *all* connections to that server will not be persistent. Let's assume that you only want to disable persistence for incoming requests that have a URI containing a hostname in the following format:

```
xxx.testexample.com
```

We'll use the `host_suffix()` match rule function to test for the above hostname format. 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 Administrative Interface using a login that has **add/del** access for the cluster (see "Logging In" on page 52).
2. In the left frame, right-click the name of the Layer 7 cluster to which you want to add the rule, and select **Add Match Rule**. The **Add Match Rule** dialog appears:
 - a. Type **nopersist** into the **match name** text box.
 - b. Select all the servers in the **servers** list using the **Ctrl** or **Shift** key and the left mouse button

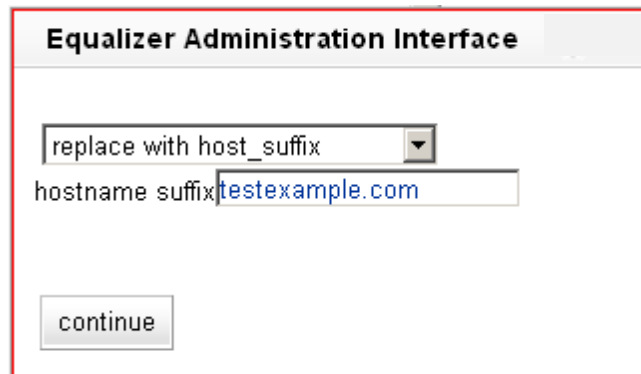


- c. Select **commit**.

The match rule is created, added to the object tree, and its **Configuration** tab is opened, as shown on the following page:

3. In the **expression** field, select **any()** to open the **Select function** dialog:
 - a. Select **replace with host_suffix** from the drop-down box.

- b. Type “**testexample.com**” into the **hostname suffix** text box. The dialog should now look like this:



The screenshot shows a dialog box titled "Equalizer Administration Interface". It contains a dropdown menu with the text "replace with host_suffix" and a small downward arrow. Below the dropdown is a text input field with the label "hostname suffix" and the text "testexample.com" entered. At the bottom left of the dialog is a button labeled "continue".

- c. Click **continue**.
4. In the **servers and options** field, disable both of the two check boxes to the right of the **persist** flag:
persist (inherit from cluster:)
5. Select the **commit** button to save your changes to the **nopersist** rule.

Changing the Spoof (SNAT) Setting Using Match Rules

By default, Equalizer uses the client IP address as the source address in the packets it forwards to servers, and then translates the server IP in server responses to Equalizer's cluster IP. This is commonly called a *Half-NAT* configuration, since Equalizer is *not* performing Network Address translation (or NAT) on client requests. Because the servers behind Equalizer see the source IP of the client, the servers need to be configured to route client requests back through Equalizer -- either by making Equalizer the default

This behavior is controlled by the **spoof** option, which is enabled by default. Half-NAT configurations are only a problem when a client is on the same subnet as the servers behind Equalizer, since the servers will try to respond directly back to the client -- which will not recognize the server connection as a response to its original request and so refuse the connection.

This 'local client' problem is solved by *disabling* the **spoof** option. When **spoof** is disabled, Equalizer translates the source IP address in the request to one of Equalizer's IP addresses before sending it on to the server. This is called *Source Network Address Translation*, or *SNAT* -- and this configuration is often called *Full-NAT*, since Equalizer is translating the client IP in packets from clients, as well as the server IP in packets from servers. In this case, servers will send responses to Equalizer's IP address, so no special routing or gateway is needed on the server.

So, clusters with clients on a different subnet than the servers behind it can have the spoof option enabled, while clusters with only local clients should have spoof disabled.

But what do you do if you expect client requests to come to the cluster from the local server subnet as well as other subnets?

In network configurations where Equalizer needs to be able to forward server responses to clients on the server subnet as well as other subnets for the same virtual cluster IP, the **spoof** option can be selectively enabled or disabled by creating a Layer 7 match rule that looks for specific client IP addresses in incoming requests. When an incoming request's source IP matches the rule, **spoof** will be set as appropriate for that connection. This is commonly called *Selective SNAT*.

For an overview of how load balancers and application accelerators like Equalizer use NAT and SNAT, please see the following website:

<http://lbdigest.com/2009/03/11/best-of-both-worlds-selective-source-nat/>

On Equalizer, implementing Selective SNAT using a Match Rule is the recommended method to allow local access to Layer 7 clusters with **spoof** enabled; other alternatives include:

- adding static routes on all your servers to clients on the server's local subnet
- creating two clusters -- one on the non-server subnet with **spoof** enabled, and one on the server subnet with spoof disabled

Selective SNAT using a match rule is more easily implemented and maintained than either of the above methods, but can be configured only for Layer 7 clusters. If you require Selective SNAT with a Layer 4 cluster, you'll need to use one of the above methods.

Selective SNAT Example

The procedure below shows you how to create a match rule that selectively disables the cluster **spoof** option based on the client IP address of an incoming connection. It is assumed that the cluster for which the match rule is created has **spoof enabled** on the cluster **Networking** tab, and that the cluster works properly for clients on subnets other than the subnet to which the servers in the cluster are connected.

1. Right-click the name of the cluster for which you want to implement selective SNAT, and choose the **Add Match Rule** command from the menu.
2. In the **Add New Match Rule** dialog:
 - a. Type in a **Match Name** or accept the default.
 - b. Select all the **Servers** in the cluster.

- c. Click **commit**.

The new match rule is created and its **Configuration** tab is opened.

3. In the **expression** field, click on **any()**.

4. In the **Edit Match Rule** dialog:

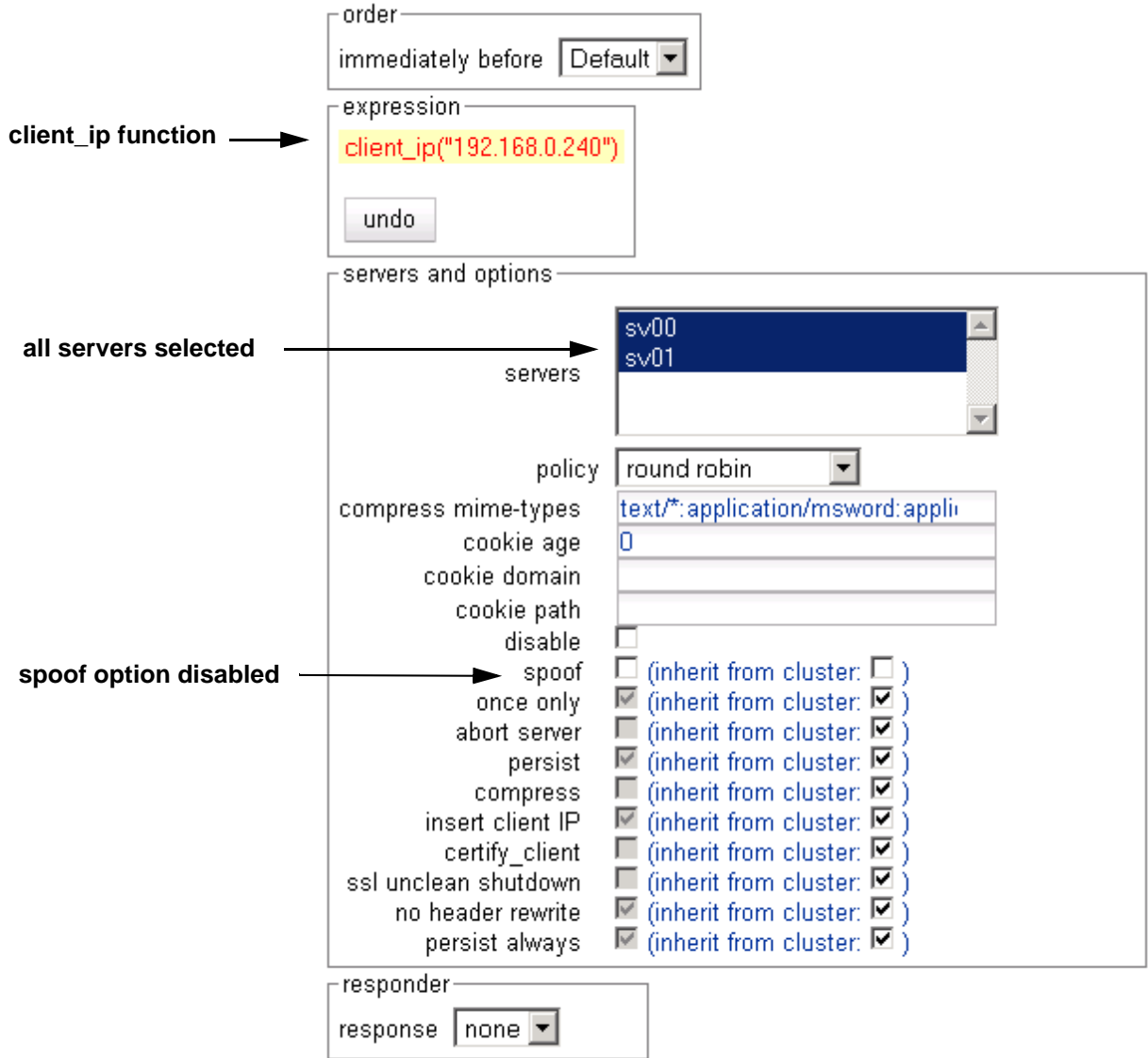
- a. Select **replace with client_ip** from the drop-down box.
- b. In the **ip** text box, specify a simple IP address (e.g., “192.168.0.240”), or an IP address in Classless Inter-Domain Routing (CIDR) notation (e.g., “192.168.0.0/24”) to specify an entire subnet.
- c. Click **continue**.

The **expression** field should now contain the **client_ip** function with the **ip** argument you specified above.

5. In the **servers and options** field, look for the **spoof** option and disable both of the checkboxes to the right, in this order:
 - a. Disable the **inherit from global** checkbox
 - b. Disable the **spoof** checkbox

(continued on the following page)

The **Configuration** tab should now look similar to the example below:



6. At the bottom of the **Configuration** tab, click **commit**.

Clients whose IP addresses are selected by the new match rule should now be able to connect successfully to the cluster IP. Right-click the name of the match rule in the left frame; the **Processed** counter in the popup menu should increase as clients are selected by the match rule. Select **Match Rule Plots** from the popup menu to display a history of the number of connections processed by the match rule.

Server Selection Based on Content Type Using Match Rules

In this example, assume a configuration that has dedicated one or more servers to return only image files (.gif, .jpg, etc.), while the remainder of the servers return all the other content for client requests.

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

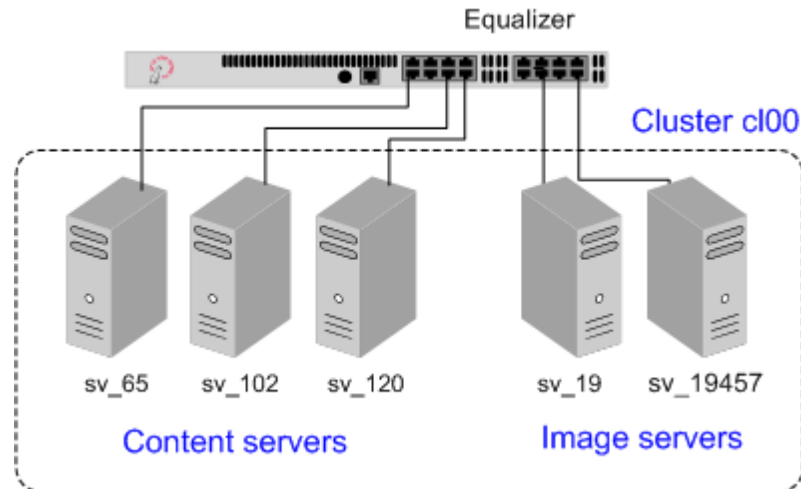


Figure 45 Match Rule Example: Dedicated Image and Content Servers

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 Administrative Interface using a login that has **add/del** access for the cluster (see "Logging In" on page 52).
2. In the left frame, click the name of the Layer 7 cluster to which you want to add the rule. The cluster **Configuration** screen appears in the right frame:
 - a. Make sure that the **once only** flag is not checked; otherwise, uncheck the **once only** flag and click **commit**.
 - b. Open the **Persistence** tab and make sure the **persist** flag is not checked; otherwise, uncheck the **persist** flag and click **commit**.

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.

3. Right-click the cluster name in the left frame and select **Add Match Rule**. The **Add Match Rule** dialog appears:
 - a. 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.
 - b. In our example, we want all the images to be served from either **sv_19** or **sv_19457**. In the **servers** field, select **sv_19, sv_19457** using the **Ctrl** or **Shift**.
 - c. Select **commit**.

The match rule is created, added to the object tree, and its **Configuration** tab is opened:

4. In the **expression** field, click **any()** to open the **Select function** dialog:
 - a. Select **replace with filename_suffix** from the drop-down box.
 - b. Type “**jpg**” into the **filename suffix** text box.
 - c. Select **continue**.
5. In the **expression** field, click **filename suffix(“jpg”)** to open the **Select function** dialog:
 - a. Select **replace with filename_suffix OR any()** from the drop-down box.
 - b. Select **continue**.
6. In the **expression** field, click **any()** to open the **Select function** dialog:
 - a. Select **replace with filename_suffix** from the drop-down box.
 - b. Type “**jpeg**” into the **filename suffix** text box.
 - c. Select **continue**.
7. Repeat Steps 5 and 6 for each of the other filename suffixes on our example servers -- **gif**, **bmp**, and **png**.

When you are done, the match expression should look like this:

```

If following expression matches
( filename_suffix(".jpg")
  OR filename_suffix("jpeg")
  OR filename_suffix("gif")
  OR filename_suffix("bmp")
  OR filename_suffix("png")
)
  
```

8. Select the **commit** button to save your changes to the **images** rule.
9. 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, right-click the name of the cluster and select **Add Match Rule**. The **Add Match Rule** screen appears.:

- a. Type “**content**” into the **match name** text box
- b. In the **servers** field, select **sv_102**, **sv_65**, and **sv_120**.
- c. Select **commit**.

The match rule is created, added to the object tree, and its **Configuration** tab is opened:

10. Select **Default** in the **immediately before** drop-down box.
11. 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.)
12. Select the **commit** button to save your changes to the **content** rule.

Using the Custom Load Balancing Policy with Match Rules

The **policy** drop down box in a match rule allows you to select an alternate load balancing policy for requests selected by the match rule. While the **custom** policy is an available choice for **policy** in a match rule, the match rule tab does not provide the controls used to set the specific custom load balancing behavior (see the section “LB Policy Tab” on page 127 for a description of these controls).

In order to use the **custom** policy in a match rule, the **custom** policy should be selected as the policy for the cluster, and match rules used to set any *other* policies. When the **custom** policy match rule matches a request, the **custom** policy will be used with the settings from the cluster configuration. The following example illustrates how to do this.

Let’s say that you want to serve most requests to a cluster using the **adaptive** load balancing policy, but want to load balance requests to a particular URL using a **custom** policy. To do this, you would:

- Set the **custom** policy in the cluster configuration (setting the slider controls to match your requirements). This changes the **policy** specified on the **Default** match rule to **custom**.
- Create a match rule that matches all requests that DO NOT specify the URL that you want to load balance using the **custom** policy. In this match rule, specify the **adaptive** policy.

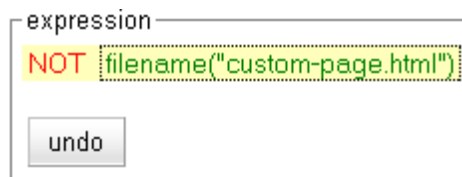
The following procedure shows you how to do this. In this example, we’ll assume that the **custom** policy is to be used for requests to the URL `http://www.example.com/custom-page.html`, and the **adaptive** policy will be used for all other requests.

1. Click the *cluster name* in the left frame, and open the **LB Policy** tab in the right frame.
 - a. Select **custom** from the policy drop-down box.
 - b. Set the slider controls to reflect the relative importance of the load balancing parameters.
 - c. Click **commit**.
2. Right-click the name of the cluster and select **Add Match Rule** from the popup menu.
 - a. Type in a **Match Name** or accept the default.
 - b. Select all of the **Servers** in the cluster.
 - c. Click **commit**.

The new match rule is created and its **Configuration** tab is opened.

3. In the **expression** field, click on **any()**.
4. In the **Edit Match Rule** dialog:
 - a. Select **replace with filename** from the drop-down box.
 - b. In the **filename** text box, type `custom-page.html`.
 - c. Click **continue**.
5. Click on the **filename** function you just added in the **expression** field.
 - d. Select **negate function** from the drop-down box.
 - e. Click **continue**.

The **expression** field should now look like this:



6. In the **servers and options** field, select **adaptive** from the **policy** drop-down box.
7. Click **commit** at the bottom of the tab to save your changes to the match rule.

Administering GeoClusters



Envoy is not supported on E250GX model Equalizers

The Envoy geographic load balancer, an optional software add-on for the Equalizer product line, supports load balancing requests across servers in different physical locations or on different networks.

- Overview of Geographic Load Balancing with Envoy254**
 - Overview of Configuration Process254
 - Overview of Envoy Site Selection254
- Licensing and Configuring Envoy258**
 - Enabling Envoy258
 - Configuring the Authoritative Name Server to Query Envoy258
 - Using Envoy with Firewalled Networks260
 - Using Envoy in a Failover Configuration260
 - Using Envoy with NAT Devices260
 - Upgrading a Version 7 GeoCluster to Version 8261
- Working with GeoClusters262**
 - Adding a GeoCluster262
 - Viewing and Modifying GeoCluster Parameters263
 - Deleting a GeoCluster265
 - Displaying Envoy Statistics266
 - Plotting GeoCluster History266
- Working with Sites266**
 - Adding a Site to a GeoCluster266
 - Displaying and Modifying Site Information268
 - Deleting a Site from a GeoCluster270
 - Displaying Site Statistics270
 - Plotting Site History270
- Envoy Configuration Worksheet271**

Overview of Geographic Load Balancing with Envoy

In non-Envoy Equalizer configurations, there is a one-to-one correspondence between a cluster and a website: when a client makes a request for a website (say, `www.example.com`), the client uses the Domain Name Service (DNS) to resolve the website name to an IP address. For a website that is load balanced by an Equalizer, the IP address returned is the IP address of an Equalizer cluster. After resolving the name, the client sends the request to the cluster IP. When Equalizer receives the client request, it load balances the request across the servers in the cluster, based on the current load balancing policy and parameters.

In an Envoy conversation, you have two or more Equalizers located in separate locations. Each Equalizer and its set of clusters and servers forms a *site* (or *Envoy site*). With Envoy, the website name in the client request is resolved to a *GeoCluster IP*. A GeoCluster is analogous to a cluster, but one level above it: in other words, a GeoCluster actually points to two or more clusters that are defined on separate Equalizers.

In the same way that Equalizer balances requests for a cluster IP across the servers in the cluster, Equalizer load balances a request for a GeoCluster IP across the clusters in the GeoCluster configuration. Once a site is chosen and the client request arrives at that site, the request is load balanced across the servers in the appropriate cluster. In this way, you can set up geographically distant Equalizers to cooperatively load balance client requests.

Overview of Configuration Process

Follow this general procedure when setting up Envoy for the first time on two or more Equalizers running Version 8:

1. Configure appropriate clusters (and servers) on all of the Equalizers to be included as Envoy sites in the GeoCluster.
2. Configure the GeoCluster on each Equalizer; the parameters used should be the same on all sites.
3. Configure the authoritative DNS server for your website's domain with DNS records for all Equalizers in the GeoCluster. The DNS server returns these records to clients in response to DNS requests to resolve the website (GeoCluster) name.

Note – While it is possible to mix Version 8 and Version 7 Equalizers in the same GeoCluster, we recommend that you run the same version of Equalizer software on all Equalizers in your GeoCluster. If you must run Version 8 and Version 7 Equalizers in an Envoy configuration, or if you are upgrading an existing Version 7 Envoy configuration to Version 8, see the section “Upgrading a Version 7 GeoCluster to Version 8” on page 261 for additional notes.

Overview of Envoy Site Selection

When a client uses DNS to resolve the address of a website name, it first contacts its local DNS server to resolve the name. The local DNS server then begins the process of resolving the website name by contacting other DNS servers to locate the authoritative DNS server for the website's domain. The authoritative DNS server for the website's domain returns a list of Envoy sites to the client's DNS server. Once it has this list, the client's local DNS server sends requests, one at a time, to each of the Envoy sites until it reaches an active site. (An overview of this process is given in the section “Distributing the Geographic Load” on page 26.) The local DNS server then waits for Envoy to resolve the website name into an IP address and return it.

Once an active Envoy site is reached, Envoy performs the following steps to determine the virtual cluster IP to return to the client's DNS:

1. Figure 46 is an illustration of a client in California whose local DNS server has contacted Envoy Site A to resolve the destination domain name for the client request -- in this example, `www.coyotepoint.com`.

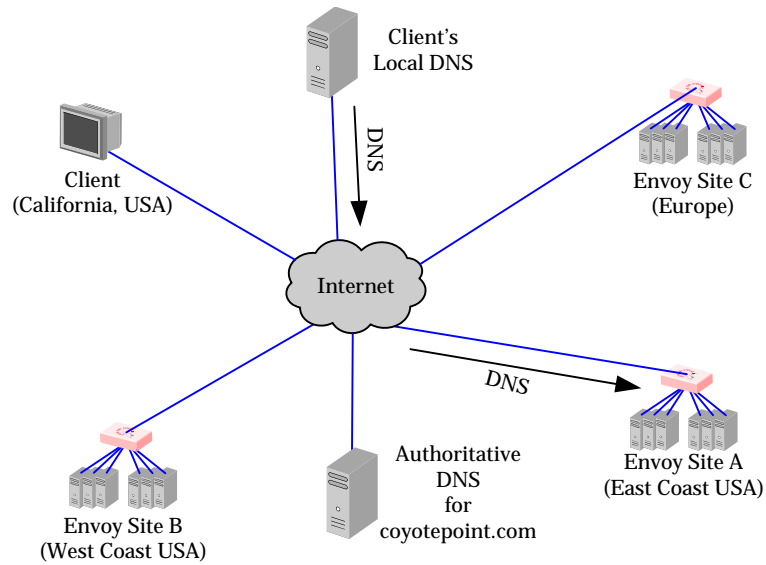


Figure 46 Sending name resolution requests to an Equalizer in a GeoCluster

2. Site A sends a *geographic query protocol probe (GQP)* to all the other Envoy sites in the GeoCluster for the requested domain (this GeoCluster has the same name -- `www.coyotepoint.com`). The probe is received by a special Envoy *agent* running at each site in the cluster (the agent for a site starts when you configure Envoy for the site). Site A also queries its local Envoy agent (see Figure 47).

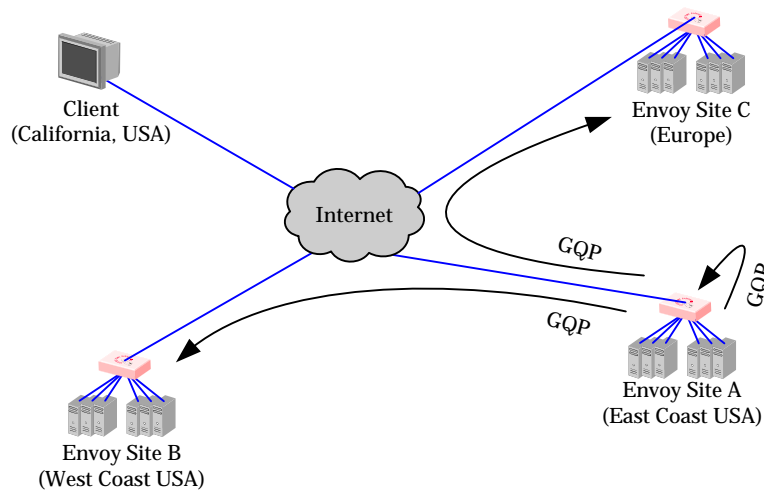


Figure 47 The selected Equalizer queries other Equalizers and its own servers in the GeoCluster

The GQP probes contain information about the requesting client and the local resource (i.e., local cluster) that is being requested by the client.

If **ICMP triangulation** is enabled, the GQP probes also tell the sites to send an ICMP echo request (*ping*) to the client's local DNS server, and to return the DNS server's response time in their GQP response to Site A. This provides more accurate client location information to Envoy in the case where a resource is available at more than one site.

3. The Envoy agent at each site checks the availability of the requested resource and sends a GQP reply to the Envoy agent running at Site A (see Figure 48).

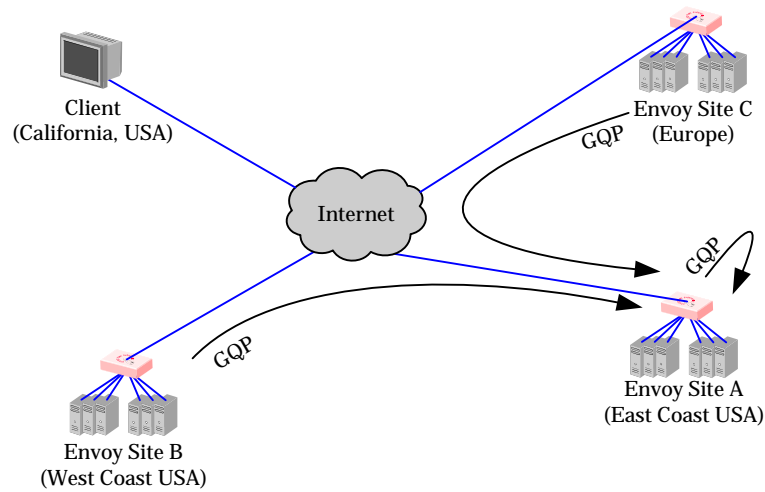


Figure 48 The selected Equalizer receives availability and triangulation (latency) information

The content of each response is as follows:

- If the resource (cluster) is not available at the site, the site sends an error message.
 - If the resource (cluster) is available at the site, the site sends a message that the resource is available.
 - If **ICMP triangulation** is enabled for the GeoCluster, and the requested resource is available at the site, the site will include the local DNS server’s ICMP echo response time (or lack thereof) in its GQP response to Site A.
4. After all GQP responses are received (or the GQP probes time out), Site A determine the ‘best available’ site to return to the client’s DNS server using this process:
 - a. If at least one GQP probe is received from a site at which the resource is available, then Envoy uses the resource availability and network latency information (if present) in the GQP replies to select a site for the client based on the current load balancing policy, and returns the address of the ‘best available’ Envoy site to the requesting client’s local DNS (see Figure 49).

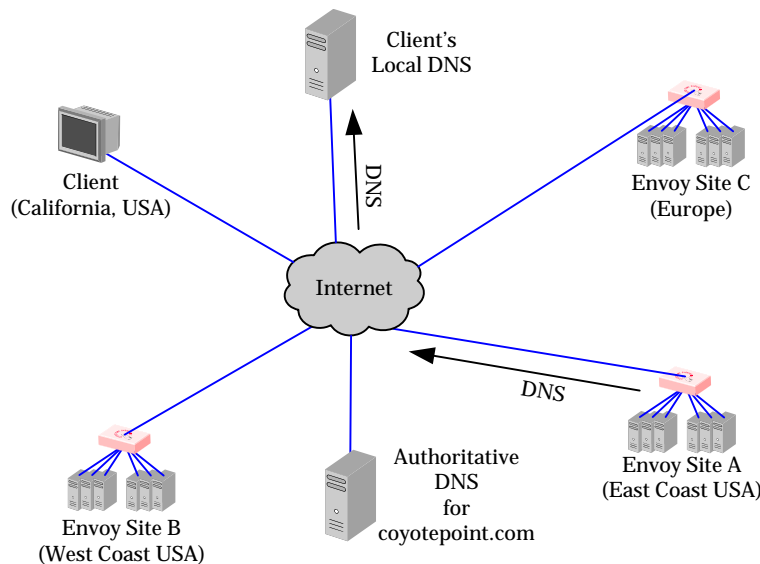


Figure 49 The client’s local DNS receives the best Equalizer site

- b. If no GQP responses are received, or if the requested resource (cluster) is not available at any of the sites that replied, then Site A returns a site to the client's DNS according to this algorithm:

```

If the site that has the default option enabled is up
    Then, send the IP address of the resource at this site (even if weight=0)
Else, if one or more sites are up
    Then, send the IP address of the resource at any site marked up
Else, if a site is marked as the default
    Then, send the IP address of the resource at that site, even if the site
    is marked down
Else, send a NULL response back to the client's DNS server.
    
```

- 5. Once the client's local DNS server sends the client the IP address of the selected site, the client sends the request to the site (see Figure 50). The site then responds to the client and the connection is thereafter managed by the chosen site (in our example, Site B).

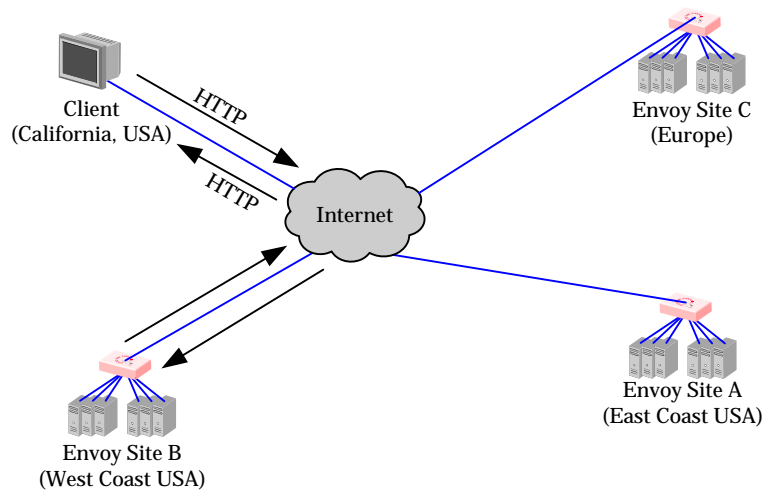


Figure 50 Site B handles the client's connection

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 GeoClusters and define the available sites for each cluster.

Enabling Envoy

To license and enable Envoy, follow these steps:

1. Log into the Equalizer Administration Interface, and expand the **Equalizer System Information** box in the right frame:
 - If the line **Envoy geographic load balancing** shows that Envoy is **enabled**, stop now; Envoy is already licensed.
 - If the line **Envoy geographic load balancing** shows that Envoy is **disabled**, go to the next step.
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 86 in Chapter 5, “Configuring Equalizer Operation”.
2. Shut down the Equalizer and reboot the machine; see “Rebooting Equalizer” on page 116 in Chapter 5, “Configuring Equalizer Operation”.
3. After the system reboots, confirm that Envoy is enabled. Log into the Equalizer Administration Interface and expand the **Equalizer System Information** box in the right frame. 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 51), assume you must balance `www.coyotepoint.com` across a GeoCluster 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.

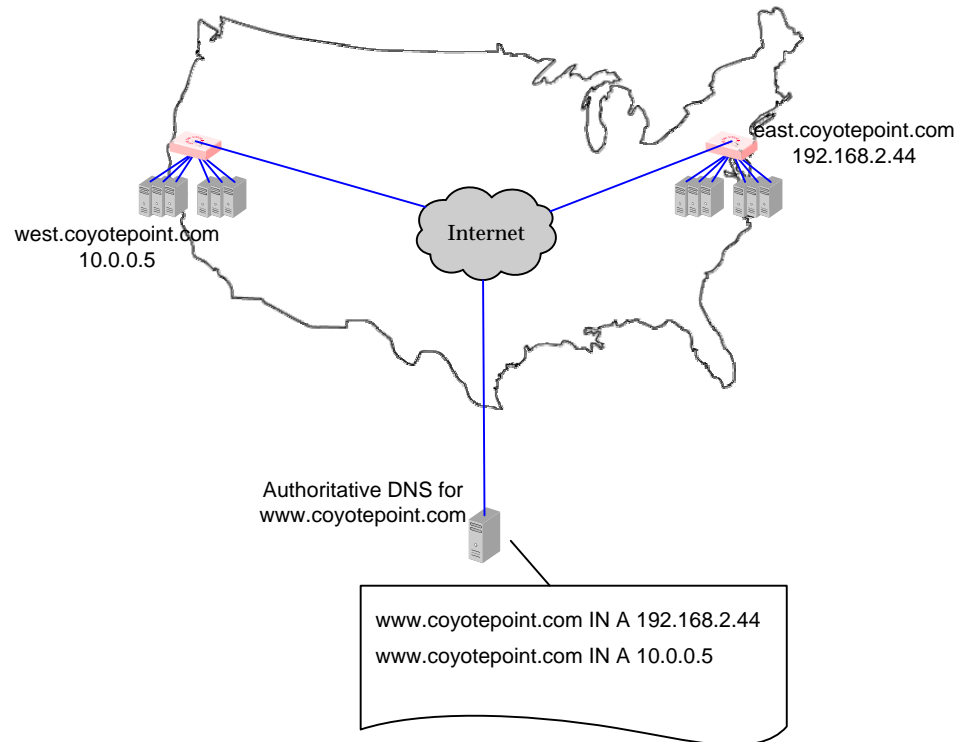


Figure 51 Two-site DNS example

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 52 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.

To ensure that you have properly configured DNS for Envoy, you can use the **nslookup** command (supported on most OS platforms) to confirm that the DNS server is returning appropriate records, as in this example:

```
nslookup www.coyotepoint.com
Server:  ns1.coyotepoint.com
Address:  ns1-IP-address

Name:    www.coyotepoint.com
Address: 192.168.2.44
```

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 site 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.

Using Envoy in a Failover Configuration

When Envoy is being used on a system that is in Standalone mode (that is, failover is not configured), Envoy opens ports on the VLAN IP address of the Default VLAN. When failover is enabled, Envoy will instead open ports on the Failover IP address of the Default VLAN. As a result, **it is required to define a Failover IP address on the Default VLAN when Envoy and failover are both configured:**

- Whenever failover is enabled, the failover subsystem will check to see if any GeoClusters are defined -- if there are, then it will refuse to enter failover mode if there is no Failover IP address on the Default VLAN.
- If failover is enabled before the first Envoy GeoCluster is defined, no check for a Default VLAN Failover IP address is performed when the GeoCluster is added. So, it is possible that the Default VLAN does not have a Failover IP address when the first GeoCluster is added to Envoy. If this occurs, Envoy will not function and the following messages are logged:

```
sibd|e| | |The Default VLAN must have a failover IP address assigned for Envoy||
sibd|w| | |Internal configuration error detected.||
sibd|w| | |Retry every 5 seconds.|
```

If the unit is rebooted while in this state, the systems will remain in the **initializing** failover mode.

To fix this issue, add a Failover IP address to the Default VLAN and reboot. After the system reboots, both Envoy and Failover will be enabled and working properly.

Using Envoy with NAT Devices

If an Envoy site is located behind a device (such as a firewall) that is performing Network Address Translation (NAT) on incoming IP addresses, then you must specify the public (non-translated) IP as the Site IP, and use the translated IP (the non-public IP) as the resource (cluster) IP in the Envoy configuration.

This is because Envoy must return the public cluster IP to a requesting client in order for the client to be able to contact that cluster -- since the request goes through the NAT device before it reaches Equalizer. The NAT device translates the public cluster IP in the request to the non-public cluster IP that is defined on Equalizer, and then forwards the packet to Equalizer.

The non-public cluster IP must still be specified as the resource IP for the site, as this is the IP that Envoy will use internally to probe the availability of the resource (cluster) on the site.

Upgrading a Version 7 GeoCluster to Version 8

Envoy in Version 8 is designed to work with existing sites running Version 7. You can upgrade a Version 7 site in-place to Version 8, and it will continue to operate seamlessly with other Version 7 sites in the GeoCluster. In order to work with resources located on other Version 8 sites, however, the configuration must be updated with the cluster name, as noted below:

1. Upgrade sites one at a time, starting with the non-default sites. Test thoroughly before upgrading the next site.
2. The resource (cluster) name for any resource that is located on a site running Version 7 of the Equalizer software must be left blank. Specify the cluster IP and port instead.
3. The resource (cluster) IP and port for any resource that is located on a site running Version 8 of the Equalizer software must be left blank. Specify the cluster name instead.

Working with GeoClusters

This section shows you how to add or delete a GeoCluster and how to configure a GeoCluster's load-balancing options. Configuring a GeoCluster and its sites is analogous to configuring a virtual cluster and its servers.

When Envoy is first enabled, there are no GeoClusters defined, so clicking on the **Envoy** icon in the left frame displays a blank GeoCluster Summary table. Below is an example of a GeoCluster Summary table with one GeoCluster and two GeoSites configured:

Use the icons in the **Actions** column below to add, delete, and modify GeoClusters.

Geo Cluster Name	Sites Status	Actions
www.example.com	2 0	

Figure 53 GeoCluster Summary

The table shows the GeoCluster name and the status of the sites in the cluster. The icons in the Actions column let you add, modify, and delete GeoClusters.

To see an expandable list of existing GeoClusters, click **Envoy > Status**:

Refresh

Status: Up Down

www.example.com					
Servers	Status	Weight	TimesChosen	TimesDown	
si00		100	0	0	
www.example.org					

Adding a GeoCluster

To add a new GeoCluster, follow these steps:

1. Log into the Administrative Interface using a login that has **add/del** access for Global Parameters (see “Logging In” on page 52).
2. Do *one* of the following:
 - In the left-frame object tree, right-click on **Envoy** and select **Add GeoCluster** from the menu.
 - Click **Envoy** in the left frame and then click the Add icon in the GeoCluster Summary table.

- The following dialog appears:


- Enter the following information:

FQDN name	Enter the GeoCluster name , which is the fully-qualified domain name (FQDN) of the GeoCluster (for example, <code>www.coyotepoint.com</code>). The FQDN must include all name components up to the top level (com, net, org, etc). Do not include the trailing period.
DNS ttl	The 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. The default is 120 (that is, 2 minutes).

- Click the **commit** button to add the GeoCluster. An entry for the new GeoCluster appears in the left frame. The right frame displays the GeoCluster Configuration screen.
- Continue with Step 3 in the next section to change the default GeoCluster parameters.

Viewing and Modifying GeoCluster Parameters

To view or modify a GeoCluster's load-balancing options, follow these steps:

- Log into the Administrative Interface using a login that has **read** (to view only) or **write** (to view or change) permission on the GeoCluster (see "Logging In" on page 52).
- Do *one* of the following:
 - Click on the GeoCluster name.
 - Click **Envoy** in the left frame and then click the Modify icon  in the GeoCluster Summary table row for the GeoCluster you want to modify.

3. The GeoCluster Configuration tab is displayed:

Geographic Cluster Configuration

default site warning

GeoClusters should contain at least one default site. Please set the default checkbox in one of the sites.

responsiveness medium ▾

DNS cache ttl 120

MX exchanger

policy adaptive ▾

ICMP triangulation

commit
show defaults
reset

The **default site warning** is displayed on this screen until you select a default site; see “Displaying and Modifying Site Information” on page 268.

The GeoCluster configuration parameters are explained in the table below:


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 <i>medium</i> setting as a starting point.
DNS cache ttl	The 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. The default is 120 (that is, 2 minutes).
MX exchanger	The fully qualified domain name (e.g., ‘mail.example.com’) to be returned if Equalizer receives a “mail exchanger” request for this GeoCluster. The mail exchanger is the host responsible for handling email sent to users in the domain. This field is not required.

policy	<p>Three basic metrics are used by the policy to load balance requests among sites: the current load on the site, the initial weight setting of the site, and ICMP triangulation responses. The policy setting tells Envoy the relative weight to assign to each metric when choosing a site.</p> <p>round trip weights the ICMP triangulation information received from each site more heavily than other criteria.</p> <p>adaptive give roughly equal weights to the site load and ICMP triangulation responses, and gives less weight to the initial weight for the site. This is the default setting.</p> <p>site load weights the current load at each site more heavily than other criteria.</p> <p>site weight weights the user-defined initial weight for each site more heavily than other criteria.</p> <p>Note: For all policies, the current site load metric is ignored for the first 10 minutes that the site is up, so that the metric value is a meaningful measure of the site load before it is used.</p>
ICMP triangulation	<p>When a request for name resolution is received by Envoy from a client's local DNS, this option (if enabled) tells Envoy to request network latency information from all sites in order to make load balancing decisions based on the proximity of each site to the client's DNS server.</p> <p>To do this, all Envoy sites send an ICMP echo request ('ping') to the client's DNS server. The reply from the DNS server allows Equalizer to select a site using the length of time it takes for the DNS server's reply to reach the site. (Consequently, this method assumes that the client's DNS server is geographically close to the client -- which is usually the case.)</p> <p>In order for triangulation data to be collected at each site, responding to ICMP echo requests must be enabled on the client DNS server and the DNS server must be allowed to respond through any firewalls between it and the Envoy sites.</p> <p>If you do not want Envoy sites to ping client DNS servers, disable this option (this is the default setting).</p>

4. Click the **commit** button to save any changes you made to the GeoCluster parameters.

Deleting a GeoCluster

To delete a GeoCluster, follow these steps:

1. Log into the Administrative Interface using a login that has **add/del** access for the GeoCluster (see "Logging In" on page 52).
2. Do *one* of the following:
 - In the left frame, right-click the name of the GeoCluster to delete and select **Delete GeoCluster** from the menu.
 - Click **Envoy** in the left frame and then click the Delete icon  in the GeoCluster Summary table row for the GeoCluster you want to delete.
3. When prompted, click **delete** to confirm removing the cluster. Equalizer deletes the GeoCluster and all its sites.

Displaying Envoy Statistics

See “Displaying Envoy Statistics” on page 203.

Plotting GeoCluster History







See “Plotting GeoCluster Performance History” on page 209.

Working with Sites

GeoSites, or Sites, are defined within GeoClusters, so before you can configure your first site, you must first have added a site as shown in the section “Adding a GeoCluster” on page 262.


Once you define a GeoCluster, you can open the GeoSite Summary table to add, modify, and delete GeoSites: click the GeoCluster name in the left frame, and then click the Sites tab in the right frame. The following is an example of a GeoSite Summary table with two sites configured:

Add, modify, and delete Sites

Geo Site Name	AAA IP	Agent IP	Status	Actions
si00	172.16.0.81	172.16.0.80		 
si01	172.16.0.142	172.16.0.140		 

The table shows the GeoSite Name, the DNS AAA Record IP address (this is the **A Record IP Address** specified when you created the site), and the Agent IP address supplied when the sites were created. It also displays the current Status of each Site. The icons in the Actions column let you add, modify, and delete GeoSites.

Adding a Site to a GeoCluster

1. Log into the Administrative Interface using a login that has **add/del** access for the GeoCluster (see “Logging In” on page 52).
2. Do *one* of the following:
 - In the left frame, right-click the name of the GeoCluster to which you want to add a Site, and select **Add Site** from the menu.
 - Click the GeoCluster name in the left frame and open the **Sites** tab. Click the Add icon  at the bottom of the GeoCluster Summary table **Action** column.

3. The following dialog is displayed:

Add New GeoSite ? X

In order to add a new site, please fill out the following required information. You will then be taken to a detailed site view, where you can select advanced options.

GeoSite Parameters

GeoSite Name:

A Record IP Address:

Agent IP Address:

Site Version: V8 V7

Resource Name:


4. The GeoSite Parameters are described in the following table:

Site Name	A symbolic name that represents this site. For example, the east coast site for <code>www.coyotepoint.com</code> might be <code>eastCOAST</code> .
A Record IP Address	The IP address returned by DNS to a client when the GeoCluster is accessed. For example, when a client opens <code>www.coyotepoint.com</code> , the local DNS server returns an A record that contains the IP address for <code>www.coyotepoint.com</code> . This is usually the address of an Equalizer cluster and in this case is also used as the resource IP. However, the site's A record IP may be different from the cluster (resource) IP if the A record IP address is NAT'ed to an internal address (the actual cluster IP). In this case, you specify the A record IP as the site IP and the cluster IP as the resource IP.
Agent IP Address	The IP address of the site monitoring agent. This is the external interface address of the Equalizer at this site; if the Equalizer is in single network mode, this is the internal interface address.
Site Version	Click V8 if the Site is running Version 8 or later of the Equalizer software. Enter a Resource Name that is same as the name of a cluster configured on the Equalizer at this site. Click V7 if the Site is running Version 7 of the Equalizer software. Enter an IP Address and Port for a cluster configured on the Equalizer at this site.

5. Click the **commit** button to add the Site. An entry for the new Site appears in the left frame. The right frame displays the Site Configuration screen. Continue with the next section to update the default Site and Resource parameters.

Displaying and Modifying Site Information

To view or modify the information for a particular GeoSite, follow these steps:

1. Log into the Administrative Interface using a login that has **read** (to view only) or **write** (to view or change) permission on the Site's GeoCluster (see "Logging In" on page 52).
2. Do *one* of the following:
 - In the left-frame object tree, expand the Envoy tree until the name of the Site is visible. Click the Site name.
 - Click the GeoCluster name in the left frame and open the **Sites** tab. Click the Modify icon  on the row for the GeoSite you want to modify.

The Site Configuration tab is displayed.

Site Configuration

ip	172.16.0.181
agent	172.16.0.180
weight	100
default site	<input type="checkbox"/>

commit
show defaults
reset

The **Site Configuration** parameters are explained in the table below:

ip	The IP address returned by DNS when the GeoCluster is accessed. For example, when a client open <code>www.coyotepoint.com</code> , the local DNS server returns an A record that contains the IP address for <code>www.coyotepoint.com</code> . This is usually the address of an Equalizer cluster and in this case is also used as the resource IP. However, the site's A record IP may be different from the cluster (resource) IP if the A record IP address is NAT'ed to an internal address (the actual cluster IP). In this case, you specify the A record IP as the site IP and the cluster IP as the resource IP.
agent	The IP address of the site monitoring agent. This is the external interface address of the Equalizer at this site; if the Equalizer is in single network mode, this is the internal interface address.

weight	<p>An integer that represents the site's capacity. (This value is similar to a server's initial 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.</p> <p>Equalizer uses a site's initial weight as the starting point for determining what percentage of requests to route to that site. Equalizer assigns sites with a higher initial weight a higher percentage of the load. The <i>relative</i> values of site initial weights are more important than the actual values. For example, if two sites are in a GeoCluster and one has roughly twice the capacity of the other, setting the initial weights to 50 and 100 is equivalent to setting the initial weights to 100 and 200.</p> <p>Dynamic site weights can vary from 50% to 150% of the assigned initial weights. To optimize GeoCluster performance, you might need to adjust the initial weights of the sites in the cluster based on their performance.</p> <p>Site weights can range from 10 to 200. When you set up sites in a GeoCluster, you should set each site's initial weight value in proportion to its capacity for handling requests. It is not necessary for all of the initial weights in a cluster to add up to any particular number.</p>
default site	<p>Designates this site as the default site for the GeoCluster. Envoy load balances to the default site whenever it cannot choose a site based on the GQP probe information it gets from the sites. This can happen, for example, when GQP probe responses are not received from any site, when the resource (cluster) is down at all available sites, etc. If no default site is selected for a GeoCluster and all sites are down, then Envoy sends a null response to the client DNS. See the explanation in Step 4b on page 257.</p>

3. Click the **commit** button to save any changes you made to the Site configuration.
4. Click on the **Resources** tab, to update the following resource parameters:

resource configuration

Site Version

V8

V7

name

If **V8** is selected, the following parameter is displayed:

name	<p>If the Equalizer at this site is running Version 8 or higher of the Equalizer software, specify the cluster name. Equalizer will query the Envoy agent at that site for the cluster's IP address and port. Leave blank if the site is running Version 7.</p>
-------------	---

If **V7** is selected, the following parameters are displayed:


ip	<p>If the Equalizer at the site is running Version 7 of the Equalizer software, specify the cluster's IP address (and port, below). It is generally the same value as the site IP address, unless the site address is NAT'ed to a cluster IP. Leave blank if the site is running Version 8.</p>
-----------	---

port	If the Equalizer at the site is running Version 7 of the Equalizer software, specify the cluster's TCP port number (and IP address, above). Leave blank if the site is running Version 8.
-------------	---

5. Click the **commit** button to save any changes you made to the resource configuration.

Deleting a Site from a GeoCluster

To delete a Site from a GeoCluster, follow these steps:

1. Log into the Administrative Interface using a login that has **add/del** access for the GeoCluster (see “Logging In” on page 52).
2. Do *one* of the following:
 - In the left frame, right-click the name of the Site to delete and select **Delete Site** from the menu. (You may need to expand the GeoCluster first to see the Sites.)
 - Click the GeoCluster name in the left frame and open the **Sites** tab. Click the Delete icon  on the row for the GeoSite you want to delete.
3. When prompted, click **delete** to confirm removing the Site. Equalizer deletes the Site and removes it from the object tree.

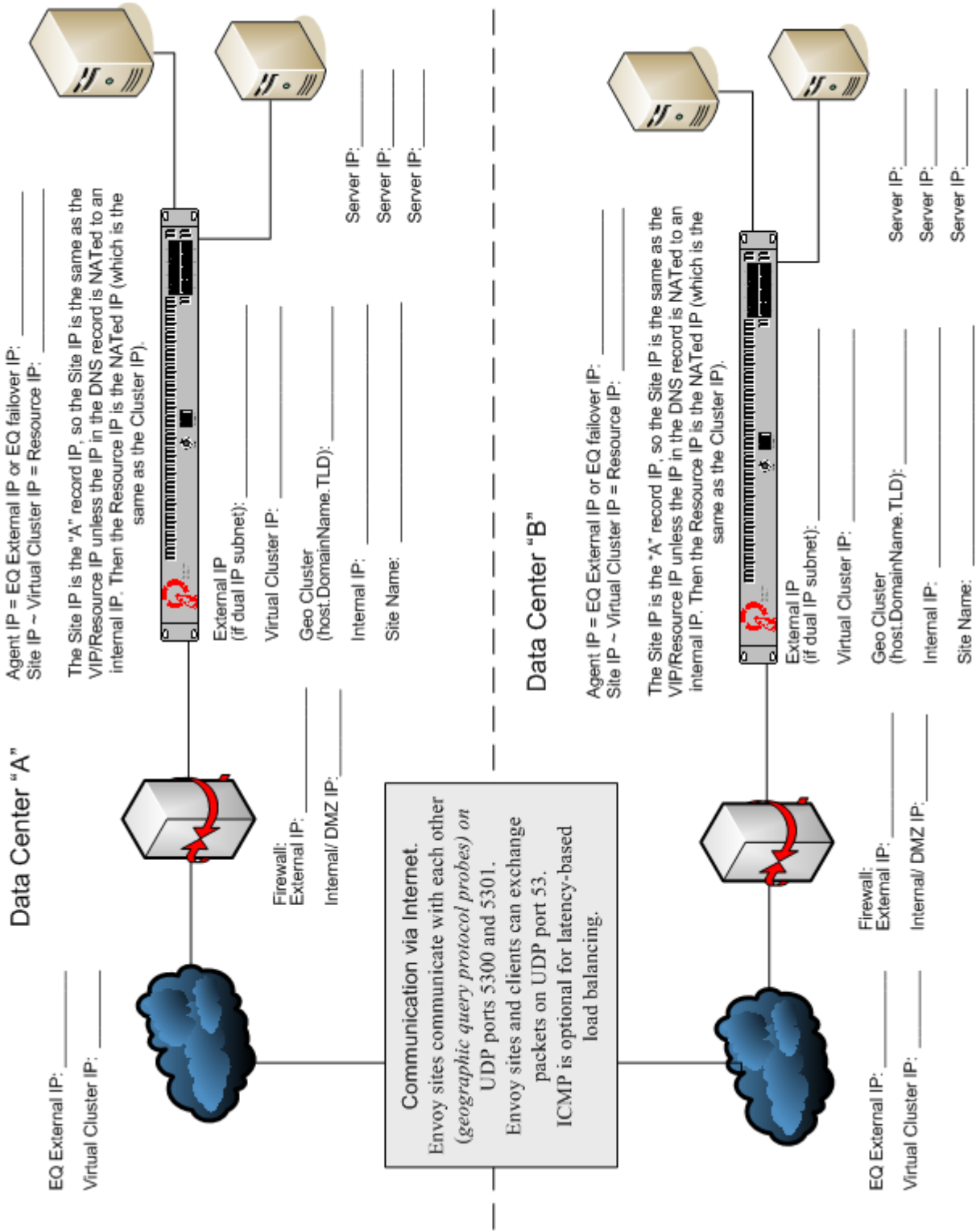
Displaying Site Statistics

See “Displaying Site Statistics” on page 204.

Plotting Site History

See “Plotting Site Performance History” on page 209.

Envoy Configuration Worksheet





Server Agent Probes

Using Server Agents273

 Enabling Agents273

 Server Agents and Load Balancing Policies274

 Server Agents and Server 'Down' Conditions274

 Sample Server Agent in Perl274

Using Server Agents

A server agent is a custom written program that runs on a server and provides direct feedback to Equalizer that is used by the load balancing algorithms. This feedback is obtained by the agent by any means available on the target server; the only requirement from Equalizer’s point of view is that the agent response is in the form of an integer between -2 and 100 that represents the status of the server and/or the application that is running on it.

100 to 0	100 indicates that the server and/or application is lightly loaded. 0 indicates that the server and/or application is heavily loaded.
-1	The application, or a required resource (such as a database), is unavailable.
-2	The server agent cannot determine the status of the application. This is the default return value used by Equalizer when an agent does not respond.
Note: Server agent code written prior to Version 8.1.0a must be adjusted to reflect the server agent return values and interpretations shown above. In particular, in previous releases the meanings of the 0 to 100 range of values were documented as the reverse of the meanings shown above.	

After returning a value to Equalizer, the agent should close the port and wait for another connection.

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.

Enabling Agents

Agents are enabled for a cluster by turning on the **server agent** cluster flag. The default **agent port** is **1510**. Make sure that any agent you deploy is listening and able to respond to TCP connections on the same port number on all the servers in the cluster.

The time between server agent probes is determined by the **agent delay** global parameter (default is 10 seconds).

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 **agent probe** field is provided for this purpose. If a string appears in this field, it is sent to the agent when an agent probe occurs.

Server Agents and Load Balancing Policies

Server agents work with all load balancing policies (see “Equalizer’s Load Balancing Policies” on page 137), except for **round robin** -- which simply ignores any agent response for all servers in the cluster. All 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 a server agent over other load balancing factors (server weight, number of current connections, etc.).

Server Agents and Server ‘Down’ Conditions

Note that there is no return code for 'server down' in the table above. This is because Equalizer normally relies on the other health check probes (ICMP, TCP, and ACV probes) to determine whether the server is up. So, even if the server agent responds with “-1” or “-2”, this by itself will not cause Equalizer to mark the server down.

To change this default behavior, enable the global **require agent response** flag (see “Modifying Global Parameters” on page 89). When this global flag is enabled *and* server agents are enabled for a cluster, then Equalizer will mark a server in the cluster ‘down’ if either of the following are true:

- Equalizer does not get a response from the server agent running on the server before the probe timeout elapses
- Equalizer receives either a ‘-1’ or ‘-2’ response from the server agent running on a server

Sample Server Agent in Perl

You can write custom agents as shell scripts, or in Java, Perl, C, or other languages. The code below is a simple server agent example written in Perl. This code assumes that an integer response value is supplied on the command line, and returns that value when a connection is made on port 1510 (configurable via the `$port` variable). This sample agent is intended for testing purposes only. In a real deployment, the server agent would determine the response value to return by polling system resources, or some other real-time method.

```
#!/usr/bin/perl -w
# serveragent.pl
#-----
#(c) Copyright 2008 Coyote Point Systems, Inc.

use strict;
use Socket;

# use port 1510 as default
my $port = 1510;
my $proto = getprotobyname('tcp');

# take the server agent response value 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 the server agent response value
print CLIENT $response;

# close connection
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]

```

Another “Connection” line prints each time the server agent is probed by Equalizer.

From Equalizer’s perspective, all that is returned by the server agent is the integer set on the command line. For example, if you use the example server agent above and set the response to “50”, here is what you will see if you use the **telnet** command to open the server agent IP and port:

```

$ telnet 10.0.0.120 1510
50
Connection to host lost.

```


Timeout Configuration



Timeouts ensure that certain operations are carried out within a finite period of time, and the resources that they use are returned for re-use. This document describes the various timeout parameters used by Equalizer, which can be divided into two major groups:

- **connection timeouts** -- used by Equalizer to manage connections to the clients on the network and the servers in clusters
- **probe timeouts** -- used by Equalizer to manage the various server health check mechanisms that assess server availability

Connection Timeouts	278
HTTP and HTTPS Connection Timeouts	278
The Once Only Option and HTTP / HTTPS Timeouts.....	281
Layer 4 Connection Timeouts	281
Application Server Timeouts	282
Connection Timeout Kernel Variables	282
Server Health Check Probes and Timeouts	283
ICMP Probes	283
High Level TCP and ACV Probes	283
TCP Probe Aggregation	286
Server Agent Probes	287
Agent Probe Process	287
Enabling and Disabling Server Agents.....	287

Connection Timeouts

Layer 7 clusters (HTTP / HTTPS) and Layer 4 clusters (TCP / UDP) each use a different set of timeout parameters. These are discussed in the sections below.

HTTP and HTTPS Connection Timeouts

Connections to HTTP and HTTPS clusters are managed closely by Equalizer from the client request to the response from the server. Equalizer needs to manage two connections for every Layer 7 connection request: the client connection from which the request originates, and the connection to the server that is the final destination of the request (as determined by the load balancing policy).

Equalizer has an idle timer for the established client connection, a connect timer to establish a server connection, and an idle timer for the established server connection. Only one timeout is in use at any given time. This is a summary of how timeouts are used when a client connects to Equalizer:

1. When a client successfully connects to a Virtual Cluster IP, the **client timeout** applies from the time the connection is established until the client request headers are completely transmitted. Equalizer parses the client's request, and verifies that the request is a valid HTTP request and that the information needed for load balancing is obtained. In general, this happens at the time that the client headers are completed -- which is indicated by the client sending two blank lines for HTTP 1.0 or 1.1; one blank line for HTTP 0.9. Once the headers are completely transmitted to Equalizer, the **client timeout** is no longer used.
2. As soon as the Equalizer is done examining the header data, it makes a connection to a server, as determined by the load balancing policy, persistence, or a match rule hit. The amount of time that the Equalizer tries to establish a connection to the server is the **connect timeout**. Once the server connection is established, the **connect timeout** is no longer used.
3. After Equalizer establishes a connection with a server, the **server timeout** is the amount of time Equalizer waits for the next bit of data from the server. Any response from the server restarts the **server timeout**.

The important distinction between the **client timeout** and the **server timeout** is that the **client timeout** is a “hard” timeout -- the client has the number of seconds specified to transmit all of its headers to Equalizer before Equalizer times out. This is done mainly for security considerations to prevent malicious clients from creating a large number of partial connections and leaking data slowly over the connection, possibly causing resource exhaustion or other undesirable effects on Equalizer.

The **server timeout** by contrast is a “soft” timeout -- the server has the number of seconds specified to send *the next piece of information* (e.g., the next packet in the sequence). Whenever the client or the server sends a piece of data on the connection, the **server timeout** is reset. This allows the server to send large data streams in small pieces without timing out, and then close the connection once all the data is sent.

For example, when a client sends a POST operation in a request, the **client timeout** is used up until the time that the POST *headers* have all been received. The **connect timeout** is used until a connection with the server is established. Then, once the connection is established, the **server timeout** is used for the POST data itself and the subsequent response from the server.

Note that there is the chance that a client will connect, send its headers, and then send continuous data to Equalizer that repeatedly resets the **server timeout**. This vulnerability is usually avoided by setting a hard client timeout on the application server itself (see “Application Server Timeouts” on page 282).

Figure 54 summarizes the connection timeout parameters Equalizer uses for Layer 7 client and server connections.

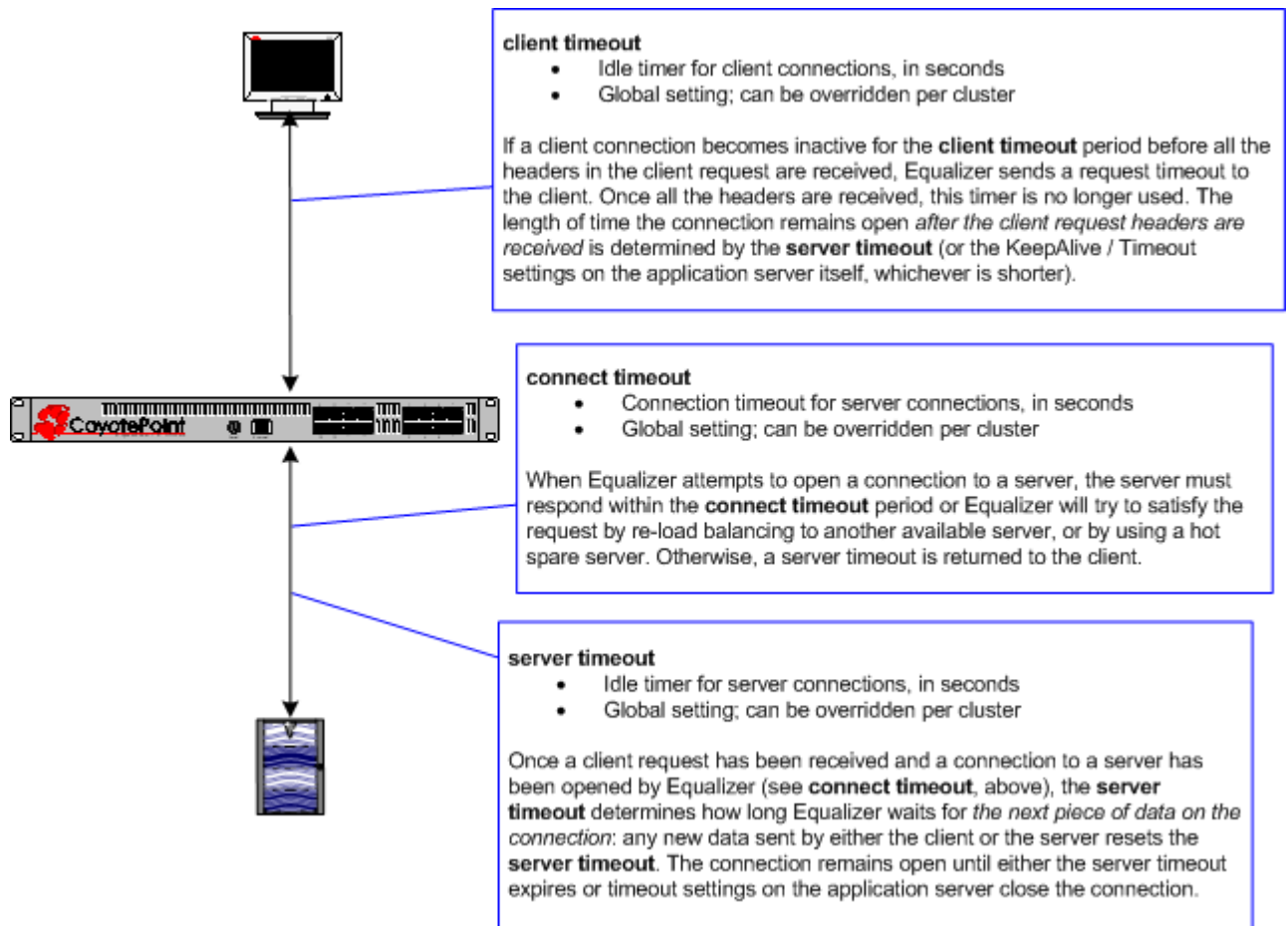


Figure 54 Layer 7 connection timeout parameters

The timeline below shows the sequence of timeout events when a new connection is received by Equalizer.

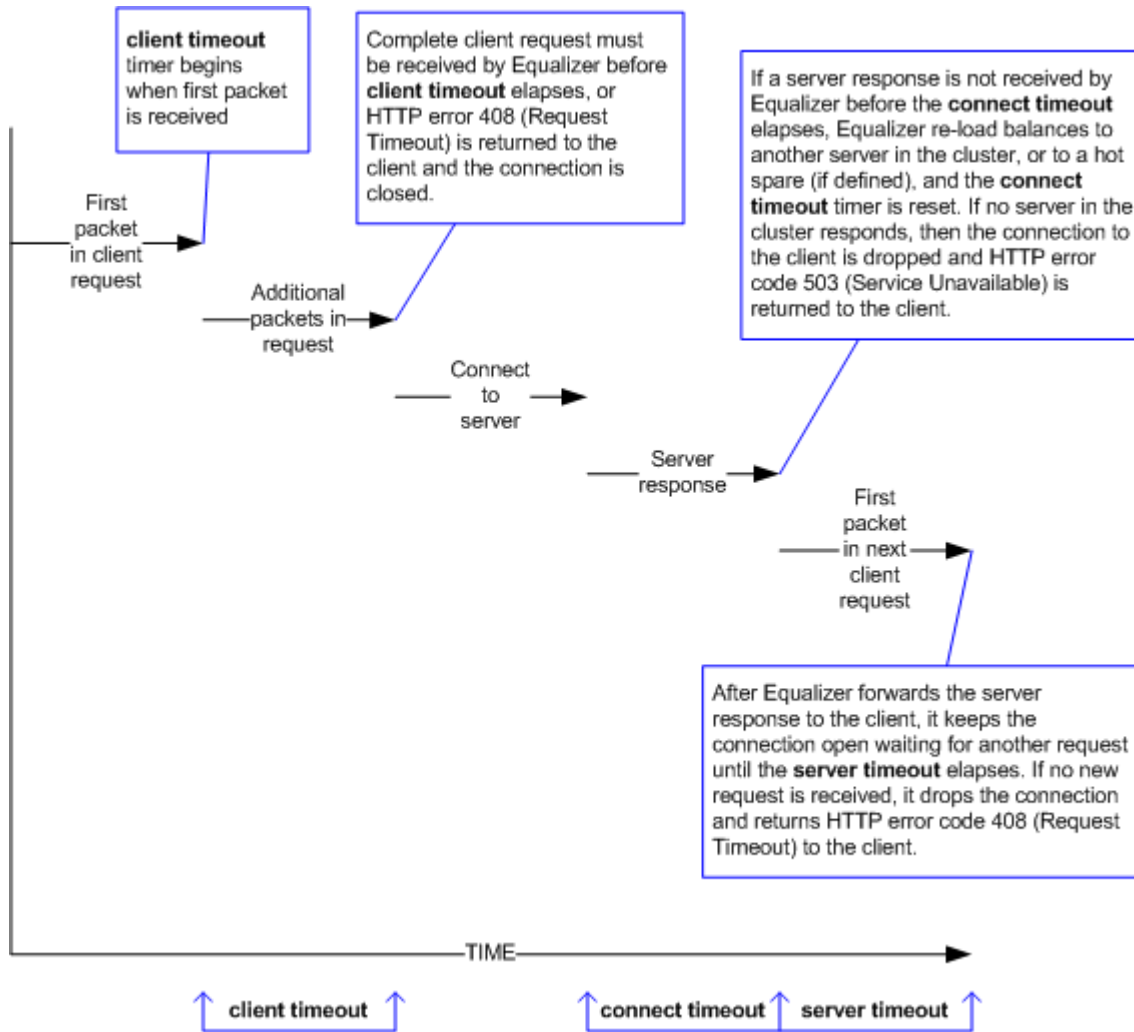


Figure 55 Layer 7 connection timeline

The following table shows the value range for the Layer 7 HTTP / HTTPS connection timeouts.

Parameter	Minimum	Default	Maximum	Units
client timeout	1.0	5.0	64535.0	seconds
server timeout	1.0	60.0	2147483647.0	seconds
connect timeout	1.0	10.0	60.0	seconds

The default timeout values are sufficient for many common applications. If timeouts are occurring using the default values, adjust the **server timeout** to the amount of time you expect your application server to respond to a client request, plus 1 second. If there is high latency between Equalizer and the servers in your cluster, then you may need to increase the **connect timeout**. The **client timeout** usually does not need to be changed, but in some situations, HTTPS clusters will require a client timeout between 15 and 30 seconds for best performance. If you do need to increase the **client timeout**, use the lowest value possible for your configuration to perform well; high values for **client timeout** increase the risk of denial of service (DoS) attacks.

The Once Only Option and HTTP / HTTPS Timeouts

The previous sections describe how the connection timeouts work when the **once only** flag is *disabled* on a cluster; that is, when Equalizer is examining *every* set of headers received on a connection. The **once only** option, when enabled, specifies that Equalizer will examine only the *first* set of headers received on a connection. This has the following effects on connection timeouts:

- If you have **once only enabled**, as soon as the initial transaction (client request and server response) on a connection completes, the connection goes into “streaming” mode and the **client timeout** is no longer used for this connection. Equalizer does not parse any additional client requests received on the connection. The **server timeout** is used for the remainder of the connection, and is reset whenever data is received from either side of the connection.
- If you have **once only disabled** as described in the previous sections, and multiple requests are being sent on the same connection, the **client timeout** starts counting down again as soon as a new request is received from the client.

Layer 4 Connection Timeouts

Connections to Layer 4 clusters are received by Equalizer and forwarded with little processing. Equalizer simply rewrites the source and/or the destination IP addresses, as appropriate for the cluster, and sends the packet to the server specified by the cluster’s load balancing policy. For Layer 4 TCP clusters, a *connection record* is kept for each connection so that address translation can be done on the packets going between the servers and clients. The Layer 4 connection timeouts specify how long a connection record is kept by Equalizer.

Layer 4 TCP clusters use the **idle timeout** and **stale timeout** parameters. The **idle timeout** can be set at the global and cluster levels, while **stale timeout** can be set at the global level only. The parameters affect how Equalizer manages Layer 4 connection records:

- Connection records need to be removed in cases where the connection is not closed by the client or server, and is left idle. If no data has been received on a connection from either the client or the server after the time period specified by the **idle timeout** has elapsed, then Equalizer removes the connection record for that connection. Any data received from either client or server resets the idle timer.

Note that when using Direct Server Return (DSR), the time that a connection record is maintained is determined by adding the **idle timeout** for the cluster to the **sticky time** (see “sticky time” on page 135). This additional time is necessary when using DSR, since no server responses are routed through Equalizer (and therefore cannot restart the **idle timeout** to keep the connection open). For more information on DSR, see “Configuring Direct Server Return (DSR)” on page 188.

- In other cases, a connection may be initiated but never established, so the connection record goes “stale” and must be removed. 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*. The **stale timeout** is the length of time that a connection record for an incomplete connection is maintained.

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. (Equalizer does *not* send a TCP RST to the client when reclaiming a connection.)

Reducing the **stale timeout** can be an effective way to counter the effects of SYN flood Denial of Service attacks on server resources. A **stale timeout** of 10.0 (see table below) would be an appropriate value for a site under SYN flood attack.

Parameter	Minimum	Default	Maximum	Units
idle timeout	0	0	2147483647.0	seconds
stale timeout	1.0	15.0	120.0	seconds

Note that if you change the **stale timeout** setting while partially established Layer 4 connections are currently in the queue, those connections *will* be affected by the new setting.

Application Server Timeouts

Keep in mind that the application server running on the physical servers in your cluster may have its own timeout parameters that will affect the length of time the server keeps connections to Equalizer and the client open. For example, an Apache 2 server has two related timeout directives: **TimeOut** and **KeepAliveTimeout**:

- The **TimeOut** directive currently defines the amount of time Apache will wait for three things:
 - The total amount of time it takes to receive a GET request.
 - The amount of time between receipt of TCP packets on a POST or PUT request.
 - The amount of time between ACKs on transmissions of TCP packets in responses.
- The **KeepAliveTimeout** directive specifies the number of seconds Apache will wait for a subsequent request before closing the connection. Once a request has been received, the timeout value specified by the **Timeout** directive applies.

In general, if you want Equalizer to control connection timeouts, you must make sure that any timeouts set on the application server are of longer duration than the values set on Equalizer.

For example, with respect to the Apache server timeouts above, the **client timeout** (for Layer 7 connections) or the **idle timeout** (for Layer 4 connections) should be of shorter duration than the timeouts set for Apache.

Similarly, the Layer 7 **server timeout** and **connect timeout** on Equalizer should be of shorter duration than the TCP connection timeouts set on the servers.

Connection Timeout Kernel Variables

Equalizer uses a number of kernel variables to track connection timeouts, as shown in the table below. You can use the **sysctl** command to display kernel variables. The two basic formats of this command are:

```
sysctl variable_name    Displays the kernel variable variable_name.
sysctl -a > file        Displays all kernel statistics. This is a long list, so we recommend capturing the
                        list to a file.
```

eq.idle_timeout	The current setting of the Layer 4 global networking idle timeout parameter.
eq.idle_timedout_count	A Layer 4 counter incremented when a connection is terminated because the idle timeout expired.
eq.stale_timeout	The current setting of the Layer 4 global networking stale timeout parameter.
eq.l7lb.timeouts	The total number of Layer 7 connections dropped because a connection timer expired.
eq.l7lb.http.client_timeouts	The total number of Layer 7 (HTTP and HTTPS) connections that were terminated because the client timeout expired.
eq.l7lb.http.connect_timeouts	The total number of Layer 7 (HTTP and HTTPS) connections that were terminated because the connect timeout expired.
eq.l7lb.http.server_timeouts	The total number of Layer 7 (HTTP and HTTPS) connections that were terminated because the server timeout expired.

Note that there are also some kernel variables associated with Secure Socket Layer (ssl) client connections, such as when someone logs into Equalizer over an SSH connection. These variables are *not* incremented by HTTPS connections:

```
eq.171b.ssl.total_clients
eq.171b.ssl.current_clients
eq.171b.ssl.max_clients
eq.171b.ssl.requests
```

Server Health Check Probes and Timeouts

There are four levels of server health check probes supported by Equalizer:

- ICMP probes; all cluster types, enabled by default
- High Level TCP Probes; all cluster types, enabled by default
- High Level ACV (Active Content Verification) Probes; all cluster types except Layer 4 UDP clusters, disabled by default
- Server Agent Probes; all cluster types, disabled by default

ICMP Probes

By default, Equalizer sends an Internet Control Message Protocol (ICMP) echo request (commonly called a “ping”) to the IP address of every server in every cluster.

The delay between successive ping requests to the same server is determined internally, but can be as short as one second on a server that is not responding to ICMP requests. On a lightly loaded Equalizer it may be 5 seconds or longer.

If a server does not respond to an ICMP echo request, Equalizer continues to issue any other probes (TCP, ACV, server agent) configured for the cluster. This means, for example, that if TCP and ICMP probes are both configured (the default), then a server can fail any number of ICMP probes and will still be marked *up* as long as it continues to respond to TCP probes.

If a server does not respond to an ICMP echo request and no other probes are configured, the server is marked *down*, and Equalizer continues to send ICMP requests to the server’s IP address. If an ICMP echo response is subsequently received, the server is marked *up* again.

ICMP probing can be turned off by disabling the **ICMP probe** flag in the global parameters. This turns ICMP echo requests off for all clusters. (ICMP probes do not use any of the timeouts and parameters defined in the following section for High Level Probes.)

Note – Responding to ICMP echo requests is an option on most server platforms. If ICMP echo reply is disabled on one or more of the servers your configuration, then you may want to disable ICMP echo requests on Equalizer to reduce traffic between Equalizer and the servers, and rely solely on the other probing mechanisms.

High Level TCP and ACV Probes

Equalizer sends High Level Probes to every server at the interval specified by **probe delay** (default: 10 seconds). By default, TCP probes are enabled for all servers, and ACV probes can be enabled for individual clusters. Both probes must complete within the same **probe timeout** period, and are controlled by the same set of parameters, as summarized in the following figure.

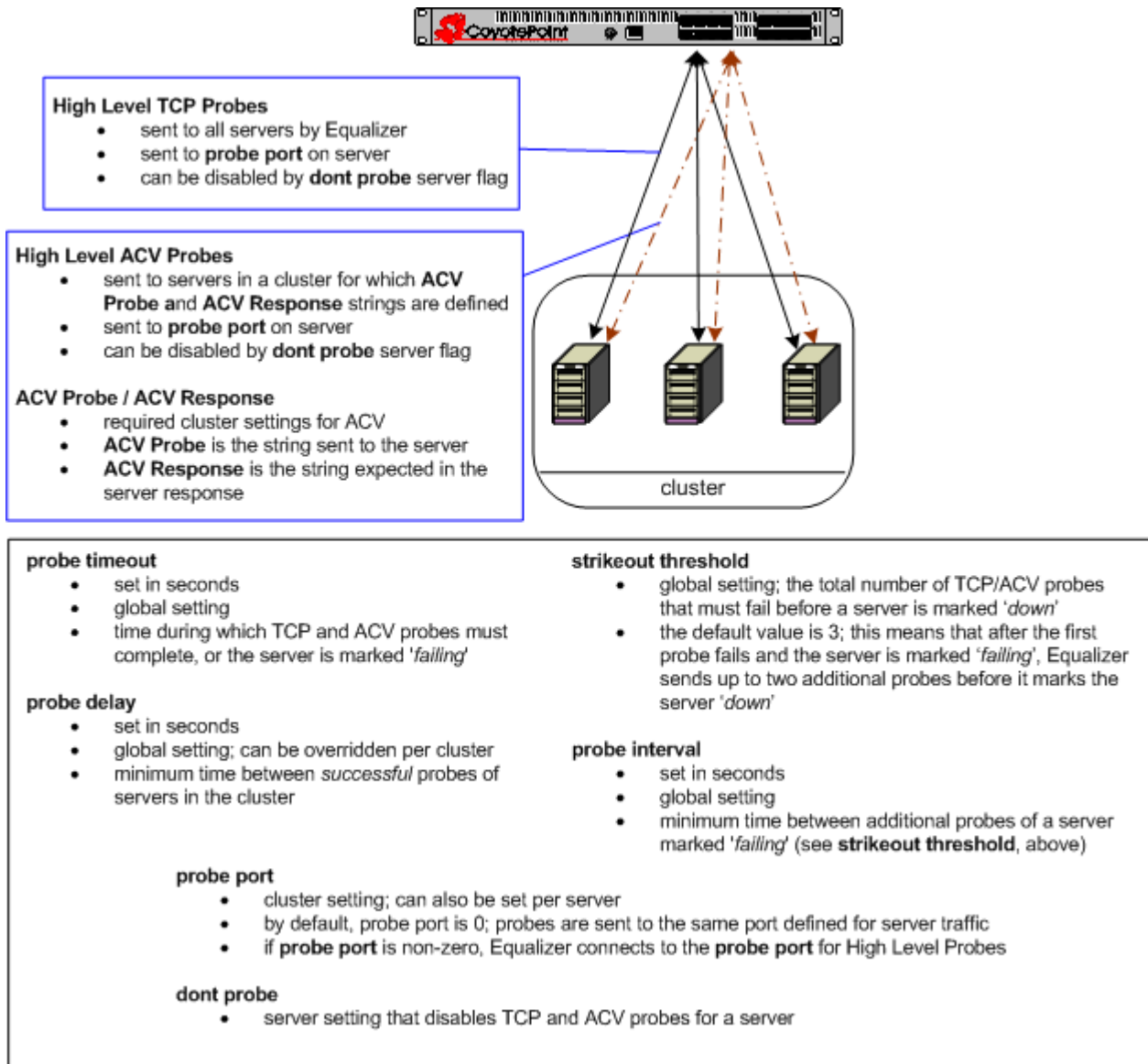


Figure 56 Probe timeout parameters

The parameters shown in the figure above determine how high level TCP and ACV server probes are handled, as follows:

- Equalizer begins a TCP probe by sending a TCP SYN packet to the server:
 - If the server and Equalizer negotiate a three-way TCP handshake (SYN, SYN/ACK, ACK) within the **probe timeout** period, go to **Step 2**.
 - If the server and Equalizer do not handshake within the **probe timeout** period, Equalizer marks the server *failing*; go to **Step 3**.
- Equalizer then determines whether or not to send the server an ACV probe:
 - If an **ACV Response** string is *not* defined for the cluster to which the server belongs, Equalizer marks the server *up* and waits for the **probe delay** period before it starts the HLP process again at **Step 1**.
 - Otherwise, if an **ACV Response** string is defined for the cluster to which the server belongs, the **ACV Probe** string (if defined) is sent to the server on the TCP connection opened for high-level probing; if there is no ACV probe string, nothing is sent, only the connection is opened. In either case, Equalizer

examines the first 1024 characters for the ACV response string: Since this is done as part of the same connection as the TCP probes, the same **probe timeout** period also applies to the ACV probe (i.e., the **probe timeout** timer is not reset):

- If the server response is received before the **probe timeout** expires *and* contains the ACV response string, it is marked *up*, and Equalizer waits for the **probe delay** period before it starts the HLP probing process again at **Step 1**.
 - Otherwise, if the server does not respond to the ACV probe before the **probe timeout** expires, or if the server responds but its response does not contain the ACV response string, then Equalizer marks the server *failing* and starts sending *strikeout* probes; go to **Step 3**.
3. This step is only performed when a server does not respond to a TCP or ACV probe within the **probe timeout**, and is marked *failing*. Before marking a *failing* server as *down*, Equalizer sends additional probes until the number of probes sent equals the value of the **strikeout threshold** parameter. The time between these additional probes is specified by the **probe interval** parameter:
- If the failing server does not respond to any of the *strikeout* probes, it is marked *down*. Equalizer then continues sending TCP probes to the server using **probe interval** as the minimum delay between probes. If a response is ever received, Equalizer marks the server *up* and waits for the **probe delay** period before it starts the probing process again at **Step 1**.
 - If a failing server responds to one of the *strikeout* probes, Equalizer marks the server *up* and waits for the **probe delay** period before it starts the probing process again at **Step 1**.

The following figure shows the relationship between the **probe timeout** and **probe delay** parameters in a successful probing sequence.

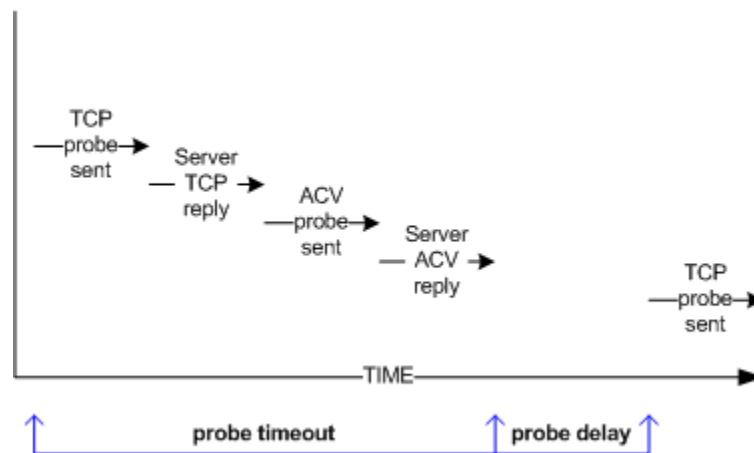


Figure 57 Successful probe sequence timeline

A server must respond to both TCP and ACV probes before the **probe timeout** elapses. Therefore, when ACV probes are enabled and probe failure messages are showing up in the log, you may need to increase the **probe timeout** so that the server has sufficient time to respond to both probes.

Assuming that a server responds successfully to TCP and ACV probes, Equalizer then waits for the **probe delay** time period before it sends the next TCP probe to the same server. (Note that the **probe delay** value is the *minimum* time between successful probes; the observed time may be longer for large configurations with many servers, during periods of high traffic, or due to Equalizer adjusting the delay internally to prevent server probes from consuming too much bandwidth on the network interface.)

In a network configuration where there is high latency between server probes and responses, the probe mechanism may falsely report that a server is down; this is indicated by messages in the event log indicating that a server is down and then comes back up again after a short period of time. In such cases you may need to increase the **probe timeout** or the **probe delay** parameters (or both) to reduce the number of false server down conditions reported by the probing mechanism.

The figure below shows the relationship between the **probe timeout** and **probe interval** parameters when a server does not respond to a High Level Probe.

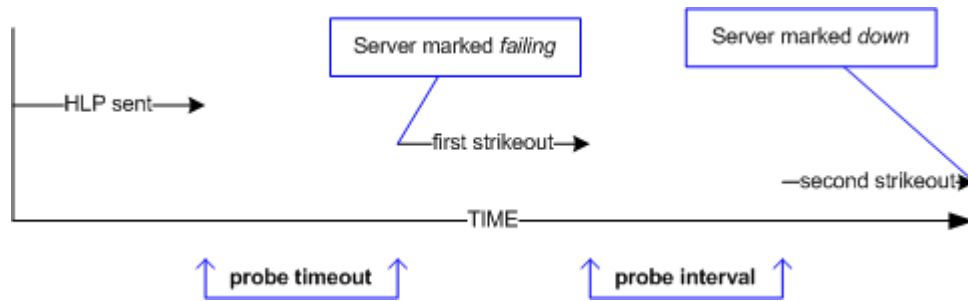


Figure 58 Unsuccessful probe timeout timeline

In the figure above, a High Level Probe (HLP) is sent to a server, which does not respond before the **server timeout** elapses. Equalizer marks the server as *failing* and sends two additional HLP probes (the default value of **strikeout threshold** is 3).

The **probe interval** specifies the time between these additional probes. If the server does not respond to either of these probes, it is marked *down*.

Note that the time periods between probes specified by the **probe interval** and **probe delay** values are *minimum* times. The observed time may be longer for large configurations with many servers, during periods of high traffic, or due to Equalizer adjusting the delay internally to prevent server probes from consuming too much bandwidth on the network interface. In addition, settings below 5 have the same effect as a setting of 5, since the Equalizer probe daemon cycles through the server probes every 5 seconds.

The range of values for each HLP parameter is shown in the table below. These apply to TCP and ACV probes only:

Parameter	Minimum	Default	Maximum	Units
probe timeout	1.0	10.0	60.0	seconds
probe delay	0.0	10.0	60.0	seconds
probe interval	0.5	20.0	25.0	seconds
strikeout threshold	1	3	6	integer

TCP Probe Aggregation

If a server is defined in more than one cluster and ACV probing is not enabled on any of the clusters to which the server belongs, then probes for that server are aggregated -- meaning, Equalizer only sends the server one TCP probe during each probe cycle, instead of sending one probe for each cluster. This reduces redundant probing.

Once ACV probing is enabled in a cluster, however, probe aggregation is disabled for all the servers in that cluster, and for all instances of these servers in other clusters.

For example, assume we have three clusters: **A**, **B**, and **C**. Let's say the same server, **sv01**, is defined in all three clusters. If ACV probing is not enabled in any of the three clusters, then **sv01** will be probed only once during each probe cycle, and the probe status will be reflected in all the clusters. Let's say we then enable ACV probing in cluster **A**. Thereafter, **sv01** will be probed independently for each cluster -- that is, Equalizer will probe **sv01** three times, once for each cluster **A**, **B**, and **C**.

It is possible, therefore, that defining a single server in many clusters and subsequently enabling ACV probes in any one of these clusters may result in a noticeable increase in probe traffic on the network.

Server Agent Probes

A server agent is a custom written application that runs on a server and listens on a specific port (default: 1510). When a connection request is received on that port, the server agent returns an integer value between -1 and 100 that indicates the relative load on the server (-1 meaning the server should be considered unavailable, 0 meaning very lightly loaded, and 100 meaning heavily loaded). Server agents can be used with any cluster type, and have an effect on all load balancing policies except **round robin**, which ignores server agent return values.

By default, server agents are disabled on all new clusters. To enable server agents for a cluster, you need to write the agent, install and run it on each server in the cluster, and then enable server agents for the cluster on Equalizer.

Agent Probe Process

When Equalizer connects to the port on which the server agent is running, it uses the number returned by the agent in its load balancing calculations, with the **server agent** policy giving highest preference to the server agent's return value over other factors.

The number returned by the agent to Equalizer is intended to indicate the current load on the server. The agent application that runs on the server can be written in any available scripting or programming language and can use any appropriate method to determine server load. The result must be an integer between -1 and 100 returned on the **server agent port**.

When enabled, server agents should be running on all servers in the cluster; however, by default, a server is not marked *down* when an agent value is not returned. Equalizer continues load balancing without the server agent return value unless the cluster parameter **require agent response** is enabled; if it is, Equalizer must receive an agent response or the server is marked *down*.

Note that server agent probing does not use any of the timeout values defined in the previous sections for High Level Probes. For example:

- The period of time between server agent probes to a server can be as short as one second. To introduce a timed delay, introduce a delay into the server agent code (for example, sleeping for 20 seconds). This does have the disadvantage of leaving the server agent port connection open for at least the length of the delay, but does reduce the frequency of agent probes.
- The period of time that Equalizer will wait for an agent response before marking it down is determined internally by Equalizer and cannot be adjusted by the administrator.

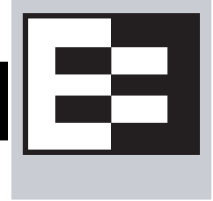
Enabling and Disabling Server Agents

Server agents are enabled for a cluster by turning on the **server agent** cluster flag, which sets the **server agent port** parameter to the default value of port **1510**. A connection to the server agent is opened on the **server agent port** specified up to every second -- depending on the cluster configuration, system load, and whether or not the server agent itself introduces a delay.

The **agent probe** cluster parameter specifies an optional string that is sent to the **server agent port** by Equalizer when it open a connection. This is not used by default, but is provided for those agents (such as agents written in Java) that require input before they reply to the probe. Agents written in C or perl, for example, usually don't require input in order to return the agent value.

Server agent probing is disabled by setting the **server agent port** parameter to **0**. Disabling the **server agent** flag automatically sets the port to 0.

Using Reserved IP Addresses



RFC 1918 defines blocks of internet IP addresses that will never be officially assigned to any entity, and will not be routed through the Internet. This means that any site can use these reserved, non-routable networks in their intranet:

- the class A network 10.0.0.0/8 (10.0.0.0 to 10.255.255.255)
- the class B networks 172.16.0.0/12 (172.16.0.0 to 172.31.255.255)
- the class C networks 192.168.0.0/16 (192.168.0.0 to 192.168.255.255)

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 as the internal network and assign virtual server addresses out of this network for the servers. Figure 59 shows an example of a reserved network configuration, where Equalizer uses public IP addresses on one subnet for clusters, and reserved IP addresses for servers.

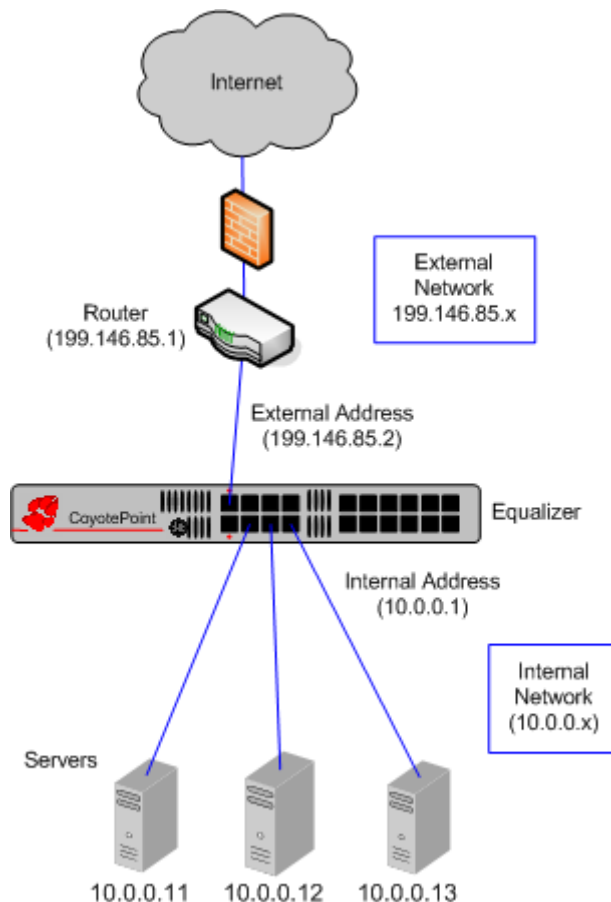


Figure 59 Servers using reserved IP addresses

Reserved IP Addresses and Outbound NAT

The only issue with using reserved IP addresses on servers behind Equalizer arises if the servers need to *originate* connections with hosts on the Internet for any reason (such as performing DNS resolution or sending e-mail), and Equalizer has clusters and servers configured on different VLANs. If this is the case, Equalizer must be configured to perform *outbound NAT*.

When you enable outbound NAT, Equalizer translates the source IP in packets *originating* from the servers on the reserved network so that external hosts will not see packets originating from non-routable addresses; Equalizer's can be configured to use one of its management IPs, a cluster IP, or a failover alias as the source IP.

If Equalizer is configured to use the same VLAN/subnet for clusters and servers, outbound NAT should be disabled. In this case, NAT should be configured on the gateway for the VLAN/subnet so that the reserved IP addresses on Equalizer are translated by the gateway.

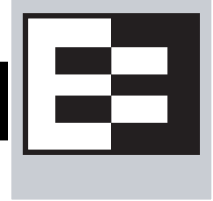
Note – Enabling outbound NAT requires additional processing for each server response. If your servers do not need to initiate connections with hosts through Equalizer, disabling outbound NAT will improve performance.

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** (or system name) entry near the top of the object tree. In the right frame, select the **Clusters > Networking** tab.
3. Enable the **enable outbound NAT** check box.
4. Click the **commit** button.
5. The Outbound NAT address used for each server can be configured as described in the section “Configuring Outbound NAT” on page 156.

Outbound NAT and Failover

If you have two Equalizers deployed in a dual network failover configuration and the **dont transfer** failover option is enabled, be sure to use the same outbound NAT setting on *both* Equalizers. For more information on failover, see “Setting Up a Failover Configuration” on page 95.



Regular Expressions in Match Rules and Responders	291
Terms	291
Learning About Atoms	292
Creating a Bracket Expression	292
Escape Sequences	293
Matching Expressions	293

Regular Expressions in Match Rules and Responders

Equalizer supports IEEE Std 1003.2 (POSIX.2) extended regular expressions in Match Rules and Responders. There are many other variants and extensions of regular expressions, including those found in Perl, Java, and various shell languages; these variants are not supported in Match Rules and Responders.

Regular expressions can be difficult to create and debug, and can use significant system resources to process. We recommend you use regular expressions only when no other method 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.

Escape Sequences

The following escape character sequences match the indicated characters:

<code>\\</code>	matches a single backslash (\)
<code>\b</code>	matches the beginning of a word (e.g.: <code>\bex</code> matches 'example' but not 'text')
<code>\n, \r, \t, \v</code>	match whitespace characters
<code>\', \"</code>	match single and double quotes

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.



Using Certificates in HTTPS Clusters

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. Most SSL certificate vendors offer basic SSL overviews on their websites.

Using Certificates in HTTPS Clusters	296
About Server Certificates	296
About Client Certificates	297
General Certificate Guidelines	297
Software vs. Hardware Encryption/Decryption	298
Using Certificates in a Failover Configuration	298
Enabling HTTPS with a Server Certificate	298
Enabling HTTPS with Server and Client Certificates	299
Generating a CSR and Getting It Signed by a CA	300
Generating a CSR using OpenSSL	300
Generating a Self-Signed Certificate	301
Preparing a Signed CA Certificate for Installation	301
Installing Certificates for an HTTPS Cluster	302
Using IIS with Equalizer	304
Generating a CSR and Installing a Certificate on Windows Using IIS	304
Converting a Certificate from PEM to PKCS12 Format	305
Private Keys for Cluster Certificates	306
Private Key Storage	306
Clearing Secure Key Storage on Xcel I	306
Private Key Length	307
Configuring Cipher Suites	307
Default Cipher Suites	307
Updating the Cipher Suites Field	308
No Xcel (Software) and Xcel II Cipher Suites	308
Xcel I Cipher Suites	309
HTTPS Performance and Xcel SSL Acceleration	309
Choosing the Cipher for an HTTPS Client Connection	309

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.

Each Layer 7 HTTPS cluster requires a *server* certificate; *client* certificates are 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.

About Server Certificates

In a 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, and sending the server certificate to the client. If you have many systems servicing requests for the same website, you need to install certificates on each server.

With Equalizer, you do not need to install a server 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, some Equalizer models are equipped with Xcel SSL Hardware Acceleration. With Xcel, all SSL processing is done by dedicated Xcel hardware, enhancing overall HTTPS throughput. For more information on Xcel, please visit the Coyote Point website (www.coyotepoint.com).

Note that HTTPS and certificates can also 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. Using a Layer 4 cluster is the preferred method for passing HTTPS traffic through Equalizer when you do not need to take advantage of features that are specific to Layer 7, such as cookie persistence, match rules, etc.

About Client Certificates

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.

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.

Client certificates can be used in two ways with Equalizer:

1. **Install the entire client certificate chain on Equalizer.** This requires that every client passes the exact same certificate to Equalizer for validation.
2. **Install an intermediate CA certificate as the client certificate on Equalizer.** This allows unique certificates to be used on clients and a single client certificate to be uploaded to Equalizer. Following this method requires some certificate processing on the servers behind Equalizer in order to prevent access by clients with revoked certificates. *This method, therefore, should be used only under the following conditions:*
 - a. If the site is able to use an intermediate CA, or multiple CAs, which signs **all and only** certificates authorized for use with the cluster,

AND
 - b. If the application running on the servers behind Equalizer is able to perform Certificate Revocation List (CRL) processing by matching the CSN (certificate serial number) to the intermediate CA's CRL, and does so for **all** requests,

THEN
 - c. The Equalizer can safely support the use of individual client certificates for different clients, by appropriately setting the **verify depth** option for the HTTPS cluster and uploading the intermediate CA's certificate to the cluster as the client certificate. If client certificates use different CAs, multiple intermediate CAs can be uploaded to Equalizer in a single file.

This method ensures that only certificates that pass the CRL check on the server can be used to access the cluster. Note that this method also assumes that validating the intermediate certificate only in (b) above is sufficiently secure for the site.

General Certificate Guidelines

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.
- Some older Equalizer models are equipped with an Xcel I Hardware SSL Acceleration, which requires a **private key length** of 1024 bits. This key length restriction does not apply to the newer generation Xcel II hardware, though a private key length of 1024 is recommended for best performance. (Note that all Equalizer GX hardware models that have Xcel are equipped with Xcel II.)
- When uploading certificates to Equalizer in **PEM** format, the certificates and private key must be contained in a single plain-text file, in the following order:
 - certificate
 - private key
 - chained root (intermediate) certificates (if any)

Software vs. Hardware Encryption/Decryption

Without Xcel hardware SSL acceleration, all Layer 7 HTTPS encryption and decryption is performed by software, using Equalizer's CPU and memory. With Xcel, all SSL operations for Layer 7 HTTPS clusters are performed on dedicated hardware, thus offloading both the servers behind Equalizer and Equalizer itself -- freeing more resources for traffic and application management.

In terms of configuration, both software and hardware SSL operations require a list of cipher suites (encryption algorithms) to be used to encrypt and decrypt HTTPS traffic. The supported cipher suites for each SSL processing mode (software, Xcel I, Xcel II) are described in the section "Configuring Cipher Suites" on page 307.

Also see the section "Private Keys for Cluster Certificates" on page 306 for a discussion of how Equalizer stores the private keys for your cluster certificates, and keeping private keys secure on Equalizer.

Using Certificates in a Failover Configuration

In failover configurations, if client and server certificates are *not* part of the configuration settings that are transferred between the failover peers, you must install the server certificates (and the client certificates, if used) on *both* of the failover peers.

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 300.
- 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 301.

2. Create the HTTPS cluster.

When creating an HTTPS cluster, the default flags and parameters are acceptable for most server certificate configurations.

For more information on SSL parameters, see the section "Layer 7 Security > SSL Tab (HTTPS only)" on page 130.

3. Install the Server Certificate on Equalizer.

Use the Equalizer Administration Interface to install the server certificate. See the section "Installing Certificates for an HTTPS Cluster" on page 302.

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 298) 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 300.
- b. **Create a certificate and sign it yourself.** See the section “Generating a Self-Signed Certificate” on page 301.

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 the **SSL** tab in the right frame.
 - b. Enable the **certify_client** flag; this tells Equalizer to request a client certificate when a client attempts to connect to this cluster.
 - c. 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.
 - d. 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.
 - e. 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.
 - f. Select **commit** to save your changes to the cluster definition.

For more information on SSL parameters, see the section “Layer 7 Security > SSL Tab (HTTPS only)” on page 130.

4. Install the Client Certificate on Equalizer.

Use the Equalizer Administration Interface to install the client certificate. See the section “Installing Certificates for an HTTPS Cluster” on page 302.

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. For several good tutorials on how to get your certificates signed, please see:

<http://sial.org/howto/openssl/>

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 in these examples. See the **openssl(1)** and **req(1)** manual pages for the SSL implementation on your system for more information.

Note – Generating a CSR from the command line on Equalizer is NOT supported. Consult the Certificate Authority that supplies your SSL certificates and use the tools that they recommend.

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*. If you already have a private key, use **-key filename** (instead of **-newkey rsa:1024**) to specify the file containing the private key.

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 301.

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*. If you already have a private key, use **-key filename** instead of **-newkey rsa:1024** to specify the file containing the private key.

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 304.

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 Certificates for an HTTPS Cluster” on page 302.

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 Certificates for an HTTPS Cluster” on page 302.

Installing Certificates 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).

If all of your clients use the same certificate to authenticate to the server, load the entire chain onto Equalizer. If each client uses a unique certificate, you can instead load all the intermediate and root certificates (minus the unique client certificate) onto Equalizer, and any client certificate presented that uses that chain will be accepted.

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 PEM (.pem) or PKCS12 (.pfx) format; see the section “Preparing a Signed CA Certificate for Installation” on page 301.

Caution – The private key for your *server* certificate 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. Note that the private key for a *client* certificate (if used) is not stored on Equalizer, only the client certificate.

To install a certificate for an Equalizer cluster, 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 Administrative Interface using a login that has **add/del** access on the cluster that requires the certificate (see “Logging In” on page 52).
3. In the left frame, click the name of the HTTPS or SSL cluster for which you want to install a certificate and select the **Security > Certificates** tab in the right frame:

select client or cluster certificate

For client verification, upload a single client certificate to authenticate all clients. For server verification, upload a single server certificate for the cluster.

client

cluster

select SSL certificate file

The certificate file must be in PEM (.pem) or PKCS12 (.pfx) format, and must contain the private key and the entire certificate chain.

Figure 60 The cluster Certificates tab

Note: If your Equalizer has **Xcel I** Hardware SSL Acceleration installed, a check box labeled **use secure key storage** will appear at the top of the **select client or cluster certificate** field. Checking this box tells Equalizer to store the private key for the server/cluster certificate in write-only memory on the Xcel hardware so that no one can access it. By default, this check box is disabled and Equalizer stores the private key in its file system, under */var/eq/ssl*. See the section “Private Keys for Cluster Certificates” on page 306, for more information. This option does not appear if your Equalizer is equipped with **Xcel II**, or does not have Xcel; in these cases, private keys are stored in Equalizer’s file system, under */var/eq/ssl*.

4. 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.
5. Enter the full path name of the certificate file (or click **Browse** to select the file).
6. Click **upload** to install the certificate on Equalizer. You’ll be prompted for a password, which is the password (or pass phrase) you provided when you generated the CSR for the certificate (or created the self-signed certificate).

Note: If you select a file that is not in PEM or PKCS12 format (or select no file at all), the following error message is displayed:

```
Certificate must be in PEM or PKCS12 format.
```

Following the error message is the output the SSL tools returned after they were run on the uploaded file. The output may be unreadable and poorly formatted; this is normal, because the file you uploaded was not in the correct format. Click **dismiss** on the error popup and then go back to the previous step to select a file that is in PEM or PKCS12 format.

After the upload is complete, the **Certificates** tab displays the certificate details (serial number, key length, etc.) at the bottom of the tab.

Additional SSL settings, including the cipher suites permitted, appear on the **SSL** tab. See “Layer 7 Security > SSL Tab (HTTPS only)” on page 130 and “Default Cipher Suites” on page 307 for more information.

7. 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 305.

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 301) 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 Certificates for an HTTPS Cluster" on page 302.

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 301. 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.

Private Keys for Cluster Certificates

When you upload a *cluster* certificate to Equalizer, the uploaded file contains:

- the cluster certificate
- zero or more intermediate certificates
- the private key for the cluster certificate (chosen by you when you created the certificate signing request or self-signed certificate)

The private key should be guarded carefully and access to it restricted to those who administer Equalizer.

Equalizer supports both software SSL acceleration and hardware acceleration. Xcel SSL Hardware Acceleration provides hardware-based SSL encryption and decryption. There are two versions of Xcel: Xcel I and Xcel II. Xcel I is an optional module found on older 'si' hardware. Xcel II is included in all E450GX and E650GX model Equalizers. Equalizers without an Xcel card use software-based SSL acceleration only.

Private Key Storage

If you do not have Xcel enabled, or if you have Xcel II, private keys are kept in Equalizer's file system. The number of keys that can be stored is limited only by the available file system space.

Xcel I provides the option to store private keys in dedicated write-only memory. This is called **secure key storage** (SKS). All private keys uploaded to write-only memory can only be accessed by the accelerator hardware. If your Equalizer has Xcel I, a check box labeled **use secure key storage** will appear on an HTTPS cluster's **Certificates** tab (see Figure 60). Checking this box tells Equalizer to store your private key in Xcel I's write-only memory so that no one can access it. Xcel I provides 128 kilobits of memory for private keys. This will hold up to 128 one-kilobit (1024-bit) keys, the only key length supported by Xcel I. (Be sure to use only 1024-bit private keys with Xcel I, regardless of whether SKS is used.)

Caution – With Xcel II and on Equalizer models without Xcel, a cluster certificate's private key is stored on Equalizer in the directory `/var/eq/ssl` and therefore will be accessible to anyone who can log into Equalizer. It is therefore essential that you restrict access to the Equalizer console via the serial line and SSH, since any user logged into the console can copy or remove your private key. **All Equalizer logins should be protected with non-trivial passwords, and logins should be given only to trusted personnel.**

Clearing Secure Key Storage on Xcel I

Over time, it is possible for the SKS memory on the Xcel I hardware 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 (assuming you have not already used all 128kb of space on the Xcel hardware with valid keys). After you clear SKS, you'll need to re-add all the certificates for all the HTTPS clusters whose keys were kept in SKS.

To clear SKS memory on Xcel I:

1. Log into Equalizer as *root* over the serial line, or login via SSH and use the **su** command to switch to the *root* login.
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.

Private Key Length

For Xcel II, and on Equalizer models without Xcel, a key length of 1024 bits or less is recommended. Xcel I supports *only* 1024-bit keys. Under no circumstances do we recommend key lengths shorter than 1024 bits be used with Equalizer.

Xcel II and software SSL acceleration without an Xcel card support larger key lengths, but it is important to understand the performance implications of choosing to use keys larger than 1024 bits.

The improvement in security with 2048-bit keys, though real, comes at a dramatic price in terms of performance. 2048-bit RSA operations are about 8 times as computationally expensive as doing the same operations with 1024-bit keys, so peak transactions per second with a 2048-bit key will be about 1/8 the figure for a 1024-bit key.

If you do choose to use 2048-bit keys, we recommend an Equalizer model with Xcel II, which provides considerably better performance with 2048-bit keys compared to software SSL acceleration.

Industry security standards continue to migrate from 1024-bit to 2048-bit RSA keys, due to concerns about the security of high-value 1024-bit keys over the course of the next decade or more. Nevertheless, 1024-bit keys are still considered to provide adequate security today for the overwhelming majority of applications and continue to be permitted by almost all standards.

Configuring Cipher Suites

The **cipher suite** HTTPS cluster parameter lists the supported encryption algorithms for incoming HTTPS requests. If a client request comes into Equalizer that does not use a cipher in this list, the connection is refused. If this field is blank, then any cipher suite supported by Equalizer's SSL implementation (or by Xcel Hardware SSL Acceleration, when enabled) will be accepted.

To view or set the **cipher suite** field for a cluster, click on the cluster name in the left frame and then select the **Security > SSL** tab in the right frame.

Default Cipher Suites

For an Equalizer with no Xcel SSL Hardware Acceleration installed and for systems with Xcel II enabled, the following default setting for **cipher suite** is used:

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

For an Equalizer with Xcel I enabled, the following default value is used:

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

Updating the Cipher Suites Field

This field can be used to specify a custom cipher suite required by the servers in a cluster. In general, to add a cipher suite, you specify a plus sign (+) and then the name of the suite. To specifically exclude a cipher suite, use an exclamation point (!).

For example, SSLv2 encryption is supported by default. If your servers are required to support medium and high encryption using SSLv3 *only*, you can add “!SSLv2” to **cipher suite**. For example, the following cipher suite string will cause all non-SSLv3 client requests to be refused:

AES128-SHA:DES-CBC3-SHA:RC4-SHA:RC4-MD5:AES256-SHA:!SSLv2:+SSLv3

The **cipher suite** field requires a string in the format described in the OpenSSL cipher suite documentation, at:

<http://www.openssl.org/docs/apps/ciphers.html>

The tables in the following sections list the cipher suites supported by Equalizer. Also see the discussion of the cluster parameter “**cipher suite**” on page 130.

No Xcel (Software) and Xcel II Cipher Suites

The following cipher suites are supported by the base Equalizer software and by the Xcel II (newer generation) SSL Acceleration Hardware:

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. (In earlier releases, the EXP-RC4-MD5 ciphers were included by default in cipher suite for older browsers that only support 40-bit encryption. If some clients for your web services support only 40-bit encryption, then add EXP-RC4-MD5 to the cipher suite list.)	
EXP-RC4-MD5	TLS_RSA_EXPORT_WITH_RC4_40_MD5 SSL_RSA_EXPORT_WITH_RC4_40_MD5 SSL_CK_RC4_128_EXPORT40_WITH_MD5

Xcel I Cipher Suites

The following cipher suites are supported by the older generation Xcel I SSL Acceleration Hardware.

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

HTTPS Performance and Xcel SSL Acceleration

The E650GX and E450GX include the Xcel SSL Accelerator Card. Equalizer models without Xcel (E250GX and E350GX) perform all SSL processing in software using the system CPU. Equalizers with Xcel perform all SSL processing using the dedicated processor on the Xcel card. This allows the system CPU to concentrate on non-SSL traffic. For most applications, Xcel will process several hundred HTTPS transactions per second with no noticeable degradation in performance either for the HTTPS cluster or for Equalizer as a whole.

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 accelerator, not a bulk data encryption device. 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 the Xcel I and Xcel II cards do 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 Xcel I card on older 'si' models also does not support "AES" ciphers.

The default configuration for HTTPS clusters on Equalizers with an Xcel card will not include ciphers that are unsupported by the Xcel card, as described above. If, however, the cluster's **cipher suite** string is modified to include them, it is possible that they may be negotiated with clients. This will not lead to incorrect operation of the system, but encryption for these cipher suites will occur in software instead of taking advantage of the improved performance provided by the Xcel hardware.

Choosing the Cipher for an HTTPS Client Connection

The cipher suite parameter for an HTTPS cluster lists all of the ciphers that can be negotiated between Equalizer and an incoming client attempting to connect to an HTTPS cluster. Similarly, the client application will have its own list of ciphers that it supports. The client and Equalizer need to go through a process of negotiating the cipher that will be used for the client connection -- if they cannot find a match, the connection will fail. The process of negotiating a cipher for a client connection is as follows:

1. During the SSL handshake phase of the connection, the client sends Equalizer a list of the ciphers it supports.
2. Equalizer examines the client cipher list in the order it is specified, chooses the first cipher that matches a cipher specified in the cluster's **cipher suite** parameter, and responds to the client. If none of the ciphers offered by the client are in the **cipher suite** list for the cluster, the SSL handshake fails.

It is therefore vital that you ensure that there is at least one match between the list of ciphers supported by clients connecting to an HTTPS cluster and the **cipher suite** list for the cluster.

Equalizer VLB



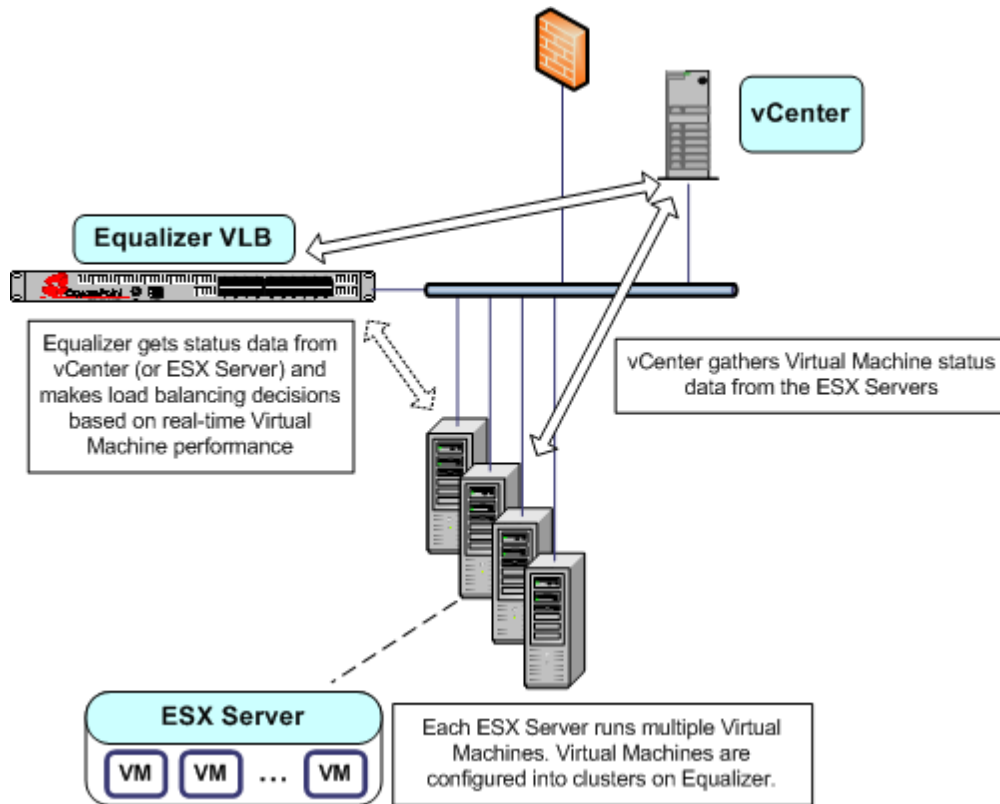
The E250GX supports VLB Basic only.

Equalizer VLB™ is Coyote Point's virtualization enabled load balancing solution for VMware Infrastructure® virtual server configurations. It is available with either a Basic or Advanced license.

Equalizer VLB Basic	312
Using VLB Basic	312
Equalizer VLB Advanced	313
Using VLB Advanced	313
Installation and Licensing	314
Enabling Equalizer VLB	314
Enabling VLB Agents on a Cluster	315
Disabling VLB Agents for a Cluster	316
Disabling Equalizer VLB for all Clusters	317
Associating a Server with a Virtual Machine	317
Smart Control Event Examples Using VLB	318
Configuring Multiple Hot Spares (VLB Only)	318
Rebooting an Unresponsive Virtual Machine (VLB only)	320
VLB Logging	323
VLB Plotting	323
Additional Operational Notes	323

Equalizer VLB Basic

Equalizer VLB Basic uses VMware's management API to retrieve real-time virtual server performance information from a VMware vCenter console that manages virtual machines running on ESX Server (or from a single ESX Server directly). The additional server availability and resource utilization information obtained from VMware allows Coyote Point's Equalizer™ traffic management appliance to more efficiently direct the traffic flowing to VMware virtual machines. The diagram below illustrates how Equalizer VLB works:



Equalizer extracts information from VMware using the VMware API and load balances requests across virtual machines using knowledge of what is going on inside each virtual machine. If there is only one ESX Server in your configuration, Equalizer can also be set up to communicate directly with the ESX Server (instead of vSphere or vCenter), and load balance among the virtual machines defined on that ESX Server only.

Equalizer uses statistics such as the amount of memory in use by a virtual machine, the amount of memory in use by all virtual machines on the physical host, and CPU utilization to automatically distribute incoming cluster requests to the virtual machines added to the cluster. Response to changes in VMware configuration is dynamic. If the virtual server performance in the pool is uneven, Equalizer automatically detects the uneven latency and sends new traffic to the best available virtual machine. If a server is overloaded and reboots, Equalizer simply detects that the server is available again, and automatically resumes sending traffic to it.

Using VLB Basic

Typically, VLB is used in the following manner:

1. Provide VMware login information and enable VLB on the global **VLB** tab.
2. For each cluster that contains (or will contain) servers that are VMware Virtual Machines:
 - a. Enable VLB agents on the cluster **Probe** tab.

- b. On the cluster's **LB Policy** tab, select either the **server agent** policy or the **custom** policy. The **custom** policy lets you adjust the slider controls for the relative influence that the VMware server agent return values will have on load balancing decisions; the **server agent** policy uses preset values.

Once you do the above, VLB automatically attempts to associate all existing (and newly added) server IP addresses with the IP address of a Virtual Machine running on VMware -- if it finds a matching Virtual Machine IP, it associates the VM server with the Equalizer server definition and thereafter checks VMware for detailed VM server status information. *Note that the VM servers must be up and running and have the VMtools software installed for this to work.*

Messages will appear in the Equalizer log (on the global **Status > Event Log** tab) when Equalizer communicates with VMware, and when the state of a VM server changes. Otherwise, VLB works behind the scenes to provide accurate and detailed VM server status information that Equalizer uses to make well-informed load balancing decisions.

Equalizer VLB Advanced

In addition to the Equalizer VLB Basic functionality described above, Coyote Point has developed additional virtualization functionality, provided as Equalizer VLB Advanced:

- The ability to explicitly associate a VMware Virtual Machine (VM) with an Equalizer server definition. When making an explicit association, Equalizer communicates with VMware and lists all the currently defined VMs, so you can pick any one of them for the association (i.e., you are not limited to associating servers automatically by IP, as in VLB Basic). An existing server's **Virtual Machine** tab displays the current association; you can also change the association by selecting a different VM from a list retrieved from VMware.
- With VLB Basic, automatic VM association will work only if the VM server is running and has the VMtools software installed. With VLB Advanced, neither is required in order to make an association.
- The cluster **LB Policy** tab displays additional sliders for VM CPU and VM RAM that allow you to set the influence that these statistics have on load balancing decisions.
- You can create Smart Events that read the status and control the functionality of virtual machines. Additional VMware-specific event triggers and actions allow you to, for example, retrieve the current load on a VM server, or power a VM server on and off.

Using VLB Advanced

Typically, VLB is used in the following manner:

1. Provide VMware login information and enable VLB on the global **VLB** tab.
2. For each cluster that contains (or will contain) servers that are VMware Virtual Machines, enable VLB agents on the cluster **Probe** tab.
3. For each cluster that contains (or will contain) servers that are VMware Virtual Machines:
 - a. Enable VLB agents on the cluster **Probe** tab.
 - b. On the cluster's **LB Policy** tab, select either the **server agent** policy or the **custom** policy. The **custom** policy lets you adjust the slider controls for the relative influence that the VMware server agent return values will have on load balancing decisions; in addition, it also lets you adjust the relative influence that the **VM CPU** and **VM RAM** statistics reported by VMware have on load balancing decisions. The **server agent** policy uses preset values.
4. If desired, open the cluster's **Smart Events** tab and create events that use VMware-specific functions.

Installation and Licensing

Equalizer VLB is installed automatically when you upgrade to Equalizer 8.0.1a, or a later release. The following table summarizes the availability of Equalizer VLB and Smart Control:

Model	VLB Basic	VLB Advanced	Smart Control
E250GX	Included	Not Available	Not Available
E350GX	Included	Licensable	Included
E450GX	Included	Licensable	Included
E650GX	Included	Included	Included

When properly licensed, the bottom of the expanded **Welcome** screen (click **Help > About**) displays the **Virtualization Load Balancing** version, as shown in following example.

```

failover mode      standalone
sequence number   321
Envoy geographic load balancing disabled
SSL acceleration  disabled
hardware GZIP compression disabled
Virtualization Load Balancing VLB Advanced

```

Enabling Equalizer VLB

In order to obtain VMware virtual machine information, Equalizer needs access information for the vCenter console (or ESX server) managing the virtual machines. To enable communication between Equalizer and a vCenter console, do the following:

1. On many VMware products, the VMware SDK is automatically installed. If it is, it will be available at:

```
http://VMwareIP/sdk
```

Where *VMwareIP* is the IP address of the VMware system.

If it is not available, follow the instructions in the VMware SDK & API documentation to install the VMware SDK on the system running vCenter (or on a single ESX Server). The SDK must be installed in order for Equalizer to be able to use VMware Infrastructure API calls and obtain virtual machine status. For instructions, see the VMware documentation at:

```
http://www.vmware.com/support/pubs/
```

2. Install the **VM Tools** software on all Virtual Machines. This is required for automatic VM association in VLB Basic, and for all of the VMware-specific Smart Event functions to work. See the VMware documentation for instructions.
3. Log into Equalizer using an account that has **add/del** permission on global parameters.

- Click **Equalizer** (or the system name) in the left frame, and then open the **Clusters > VLB** tab:

Virtualization Load Balancing parameters

URL:

Username:

Password:

VLB enable/disable

When enabled, allows Equalizer to communicate with virtual machines. When disabled, no connections to Virtual Center will be made.

enable VLB

disable VLB

- Enter the following information:

URL	The URL configured on the system running vCenter (or on an ESX Server) for VMware API connections. By default, this is an https:// URL using the IP address of the vCenter system followed by /sdk , as in: <code>https://192.168.1.50/sdk</code>
Username	The VMware user account that you normally use to log into the vCenter or ESX Server that manages your VMware configuration.
Password	The password for your VMware user account. (Note that this text box is blank when you open the tab, even if a password has been previously saved.)

- Click **enable VLB** to have Equalizer log into VMware automatically and enable virtual machines in clusters. (See “Disabling Equalizer VLB for all Clusters” on page 317 for a description of the **disable VLB** option.)
- Click the **test** button to attempt to login to VMware using the information provided above.
- Click **commit** to save your settings.
- Optionally set the **agent delay**, the number of seconds between probes of VMware vCenter (or ESX Server) for the status of all virtual machines in all clusters (default: 10 seconds). To change the default:
 - Select the **Equalizer > Probes** tab.
 - Specify a new value in the **agent delay** text box.
 - Click **commit** to save the new value.

Enabling VLB Agents on a Cluster

Once you have enabled VLB on Equalizer as shown in the previous section, you can configure clusters with VLB Agents. Doing so enables Equalizer to communicate with the vCenter and get detailed information on all the virtual machines configured in the cluster.

To enable VLB Agents on a cluster:

- Log into Equalizer using an account that has **add/del** permission on the cluster to be modified.
- Do one of the following:

- a. **For VLB Basic:** Click the cluster name in the left frame. In the **Configuration > Required** tab, select **server agent** in the **policy** drop down box. The server agent policy gives preference to the values returned by the VLB agent, and is the recommended setting for VLB clusters.
 - b. **For VLB Advanced:** In addition to choosing the **server agent** policy, as described above, you can also choose the **custom** policy and adjust the relative influence that **VM CPU** and **VM RAM** statistics from VMware have on load balancing decisions.
3. Click **commit** to save the policy change.
 4. Select the **Configuration > Probes** tab:

cluster parameters

probe port	<input type="text" value="0"/>
ACV probe	<input type="text"/>
ACV response	<input type="text"/>
probe delay	<input type="text" value="10.0"/>
server agent port	<input type="text" value="1510"/>
agent probe	<input type="text"/>

agent type

Server agent requires custom agent running on each server. Virtualization Load Balancing (VLB) agent uses Virtual Center configuration to monitor servers.

server agent

VLB

none

5. Select **VLB** in the **agent type** field.
6. Click **commit** to save your settings.

Disabling VLB Agents for a Cluster

Disabling VLB Agents for a cluster means that Equalizer will no longer query the VMware virtual machine manager for status information for virtual machines associated with servers in that cluster. Traffic to the cluster is still load balanced across all servers associated with VMware virtual machines *without* the VLB Agent return value. Smart Rules defined for the cluster that use VLB-specific functions to query VMware will continue to be executed. You can still also add servers and associate virtual machines with them, as long as the VMware login information on the **VLB** tab is correct (see “Enabling Equalizer VLB” on page 314).

1. Log into Equalizer using an account that has **add/del** permission on the cluster to be modified.
2. Click the cluster name in the left frame, then select the **Configuration > Probes** tab in the right frame.
3. Select **none** in the **agent type** field.
4. Click **commit** to save your settings.

Disabling Equalizer VLB for all Clusters

Disabling VLB globally means that Equalizer will no longer query the VMware virtual machine manager for status information for virtual machines associated with servers in any cluster (in other words, VLB Agents are disabled for all clusters). In addition, any Smart Rules that use functions that query VMware will *not* be executed. Cluster traffic is still load balanced across all servers associated with VMware virtual machines *without* the VLB Agent return value. You can still also add servers and associate virtual machines with them, as long as the VMware login information on the **VLB** tab is correct (see “Enabling Equalizer VLB” on page 314).

1. Log into Equalizer using an account that has **add/del** permission on global parameters.
2. Click **Equalizer** in the left frame, and then select the **VLB** tab in the right frame.
3. Click the **disable VLB** button near the bottom of the tab.
4. Click **commit** to save your settings.

Associating a Server with a Virtual Machine

Note – This section applies to VLB Advanced only. VLB Basic automatically associates servers defined on Equalizer with VMware Virtual Machines *by IP address*. That is, when you add a server to a cluster with VLB probing enabled, VLB Basic determines if there is a VMware virtual machine with the same IP address, and if so makes the association automatically. The server **Virtual Machine** tab is *not* present on systems with VLB Basic.

Before you can associate a server with a virtual machine, the VMware login information must be supplied as shown in the section “Enabling Equalizer VLB” on page 314. Associations are established either when a server is added to a VLB cluster, or later by editing an existing server’s configuration:

1. Do *one* of the following:
 - If you are adding a new server to a VLB cluster, enable the Virtual Machine check box on the **Add New Server** dialog, and click **Next** (>). On the following screen, click the **Associate with Virtual Machine** button to query VMware.
 - To associate an existing Equalizer server with a virtual machine, click the server name in the left frame and open the **Virtual Machine** tab in the right frame. Click the **Associate with Virtual Machine** button to query VMware.
2. If the query is successful, a list of available virtual machines is displayed. Choose the virtual machine you want to associate with the Equalizer server definition and click the **Associate** button. (Note that in order for VM selection to work, VLB must be enabled as described in the section “Enabling Equalizer VLB” on page 314.)

You can add both virtual machines and non-virtual machines (physical servers) to a VLB cluster. The non-virtual machines will be load balanced without any VLB server agent value.

Similarly, you can mix virtual and non-virtual machines as servers in a non-VLB cluster. The virtual machines will be load balanced as if they were physical servers, using no VMware data.

Finally, whether you are using VLB as the probe method for the cluster or not, you can use servers which are associated with virtual machines within Smart Control Events, and these servers will be monitored and controlled as specified in the events.

Smart Control Event Examples Using VLB

VLB Advanced provides VLB-specific extensions to the Smart Events feature -- additional functions are provided that let you query a VMware configuration for status information and perform VMware specific operations, such as powering down a virtual server. See the section “Configuring Smart Events” on page 171 for a complete list of supported actions and triggers.

The following examples show you how to use Smart Events in an Equalizer VLB configuration. We assume that you have already configured Equalizer to work with an existing VMware installation, by supplying the appropriate login information on the **Equalizer > Clusters > VLB** tab.

Note – VLB-related Smart Event functions require that the VMTools software is installed on your Virtual Machine servers. This is usually added after a Virtual Machine is created. See the VMware documentation for instructions.

Configuring Multiple Hot Spares (VLB Only)

The **hot spare** server flag is used to designate a single hot spare server in a cluster. In some cases, you may want to designate multiple hot spare servers for a cluster. This can be done using Smart Events.

For example, assume a cluster that has three servers: **sv00**, **sv01**, and **sv02**. As long as it is up, we want **sv00** to actively pass traffic and the other two servers act as successive hot spares. That is, **sv01** and **sv02** will have the **quiesce** option enabled while **sv00** is running. If **sv00** fails, we want **sv01** to take over. If both **sv00** and **sv01** are unavailable, we want **sv02** to take over.

To accomplish this, we’ll create three Smart Events to be evaluated whenever Equalizer processes cluster events:

Event 1: If the first server is running, enable the quiesce option on the other two servers; otherwise, do nothing.

Event 2: If the first server is not running and the second server is, enable the quiesce option on the first and third servers, and disable it on the second server; otherwise, do nothing.

Event 3: If both the first server and the second server are not running, enable the quiesce option on the first and second servers, and unquiesce the third server; otherwise, do nothing.

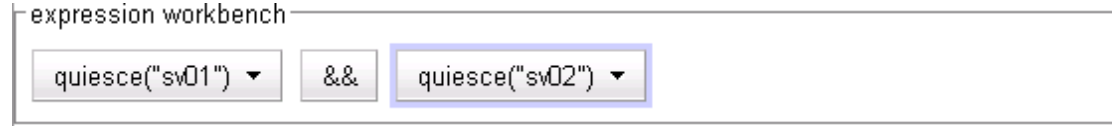
To create Event 1:

1. Right-click on the cluster name in the left frame and select **Add Event** from the menu:
2. Type in an event name, such as **activate-sv00**, or accept the default. Click the next icon (>) at top to open the **Event Trigger** editor.
3. In the **functions** field, click **running**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **running**. Select **sv00** and click **accept**. The **expression workbench** should now look like this:



4. Click the next icon (>) at top to open the **Event Action** editor.
5. In the **functions** field, click **quiesce**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **quiesce**. Select **sv01** and click **accept**.
6. In the **operators** field, click **&&**.

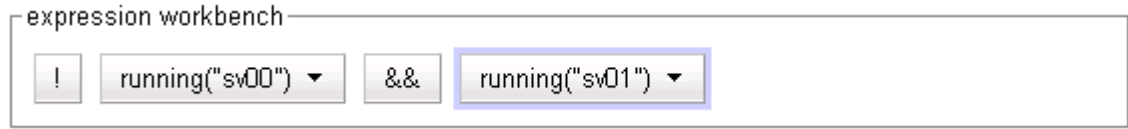
7. In the **functions** field, click **quiesce**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **quiesce**. Select **sv02** and click **accept**. The **expression workbench** should now look like this:



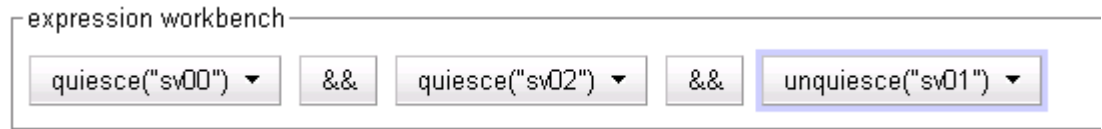
8. Click the next icon (>) at top, and then **commit** to create the event. The Configuration tabs for the event open in the right frame.

To create Event 2:

1. Right-click on the cluster name in the left frame and select **Add Event** from the menu:
2. Type in an event name, such as **activate-sv01**, or accept the default. Click the next icon (>) at top to open the **Event Trigger** editor.
3. In the **operators** field, click on the NOT operator (!).
4. In the **functions** field, click **running**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **running**. Select **sv00** and click **accept**.
5. In the **operators** field, click **&&**.
6. In the **functions** field, click **running**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **running**. Select **sv01** and click **accept**. The **expression workbench** should now look like this:



7. Click the next icon (>) at top to open the **Event Action** editor.
8. In the **functions** field, click **quiesce**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **quiesce**. Select **sv00** and click **accept**.
9. In the **operators** field, click **&&**.
10. In the **functions** field, click **quiesce**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **quiesce**. Select **sv02** and click **accept**.
11. In the **operators** field, click **&&**.
12. In the **functions** field, click **unquiesce**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **unquiesce**. Select **sv01** and click **accept**. The **expression workbench** should now look like this:



13. Click the next icon (>) at top, and then **commit** to create the event. The Configuration tabs for the event open in the right frame.

To create Event 3:

1. Right-click on the cluster name in the left frame and select **Add Event** from the menu.
2. Type in an event name, such as **activate-sv02**, or accept the default. Click the next icon (>) at top to open the **Event Trigger** editor.
3. In the **operators** field, click on the NOT operator (!).

4. In the **functions** field, click **running**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **running**. Select **sv00** and click **accept**.
5. In the **operators** field, click **&&**.
6. In the **operators** field, click on the NOT operator (**!**).
7. In the **functions** field, click **running**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **running**. Select **sv01** and click **accept**. The **expression workbench** should now look like this:



8. Click the next icon (**>**) at top to open the **Event Action** editor.
9. In the **functions** field, click **quiesce**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **quiesce**. Select **sv00** and click **accept**.
10. In the **operators** field, click **&&**.
11. In the **functions** field, click **quiesce**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **quiesce**. Select **sv01** and click **accept**.
12. In the **operators** field, click **&&**.
13. In the **functions** field, click **unquiesce**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **unquiesce**. Select **sv02** and click **accept**.
14. In the **operators** field, click **&&**.
15. In the **functions** field, click **log**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **log** and type "Last hot spare activated! Check servers!". Click **accept**. The **expression workbench** should now look like this:



16. Click the next icon (**>**) at top, and then **commit** to create the event. The Configuration tabs for the event open in the right frame.

Note that the above example is very basic and does not handle all possible event combinations. In particular:

- No action is taken when one of the hot spares becomes unavailable while the first server is passing traffic. These will be indicated as down in the Cluster and Server tabs.
- No action is taken if the third server is not available. This could be handled by a separate event.
- No action is taken if all three servers are not available. This is best handled by a Responder combined with a Match Rule.

Rebooting an Unresponsive Virtual Machine (VLB only)

In this example, we want to reboot a particular virtual machine running under VMware when the machine has been unresponsive for 15 minutes.

To do this, we create two events to power up and power down the machine, and a third event that acts as a timer for the power down and power up events.

Event 1: If the server is not running, power the server down; otherwise, do nothing.

Event 2: If the server is not powered on, power the server on; otherwise, do nothing.

Event 3: If the server is running, block Event 1 and Event 2 for 15 minutes; otherwise, do nothing.

While the server is running, Event 3 continually blocks the other two events from being evaluated. If the server goes down, Event 1 stops blocking after about 900 seconds. The first time that Events 1 and 2 are evaluated, Event 1 is triggered while Event 2 does nothing (Event 1 has only just triggered, so VMware is not reporting the server as down yet). At the next event cycle, Event 2 determines that the server is not powered on and will power it back on, completing the reboot.

To create Event 1:

1. Right-click on the cluster name in the left frame and select **Add Event** from the menu:
2. Type in an event name, such as **poweroff-sv00**, or accept the default. Click the next icon (>) at top to open the **Event Trigger** editor.
3. In the **operators** field, click on the NOT operator (!).
4. In the **functions** field, click **running**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **running**. Select **sv00** and click **accept**. The **expression workbench** should now look like this:



5. Click the next icon (>) at top to open the **Event Action** editor.
6. In the **functions** field, click **power_off**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **power_off**. Select **sv00** and click **accept**. The **expression workbench** should now look like this:



7. Click the next icon (>) at top, and then **commit** to create the event. The Configuration tabs for the event open in the right frame.

To create Event 2:

1. Right-click on the cluster name in the left frame and select **Add Event** from the menu:
2. Type in an event name, such as **poweron-sv00**, or accept the default. Click the next icon (>) at top to open the **Event Trigger** editor.
3. In the **operators** field, click on the NOT operator (!).
4. In the **functions** field, click **powered**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **powered**. Select **sv00** and click **accept**. The **expression workbench** should now look like this:



5. Click the next icon (>) at top to open the **Event Action** editor.

- In the **functions** field, click **power_on**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **power_on**. Select **sv00** and click **accept**. The **expression workbench** should now look like this:



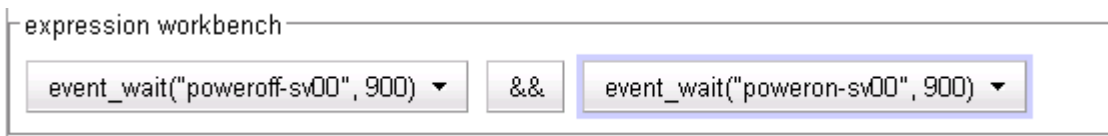
- Click the next icon (>) at top, and then **commit** to create the event. The Configuration tabs for the event open in the right frame.

To create Event 3:

- Create the timer event. Right-click on the cluster name in the left frame and select **Add Event** from the menu.
- Type in an event name, such as **reboot-timer**, or accept the default. Click the next icon (>) at top to open the **Event Trigger** editor.
- In the **functions** field, click **running**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **running**. Select **sv00** and click **accept**. The **expression workbench** should now look like this:



- Click the next icon (>) at top to open the **Event Action** editor.
- In the **functions** field, click **event_wait**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **event_wait**. Select **poweroff-sv00**, and then type "900" for the **wait seconds**. Click **accept**.
- In the **operators** field, click **&&**.
- In the **functions** field, click **event_wait**. In the **expression workbench** field at bottom, click on the drop-down arrow next to **event_wait**. Select **poweron-sv00**, and then type "900" for the **wait seconds**. Click **accept**. The **expression workbench** should now look like this:



- Click the next icon (>) at top, and then **commit** to create the event. The Configuration tabs for the event open in the right frame.

VLB Logging

Equalizer VLB writes a number of messages to the equalizer log (**Equalizer > Status > Event Log**). These messages are described below (timestamps normally displayed at the beginning of each line have been omitted):

```
Logged into Virtual Machine manager successfully
Failed to connect to Virtual Machine manager
Failed to log into Virtual Machine manager
```

The messages above indicate that Equalizer attempted to log into the vCenter or ESX Server IP configured on the **VLB** tab. The status of the first login attempt after a reboot is recorded in the log; subsequent attempts are only logged if the login status changed since the last login. For example, the first successful login attempt is logged; subsequent successful attempts are not recorded. Likewise, the first failure is recorded; no further messages are logged during subsequent attempts until a login attempt succeeds.

```
VLB: probe: Server IP_address VLB state changed from old_value to new_value
```

A message in the above format indicates that the VLB agent return value for the virtual machine at *IP_address* has changed since the last probe of the vCenter (or ESX Server). Both the previous return value (*old_value*) and the latest return value (*new_value*) are logged in the message.

For example, the following series of messages was logged when a spike of CPU activity reduced availability for one virtual machine (server) in a VLB cluster:

```
VLB: probe: Server 192.168.1.51 VLB state changed from 0 to 100
VLB: probe: Server 192.168.1.51 VLB state changed from 100 to 20
VLB: probe: Server 192.168.1.51 VLB state changed from 20 to 0
VLB: probe: Server 192.168.1.51 VLB state changed from 0 to 1
VLB: probe: Server 192.168.1.51 VLB state changed from 1 to 100
```

As the messages indicate, Equalizer continually adjusts to changing conditions on the server. Without VLB agents, Equalizer would not have known about the CPU utilization spike since the ‘ping time’ of the server IP did not change during this period.

VLB Plotting

The VLB agent return values can be plotted for any virtual machine in a VLB cluster.

1. Click on the server name in the left frame object tree. Select the **Reporting > Plots** tab in the right frame.
2. In the **display** multi-pick box, select **Server Agent**. Select other options as desired (click **Help > Context Help** for descriptions of each setting).
3. Select **plot** to display the graph.

Additional Operational Notes

1. **Failover:** All Equalizer VLB configuration settings are stored in the Equalizer configuration file, and so are transferred over to the failover peer when the configurations are synchronized.

We recommend that both failover peers run Equalizer VLB. If Equalizer VLB is used in a failover configuration with an Equalizer that is not running Equalizer VLB, then the **dont transfer** flag must be enabled on both peers. To view or set this option, select the object at the top of the left frame and then open the **Parameters** tab.

2. **Envoy:** Equalizer VLB operation is transparent to Envoy. In other words, you can use a VLB cluster in a GeoCluster configuration just like any other cluster.



Equalizer Doesn't Boot for First Time	325
Clients Time Out Trying to Contact a Virtual Cluster	326
Backup Equalizer Continues to Boot	326
Can't View Equalizer Administration Pages	326
Equalizer Administration Interface Unresponsive	327
Equalizer Administration Page Takes a Long Time to Display	327
Equalizer Doesn't Respond to Pings to the Admin Address	327
Browser Hangs When Trying to Connect Via FTP to an FTP Cluster	327
Return Packets from the Server Aren't Routing Correctly	328
Web Server Cannot Tell Whether Incoming Requests Originate Externally or Internally	328
Why aren't my clusters working if the server status is "up"?	328
Context Help Does Not Appear	328
Restoring IP Access to the Administrative Interface	328
Restoring Login Access to the Administrative Interface	329
Log Contains 'interrupted system call' Messages	329
Log Contains SSL Errors with "wrong version number"	330
GUI Always Reports All Configuration Errors	330
Updating the Configuration File Sequence Number	330

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 for First Time

Terminal or terminal emulator not connected to Equalizer

Check the serial cable connection and the communication settings of the terminal or terminal emulator. Required settings are 9600 bps, 8 data bits, no parity, one stop bit, and VT100 terminal emulation. If you are using terminal emulation software on a Windows or Unix system, make sure the terminal emulation software is connecting to the port to which the serial cable is connected.

Newer Equalizer models also have a USB keyboard connector and VGA display adapter at the back of the unit. You can connect a USB cable and VGA display and use these as a console instead of the serial port.

Clients Time Out 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 `tracert` from the server to the client. Adjust the routing until Equalizer's address shows up in the `tracert` output.



All packets sent from the server back to clients must pass through Equalizer unless the spoof cluster option is disabled.

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 and 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

This behavior is likely to cause problems if you're using reserved internal addresses for the server. Enabling PASV mode FTP translation on the **Networking** tab of the Equalizer Administration Interface substitutes the cluster IP for the server IP in PASV responses. For more information, see "**passive FTP translation**" on page 93.

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 `tracert` 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 `tracert` (Unix) and `tracert` (Windows) commands are 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.

Restoring IP Access to the Administrative Interface

The browser-based Administrative Interface can be accessed via the internal IP, the external IP, or the failover IP, using the `http://` or `https://` protocols. Settings for interface access appear in the configuration file. While the Administration Interface prevents you from disabling all access to the interface, all access can be disabled if the access settings are removed from the configuration file manually or if the configuration file becomes corrupted.

If access to the Administrative Interface is disabled on all available IP addresses and protocols, 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 to enable access via all IP addresses and protocols:


```
parse_config -a -H 1 -i /var/eq/eq.conf -E -I -F -p -s
```

 - `-a`: Update the Apache server configuration.
 - `-H 1`: Restart Apache after one second (seconds must be greater than 0).
 - `-i /var/eq/eq.conf`: Location of Equalizer’s configuration file.
 - `-E -I -F`: Start Apache on the External IP, Internal IP, and Failover IP; respectively.
 - `-p -s`: Enable the HTTP and HTTPS protocols; respectively.
3. Running the above command *does not* update the configuration file, so access may be lost the next time the Apache server is restarted. To restore interface access in the current configuration, see the section “Managing Access to Equalizer” on page 55.

Restoring Login Access to the Administrative Interface

The Administrative Interface prevents you from deleting the login that you are currently using. For example, you cannot log in as **touch** and delete the **touch** login; to delete **touch**, you must log in using a different user name that has the **add/del** permission on users. This also prevents you from deleting all logins via the interface. However, it is possible that all user logins could be deleted by manually editing the configuration file, or in the unlikely event the configuration file becomes corrupted. If this occurs, do the following:

1. Log into Equalizer using the serial line or SSH as *eqadmin* or *root*.
2. Enter:


```
eqadmin
```
3. Select **4 Manage users** and press **Enter**.
4. Select **1 Full Access** to create an Administrator login; select **2 Read Only** to create a read-only login. Press **Enter**.
5. A series of prompts appears at the bottom of the screen. Type in the information described below at each prompt and press **Enter**:

```
Enter username: <Login>
Enter full name: <Description or Name>
Enter password: <password>
Enter password again: <repeat password>
```

6. After you press **Enter** at the final prompt above, the system should respond with:

```
User <Login> created successfully.
```

The **eqadmin** utility then returns to the main menu.

Note that any logins you create via **eqadmin** are automatically added to */var/eq/eq.conf*, and will appear in the Administration Interface’s **Users** table (**Equalizer > Permissions > Users**) when you log in again.

Log Contains ‘interrupted system call’ Messages

These messages are normal and indicate that the Equalizer configuration file was updated; the load balancing daemon stopped what it was doing in order to re-load the configuration. These messages can be safely ignored.

Log Contains SSL Errors with “wrong version number”

If you have one or more HTTPS clusters defined, you may see the following messages in the Equalizer log:

```
ssl_err: 425:error:1408F10B: SSL routines:SSL3_GET_RECORD:wrong version number:s3_pkt.c:360:
ssl_err: fatal error with ip_address
```

These messages indicate that a client has sent an HTTPS request to an HTTPS cluster, but has requested an SSL/TLS version that is not configured on the cluster. These messages are logged by the SSL implementation used by Equalizer and do not necessarily indicate a problem on Equalizer.

For example, if you configure the HTTPS cluster to support SSLv3 ciphers only, then any time a client requests a connection using an SSLv2 cipher, SSL will log these messages. Check the cipher suite for the HTTPS cluster and the configuration of the client to ensure that the desired SSL versions are being used.

GUI Always Reports All Configuration Errors

When a configuration update is made and an error occurs, a popup screen informs the user of the error. This error output contains *all* errors detected in the configuration file, not just the error associated with the current operation.

For example, let's say that you have defined an HTTPS cluster but that you have not added an SSL certificate for the cluster yet. If you then try to change another setting in the configuration and make an error, the GUI will report *both* the error you just made *as well as* the missing SSL certificate for the HTTPS cluster.

Updating the Configuration File Sequence Number

If you are establishing a failover configuration between two Equalizers, you should check the sequence number of the configuration file on both Equalizers by clicking **Help > About** and expanding the **Equalizer System Information** box. The configuration file with the highest sequence number will be transferred to the other system during the first synchronization between the systems. If the configuration that you require has a lower sequence number, follow this procedure to edit the configuration file manually:

1. Log in to the Equalizer with the configuration file you require via **ssh** (as *eqsupport*) or the serial interface (as *root*). If you use ssh, enter the following command to switch to the root login:

```
su root
```

2. Copy the configuration file to a temporary location:

```
cp /var/eq/eq.conf /var/tmp
```

3. Edit the file */var/tmp/eq.conf* using either of the text editors supplied with Equalizer: **ee** or **vi**. For example:

```
ee /var/tmp/eq.conf
```

For instructions on using these standard text editors, see their manual pages on the FreeBSD Hypertext Manual Pages website: www.freebsd.org/cgi/man.cgi.

4. Search for the string “**sequence**” in the file, and increase the number on that line until it is higher than the sequence number on the other Equalizer that you will configure into failover. Do not make any other changes to the file.
5. Save your change and exit the editor.

6. To check the new configuration file for syntax errors, enter the following:

```
parse_config -i /var/tmp/eq.conf
```

The **parse_config** command prints any syntax errors it finds to standard out, and nothing if the file is correct.

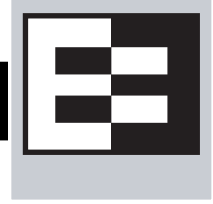
7. If there is a syntax error, repeat Steps 3 through 6 to fix the problem, so that **parse_config** returns no output.

8. Enter the following two commands:

```
mv /var/tmp/eq.conf /var/eq/eq.conf  
shadow /var/eq/eq.conf
```

9. Restart the load balancing daemon to enable the new configuration file:

```
lbd -H
```

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

The Limited Warranty for your Coyote Point Systems product is available online at:

http://www.coyotepoint.com/pdfs/warranty_detail.pdf



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'ACCÈS.

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 in this unit. 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, as recommended by the manufacturer. Dispose of a used battery according to the manufacturer's instructions and in accordance with your local regulations.

Specifications

Power Requirements

The unit's power supply is rated at **100-240 VAC** auto selecting **60/50 Hz @ 4.0A**.

Power Consumption

Use the following power consumption information to determine how many units can be connected to available power circuits without overload. The 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

110V Test Results

Model	110V/60Hz	Watts	PF/VA	Volts	Amps
E650GX					
	Rush-in	112.5	1.000	118.9	0.954
	No Load	112.2	1.000	118.7	0.943
	100% CPU	145.0	1.000	118.2	1.222
E450GX					
	Rush-in	112.5	1.000	118.9	0.954
	No Load	112.2	1.000	118.7	0.943
	100% CPU	145.0	1.000	118.2	1.222
E350GX					
	Rush-in	76.7	0.993	119.9	0.644
	No Load	70.9	0.992	119.7	0.597
	100% CPU	99.5	1.000	119.6	0.831
E250GX					
	Rush-in				
	No Load				
	100% CPU				

220V Test Results

Model	220V/50Hz	Watts	PF/VA	Volts	Amps
E650GX					
	Rush-in	109.1	0.645	224	0.752
	No Load	109.9	0.925	222	0.536
	100% CPU	140.5	0.943	222	0.671
E450GX					
	Rush-in	109.1	0.645	224	0.752
	No Load	109.9	0.925	222	0.536
	100% CPU	140.5	0.943	222	0.671
E350GX					
	Rush-in	74.6	0.877	445	0.378
	No Load	68.5	0.862	225	0.354
	100% CPU	96.3	0.923	224	0.466
E250GX					
	Rush-in				
	No Load				
	100% CPU				

Operating Environment

- **Temperature:** 40 - 105 °F, 5 - 40 °C.
- **Humidity:** 5 - 90%, non-condensing.

Physical Dimensions

Model	Weight	Height	Width	Depth
E250GX	7 lbs. (3.2kg)	1.75 in.	17.25 in.	10.5 in.
E350GX E450GX / E650GX	14 lbs. (6.4kg) 15 lbs. (6.8kg)	1.75 in.	17.25 in.	15.5 in.

Regulatory Certification

Please see the product data sheets on the Coyote Point Website (www.coyotepoint.com) for product certification details.



Glossary

active content verification (ACV)	A method for checking a server for valid content. As part of a TCP probe, Equalizer sends a custom string to the server's probe port and checks the response for a specific string. ACV does not support UDP-based services.
administration address	The IP address assigned to Equalizer on any VLAN. Access to Equalizer can be configured for each VLAN.
administration interface	The browser-based interface for setting up and managing 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 <i>server agent</i> .
aggregation	See <i>link aggregation</i> and <i>sticky network aggregation</i> .
alias	A nickname that replaces a long name or one that is difficult to remember or spell. See <i>IP alias</i> .
aliased IP address	A nickname for an IP address. See <i>IP alias</i> .
algorithm	Instructions, procedures, or formulas used to solve a problem.
application layer	Layer 7 of the Open Systems Interconnection (OSI) network model, where communication between endpoints is defined by the application.
atom	The smallest part of a regular expression in Equalizer. See <i>branch</i> , <i>piece</i> , and <i>regular expression</i> .
authoritative name server	A name server that maintains the master records for a particular domain. See <i>name server</i> .
back-end server	A physical server that is part of a virtual cluster on Equalizer.
backup Equalizer	The backup unit in a failover pair of Equalizers. The backup unit constantly monitors the health of the active (primary) unit, and replaces the primary unit in the event that the primary becomes unavailable. See <i>hot backup</i> and <i>primary Equalizer</i> .
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 <i>atom</i> , <i>piece</i> , and <i>regular expression</i> .
broadcast domain	
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 <i>Computer Dictionary</i> , 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.
connection	A connection is a Layer 4 transmission path established between two endpoints. Clients open connections to Equalizer cluster IPs, and Equalizer opens connections to the servers behind it. The notion of a connection is supplied by the underlying protocol. There are <i>connection-oriented</i> protocols, like TCP and <i>connectionless</i> protocols, such as UDP.
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 <code>www.coyotepoint.com</code> , <code>com</code> is the domain, where <code>com</code> represents a <i>commercial</i> site. See domain name, IP address, and subdomain. See <i>also</i> 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 <code>www.coyotepoint.com</code> , <code>coyotepoint</code> is the domain name. See domain, IP address, and subdomain. See <i>also</i> DNS.
dynamic weight	The weight that Equalizer assigns to a particular server during operation. See server weight, initial weight, and weight.
echo	The transmittal of data that has been sent successfully back to the originating computer. See ping. See <i>also</i> CMP echo request.
endpoint	An IP address-port pair that identifies the start or end of an address; a value that ends a process.

Envoy	Equalizer add-on software that supports geographic clustering and load balancing. See geographic cluster, geographic load balancing, and load balancing. See <i>also</i> 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, <code>east</code> is a hostname and <code>east.coyotepoint.com</code> is its fully qualified domain name. See <i>also</i> 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 <i>also</i> 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 <i>also</i> 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). The SSL/TLS protocol is used in combination with the HTTP protocol to provide secure identification and data encryption.
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.
initial 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 initial weight originally set by the administrator. See dynamic weight, server weight, and weight.
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	<p>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.</p> <ul style="list-style-type: none"> • 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. <p>See Layer 4, Layer 7, and transport layer.</p>
L4	See Layer 4.
L7	See Layer 7.
LAN	See <i>Local Area Network</i> .
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.
link aggregation	
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.
Local Area Network (LAN)	
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.

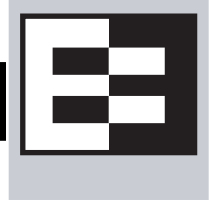
NAT subsystem	The Equalizer subsystem responsible for transferring connections to and from the back-end servers.
netmask	Address mask; a bit mask used to select bits from an Internet address for subnet addressing. The mask is 32 bits long and selects the network portion of the Internet address and one 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 <i>does not</i> 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 <i>also</i> 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 <i>also</i> 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 <i>also</i> 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 <i>also</i> firewall.
quiesce	Quiesce is a server option that, when enabled, tells Equalizer not to send the server any new traffic, while existing connections are allowed to complete. Eventually, the server should not be serving any traffic. Sometimes called 'server draining'.
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 initial 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, <i>routing</i>) 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	Refers to the TCP protocol's reset command, which instructs a device to end a connection.
Secure Sockets Layer (SSL)	A protocol that enables secure communication between two hosts, using data encryption and authentication.
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 <i>also</i> FTP cluster, geographic cluster, and virtual cluster. See server.
server draining	The process of allowing existing connections to a server to complete while not allowing any new connections, so that the server is eventually not serving any traffic. Usually done in preparation for shutting down, rebooting, or upgrading a server. On Equalizer, server draining is enabled using the quiesce server option.
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, initial weight, and weight.
session	A logical connection between a server and a client that may span a series of individual client requests and server responses (i.e., transactions). Depending on the application, a session may also span multiple client-server connections as well as transactions. Session data is typically maintained using cookies inserted into client requests and server responses, by Equalizer, servers, or both. Session data may also be maintained on clients and servers. Equalizer uses cookies at Layer 7 and a sticky timer at Layer 4 to provide server persistence; the cookie lifetime or sticky time to set on Equalizer is determined by the application, and should usually match the corresponding cookie or session timeouts set on the real servers in a cluster.
site	An Envoy site is part of an Envoy geocluster. It points to an existing virtual cluster on an Equalizer running Envoy.
spoofing	Using the client's IP address for the source IP address in client requests. This fools (or spoofs) the server into regarding the client as the source of the request. For spoofing to work, the default gateway for the server must be set to Equalizer's internal IP address.
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 <i>also</i> DNS and RADIUS.
sticky connection	A Layer 4 connection in which a particular client remains connected to same server to handle subsequent requests within a set period of time. Sticky connections are managed on Equalizer using sticky records, which record the server-client connection details; sticky records expire according to the configured sticky timer setting.

sticky network aggregation	<p>Basically, this is server affinity determined by a network mask at Layer 4. If the following conditions are all true:</p> <ul style="list-style-type: none"> • an incoming request to a Layer 4 cluster has a source IP that matches the sticky network mask set for the cluster • the destination port on the server is not responding • the same server IP with another port is defined in another cluster <p>Then Equalizer will attempt to forward the request to the same server on the other port.</p>
sticky timer	A countdown timer used to manage sticky connections to a Layer 4 cluster. When this timer expires (i.e., there is no activity between the server and client for the duration of the timer setting), Equalizer removes the sticky record for the connection.
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 <code>www.coyotepoint.com</code> , <code>www</code> is the subdomain. See domain, domain name, and IP address. See <i>also</i> DNS.
subnet	Part of a network that has the same address as the network plus a unique subnet mask.
switch	A Layer 2 device that connects network segments into one broadcast domain.
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. TCP providesSee ISO/OSI model, Layer 4, packet, transport layer.
TCP/IP	Transmission Control Protocol/Internet Protocol; the rules for transmitting data over networks and the Internet.
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.
transaction	A transaction is a Layer 7 interaction between a client and a server over a network protocol that defines the format of the interaction. An HTTP transaction, for example, is a single client request and associated server response. HTTP is thus a <i>transaction-oriented protocol</i> . This is in contrast to Layer 4 TCP, which is a <i>connection-oriented protocol</i> and does not provide the notion of a transaction to applications running over it.
Transmission Control Protocol (TCP)	See TCP.
Transmission Control Protocol/Internet Protocol (TCP/IP)	See TCP/IP.
transport layer	See Layer 4. See <i>also</i> 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.
VLAN	See <i>Virtual Local Area Network</i> .
VLAN ID	
VLAN tagging	
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 Local Area Network (VLAN)	
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 <i>also</i> 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 initial 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.



- ! 227
- && 227
- || 227
- A**
- abort server 94
- active
 - connections 137
- Active Connections cluster value 206
- Active Connections server value 207
- active connections weight 127, 135
- Active Content Verification 21
- Active Content Verification. See ACV.
- actual value, server static weight 157
- ACV 21, 144, 339
 - enabling 145
- ACV probe string 22
- ACV Probe String field 145
- ACV Response string 22
- ACV Response String field 146
- adaptive load balancing 137, 138, 265
- adding
 - geographic cluster 262
 - match rule to virtual cluster 231
 - server to cluster 152
 - server to virtual cluster 152
 - site to geographic cluster 266
 - virtual cluster 122
- address
 - administration 339
 - aliased IP 339
 - cluster 340
 - external 341
 - internal 342
 - IP 42, 43
 - server 345
 - translation 339
 - virtual server 348
- adjusting
 - server's static weight 157
- administration
 - address 339
 - interface 339
 - interface, changing password 44
- agent 339
 - Equalizer 255
 - retries 204
 - server 138, 345
 - site 204
- agent delay 91
- Agent Misses status 204
- Agent Retries status 204
- agent site parameter 268
- agent weight 127, 135
- agent-to-client triangulation probe 204
- aggregation 339
 - sticky network 92
- aggressive load balancing 138
- ALB algorithm 158
- algorithm
 - definition 339
- algorithms
 - load balancing 26
- alias 339
 - failover 97
 - failover gateway 37
 - server 194
- alias, failover 36
- aliased IP address 339
- all 228
- allow extended chars 94
- and hot spares 170
- any() 227
- application layer. See Layer 7 (L7).
- atom 291, 339
- authoritative name server 26, 27, 48, 339
 - configuring 258
- automatic responses 162
- auto-sensing power supply 40
- B**
- back-end server 339
- backup 36
 - default 95
 - Equalizer 37, 339
 - failover 95
 - hot 36, 48, 341
 - mode 37

Index

- server 155
 - unit 36
- backup Equalizer 36
- backup unit 36, 197
- beginning configuration 42
- boot process 42
- bound 291, 339
- BPDU (bridge protocol data unit) 95
- bracket expression 292, 339
- branch 291, 339
- bridge protocol data unit (BPDU) 95
- browser
 - Javascript-enabled 52
- C**
- cache 339
- cache-time-to-live field 263, 264
- card, XCEL 197
- certificate
 - admin interface 56
- certificates 295–306
 - client verification depth 130
 - convert format 305
 - require client 130
 - verify once 130
- certify_client 130
- changing
 - administration password 44
 - configuration 69, 71, 80, 81, 89, 101, 103, 112, 114, 115, 116, 213
 - console password 45
 - server's static weight 157
- character-based interface 42
- checkboxes
 - ICMP Triangulation 265
- checking
 - validity of server 144
- cipher suite 130
- Class A 340
- Class A network 92
- Class B 340
- Class B network 92
- Class C 340
- Class C network 92
- client timeout 92, 129
- cluster 340
 - adding 122
 - adding geographic 262
 - adding server to 152
 - adding site to geographic 266
 - address 340
 - deleting 136
 - deleting geographic 265
 - FTP 341
 - geographic 25, 341
 - geographic load balancing 263
 - Layer 4 (L4) 139, 146, 160
 - Layer 7 (L7) 146, 160
 - NFS server 21
 - Responders 162
 - server 346
 - statistics, plotting 205
 - virtual 348
- cluster performance, optimizing 158
- cluster value
 - Active Connections 206
 - Hit Rate 206
 - Server Agent 206
 - Servers 205
 - Service Time 206
- cluster, virtual 193
- clusters
 - heterogeneous 158
 - setting static weight for homogenous 158
 - setting static weights for mixed 158
- collating element 292
- Commit option 43, 44
- computed load 340
- Computed Load server value 207
 - server value
 - Computed Load 207
- configuration
 - backup 36
 - beginning 42
 - failover 36, 95
 - initial 41
 - network 151
 - restoring saved 113
 - server 82
 - single network 30, 31
 - testing 49, 194
 - two-network 194
 - understanding 29
- configuration utility, Equalizer 42
- Configure Network Interfaces window 42
- configuring
 - authoritative name server to query Envoy 258
 - cluster to use server agents 138
 - cluster's load balancing options 137
 - Equalizer 41
 - geographic cluster load balancing options 263
 - redundancy 69, 71, 80, 81, 89, 101, 103, 112, 114, 115, 116, 213
 - second Equalizer as hot backup 95
 - servers 82
- connect timeout 92, 125, 129, 134
- connection
 - passive FTP 344
 - record 281
 - stale 346
 - sticky 346
- connection record 281
- connection timeout, stale 92
- connections

- FTP data 148
 - maximum 155
 - sticky 23, 92, 139
- connector, RJ-45 network 40
- console
 - changing password 45
 - logging into 42
- Console option 45
- cookie 340
 - lifetime 126
 - stuffing 140
- cookie generation 126
- Cookie Lifetime option 126
- cookie scheme 126
- cookies
 - scheme 126
- cord, power 40
- CTTL field 263, 264
- custom event handling 213
- custom header 128
- custom headers 146
- custom load balancing policy 127, 135
- cycle, diagnostic 37

D

- daemon 340
 - server agent 138
- data connections, FTP 148
- date 109
- date, setting 44
- default
 - backup 95
 - route 82
- default match rule 224
- Default Router field 41
- default site parameter 269
- defining
 - match rule 231
- delay weight 127, 135
- delegating authority to Envoy site 258
- deleting 235
 - cluster 136
 - geographic cluster 265
 - match rule 235
 - server 161
 - site from geographic cluster 270
- device probe message 42
- diagnosing Equalizer installation and configuration problems 325
- diagnostic cycle 37
- diagnostic messages 42
- Direct Server Return 188
 - loopback interface 190
- displaying
 - site information 268
 - system log 198
 - virtual cluster summary 199

- DNS 21, 26, 41, 49, 263, 264, 290, 340
 - zone file 258
- DNS Server field 41
- DNS TTL 340
- domain 26, 340
- domain name 26, 340
 - fully-qualified 26
- domain name server 42, 44
- domain name service 26
- domain name, fully-qualified 263
- down 21, 37
- DSR 188
- dynamic weight 138, 158, 340
 - oscillations 138
 - spread 138
- Dynamic Weight server value 207
- Dynamic Weight Spread option 138

E

- echo 340
- echo request, ICMP 255
- editing
 - match rule 235
- element, collating 292
- emulation, VT100 42
- emulator, terminal 40
- enable outbound NAT 93, 290
- enabling
 - ACV 145
 - inter-cluster stickiness 140
 - persistent sessions 139
 - sticky connections 139
- endpoint 340
- endpoint, server 346
- Envoy 20, 25, 48, 209, 253, 341
 - DNS zone file 258
 - installing 258
 - site 254
- Envoy Geographic Load Balancing parameter 197
- Envoy site, delegating authority to 258
- eqcollect 116
- Equalizer
 - agent 255
 - ALB algorithm 158
 - backup 37, 339
 - configuration utility 42
 - configurations 29
 - entry 290
 - kernel 37
 - primary 344
 - second 36, 48
 - shutting down 46, 116
 - updating software 45, 117
 - upgrading software 45
- Equalizer Administration interface 51, 53, 101, 103, 341
 - login 52
- Equalizer Configuration Menu window 42, 44

Index

- Equalizer Configuration Utility 341
- Equalizer front panel 40
- Equalizer Version parameter 196
- Equalizer VLB 311
- event handling, custom 213
- event interval 90
- events 171
- expression
 - bracket 339
 - regular (RE) 345
- expression editor 179
- expressions
 - bracket 292
- extended characters 94
- external
 - address 341
 - interface 341
 - network 341
 - test machine 193
- F**
- failover 36, 341
 - alias 97
 - backup 95
 - configuration 36, 95
 - primary 95
 - process 37
- failover alias 36, 97
- failover gateway
 - alias 37
- failover peer 95
- false 224, 227
- firewall 48, 341
 - network 260
- firewalled networks, using Envoy with 260
- FQDN 263, 341
- front panel 40
- FTP 341
 - data connections 148
 - passive mode 148
 - passive translation 93
 - services, providing 148
- FTP cluster 341
- FTP connection, passive 344
- FTP PASV 93
- FTP translation 93
- Fully Qualified Domain Name (FQDN) 341
- fully-qualified domain name 26, 263

- G**
- gateway 36, 42, 43, 151, 341
 - default route 82
- Gateway field 42, 43
- GeoCluster
 - defined 254
 - site 254
- GeoCluster value

- Network Latency 209
- Site Summary 204
- geographic
 - cluster 25, 341
 - load balancing 20, 25, 26, 48, 341
 - probe 255, 341
- geographic cluster
 - adding 262
 - adding site to 266
 - deleting 265
 - load balancing options 263
 - removing site from 270
- Geographic Cluster Name field 263
- geographic load balancing 49
- Geographic Query Protocol 49
- global statistics 205

- H**
- header 341
- header insertion 146
- headers
 - custom 146
- Help
 - Save System Info 116
- heterogeneous clusters 158
- history, plotting geographic cluster 209
- Hit Rate cluster value 206
- homogenous clusters, setting static weight for 158
- host 26
- Host field 42, 43
- Hostname field 41
- hot
 - backup 48
 - spare 48
- hot backup 36, 37, 341
- hot spare
 - and maximum connections 159
 - and Responders 170
- HTTP 25, 144, 146, 160, 341
 - protocol 152
 - request 222
- HTTPS 25, 146, 160, 342
 - custom headers 146
 - header insertion 146
 - request 222
- hub 95, 342
- HyperText Transfer Protocol (Secure). See HTTPS.
- HyperText Transfer Protocol. See HTTP.

- I**
- ICMP
 - drop redirects 93
- ICMP ECHO request 210
- ICMP echo request 255, 260, 342
- ICMP echo request packet 49
- ICMP echo response packet 49
- ICMP probe 91

- ICMP triangulation 256, 265, 342
- ICMP Triangulation checkbox 265
- idle timeout 92
- ignore case 93
- initial configuration 41
- initial weight 21
- installing
 - Envoy 258
 - latest Equalizer software 45, 117
- intelligent load balancing 342
- inter-cluster stickiness 140
- interface 342
 - administration 339
 - Equalizer Administration 51, 53, 101, 103, 341
 - external 341
 - network 344
- interface ports 75
- interfaces
 - character-based 42
- Interfaces option 42, 43
- internal
 - address 342
 - network 342
- internal interface parameters 43
- internal-network test machine 194
- Internet Control Message Protocol (ICMP) 342
- Internet Information Services (IIS)
 - certificates 304
- IP 342
- IP address 22, 42, 43, 194, 342
- IP Address field 43
- IP address, aliased 339
- IP spoofing 151
- ISO/IEC 342
- ISO/IEC 11801 standard 340
- ISO/OSI model 343

J

- Javascript-enabled web browser 52

K

- kernel, Equalizer 37

L

- L4. See Layer 4 (L4).
- L7. See Layer 7 (L7).
- latency 25, 343
- layer
 - Secure Sockets 345
- Layer 4 (L4) 161, 343
 - cluster 139, 146, 160
- Layer 4 load balancing 92
- Layer 7 (L7) 20, 24, 160, 343
 - cluster 146, 160
 - load balancing 222
 - rules 222
- Layers 1, 2, 3, 5, and 6 343

- license 86, 333
- licensing 86
- load 343
 - computed 340
- load balancing 223, 343
 - adaptive 137, 138, 265
 - aggressive 138
 - geographic 20, 25, 26, 48, 49, 341
 - geographic cluster 263
 - intelligent 342
 - Layer 4 92
 - Layer 7 (L7) 222
 - methods 137
 - options 137
 - policy 127, 135, 137, 158
 - response 264
 - responsiveness 127, 135
 - round robin 137, 158
 - round trip 265
 - site load 265
 - site weight 265
 - static weight 137, 158
 - WAP gateways 21
- load balancing algorithms 26
- Load Balancing Response option 264
- load distribution, geographic 26
- local name server 27
- logging into
 - Equalizer console 42
- logical AND 227
- logical NOT operator 227
- logical OR 227
- login
 - Equalizer Administration interface 52
- login prompt 42
- logins 56
 - permissions 56
- loopback interface 190

M

- machine
 - external test 193
 - internal-network test 194
 - test 194
- Management Information Base
 - description 218
- managing
 - servers 150
- match body 223, 228
- match expressions 226
- match rule 222, 235
 - adding to virtual cluster 231
 - defining 231
 - editing 235
 - statistics, plotting 208
- match rule, default 224
- Match Rules

Index

- Responders in 168
 - matching expressions 293
 - maximum number of connections 155
 - messages
 - device probe 42
 - diagnostic 42
 - server status 198
 - start-up 198
 - MIB. See Management Information Base.
 - mode
 - backup 37
 - operation 196
 - primary 37
 - model
 - ISO/OSI 343
 - monitoring
 - cluster performance 158
 - multibyte characters 94
 - multiple logins 56
 - MX exchanger 343
 - MX Exchanger field 264
- N**
- name resolution request 26
 - name server 343
 - Name Server field 42, 44
 - name server, authoritative 48, 339
 - NAT 343
 - enabling outbound 93
 - outbound 290
 - server options 156
 - subsystem 22
 - NAT subsystem 344
 - netmask 344
 - sticky 92
 - network
 - address translation 22
 - configuration 151
 - external 341
 - interface 344
 - internal 342
 - latency 25
 - OSI 344
 - reserved 345
 - RJ-45 connector 40
 - sticky aggregation 92
 - troubleshooting techniques 325
 - Network Address Translation. See NAT.
 - Network Configuration window 42
 - network environment, using Equalizer in single 30, 31
 - network firewall 260
 - Network Latency GeoCluster value 209
 - network ports 75
 - Network Time Protocol. See NTP
 - networks
 - Class A 92
 - Class B 92
 - Class C 92
 - NFS server cluster 21
 - none 228
 - NOT operator 227
 - NTP
 - configuration 109
- O**
- once only 140
 - operation modes 196
 - Optimization Threshold 138
 - optimization threshold 138
 - optimizing
 - cluster performance 158
 - optimizing cluster performance 158
 - options
 - load balancing 137
 - oscillations, dynamic weight 138
 - OSI network 344
 - outbound
 - NAT 290
 - outbound NAT
 - server options 156
 - outbound NAT, enabling 93
 - Outlook Web Access (OWA) 146
- P**
- packet 82, 344
 - ARP 37
 - ICMP echo request 49
 - ICMP echo response 49
 - request 345
 - response 345
 - SYN 92
 - TCP/UDP 25
 - panel, front 40
 - parameters
 - internal interface 43
 - passive
 - FTP translation 93
 - passive FTP connection 344
 - passive FTP mode 148
 - passive FTP translation 93
 - password 42, 52
 - administration interface 44
 - console 45
 - Password option 44
 - PASV 93, 148, 344
 - pattern match 344
 - payload 344
 - pedantic agent 91
 - peer 95
 - performance
 - improving 25
 - monitoring 158
 - optimizing 158
 - optimizing cluster 158

- statistics 138
- permissions 56
- persistence 344
- persistent sessions
 - enabling 139
- physical server 194, 344
- piece 291, 344
- ping 49, 194, 255, 260, 344
- Plot GeoCluster History 209
- Plot Site 209
- plotting
 - cluster statistics 205
 - geographic cluster history 209
 - geographic cluster statistics 209
 - global statistics 205
 - match rule statistics 208
 - site statistics 209
- policy 127, 135
 - custom 127, 135
- port 22, 344
 - redirection 152
- port number 344
- port range 132, 133, 152, 153, 154
- PortFast 95
- ports 75
- power cord 40
- power supply, auto-sensing 40
- preferred primary 95
- primary
 - failover 95
 - mode 37
 - unit 36, 95
- primary Equalizer 344
- primary unit 197
- private keys 306
- probe 344
 - agent-to-client triangulation 204
 - device 42
 - geographic 255, 341
 - site 204
- probe delay 90, 125, 134
- probe interval 90
- probe port 125, 134
- probe timeout 90
- protocol 344
 - SSL 25
- protocol stack 345
- protocols
 - HTTP 152
 - UDP-based Geographic Query 49
- providing
 - FTP services on virtual cluster 148
- proxy server 345

Q

- quiesce 345
- quiescing servers 160

R

- RADIUS 21, 345
- receive buffer 92, 128
- redirect responder 162
- redirection 345
- redirection, port 152
- redirects
 - drop 93
- register (see license) 86
- regular expression 228
- regular expression (RE) 345
- regular expressions
 - Responders 164
- relative value, server static weight 157
- relative workload 210
- Remote Authentication Dial-In User Service. See RADIUS.
- removing
 - cluster 136
 - geographic cluster 265
 - Layer 4 server from service 161
 - Layer 7 server from service 160
 - server 161
 - site from geographic cluster 270
- request
 - HTTP 222
 - HTTPS 222
- request max 128
- request packet 345
- reserved network 345
- resolution 345
- resolution request 26
- Resource Down site value 210
- Resource Load site value 210
- Resource Load status 204
- resource port site parameter 270
- Responders 162, 170
 - in Match Rules 168
 - regular expressions in 164
- response
 - settings 137, 138
- response max 128
- response packet 345
- response time, server 137
- responsiveness 127, 135
- restoring
 - saved configuration 113
- retries, agent 204
- Returned as Default status 204
- RJ-45 network connector 40
- round robin 345
- round robin load balancing 137, 158
 - weighted 137
- round trip load balancing 265
- route command 82
- router 41, 345
- routes
 - static 80

Index

- routing table 345
- RST 345
 - forwarding untranslated 93
- RST on server failure 94
- rules
 - Layer 7 (L7) 222
 - match 222
- S**
- save system information 116
- second Equalizer 36, 48
- secure key storage (SKS) 306
- Secure Sockets Layer (SSL) 345
- send buffer 92, 128
- serial
 - terminal 40
- server 345
 - adding 152
 - address 345
 - agent 345
 - alias 194
 - authoritative name 26, 27, 48, 339
 - back-end 339
 - checking validity 144
 - cluster 346
 - configuration 82
 - domain name 42, 44
 - endpoint 346
 - IP address 194
 - local name 27
 - maximum number of connections 155
 - name 343
 - physical 344
 - proxy 345
 - resource availability 139
 - response time 137
 - shutting down 160
 - virtual web 348
 - weight 157, 346
 - weights 158
- server address, virtual 348
- server agent 138, 206
 - daemon 138
 - using 138
 - value 137
- Server Agent cluster value 206
- Server Agent server value
 - server value
 - Server Agent 208
- server agents 21
- server status messages 198
- server timeout 92, 129
- server value
 - Active Connections 207
 - Computed Load 207
 - Dynamic Weight 207
- servers
 - backup 155
 - deleting 161
 - Layer 4 (L4) 161
 - Layer 7 (L7) 160
 - managing 150
 - quiescing 160
- Servers cluster value 205
- Service Time cluster value 206
- Service Time server value
 - server value
 - Service Time 207
- session 346
 - telnet 82
- session cache kbytes 130
- session cache timeout 130
- sessions
 - enabling persistent 139
- setting
 - date and time 44
 - static weights for homogenous clusters 158
 - static weights for mixed clusters 158
 - time zone 44
- settings
 - response 137
- Shutdown option 46
- shutting down
 - server 160
- shutting down Equalizer 46, 116
- Simple Network Management Protocol 216–219
 - community string 217
 - management station 218
 - SNMP Agent 217
 - traps 217
 - version 216
- single network environment, using Equalizer in 30, 31
- single-network
 - configuration 30, 31
- site 25, 346
 - adding to geographic cluster 266
 - defined 254
 - deleting 270
 - displaying information about 268
- Site Chosen site value 209
- site load balancing 265
- site parameter
 - agent ip address 268
 - default site 269
 - resource port 270
 - wewight 269
- Site Returned status 204
- Site summary GeoCluster value 204
- site value
 - Resource Down 210
 - Resource Load 210
 - Site Chosen 209
- site weight
 - load balancing 265

- site-wide failure 25
- SKS 306
- Smart Control Events 171
- Smart Controls 171
- Smart Event
 - expression editor 179
- Smart Events 171
 - and VLB 171, 318
 - event interval 90
- SNMP. See Simple Network Management Protocol.
- software license 333
- software, updating Equalizer 45, 117
- sorry page 162
- Spanning Tree 95
- spoofing 346
 - IP 151
- ssh 194
- SSL Acceleration parameter 197
- SSL protocol 25
- SSL. See Secure Sockets Layer.
- ssl_unclean_shutdown 130
- stack 346
- stack, protocol 345
- stale connection 346
- stale connection timeout 92
- stale timeout 92
- standards
 - ISO/IEC 11801 340
- start-up messages 198
- state 346
- stateless 346
- static routes 80
- static weight 342
 - changing 157
 - load balancing 137, 158
- statistics
 - performance 138
 - plotting 205, 208
 - plotting geographic cluster history 209
 - plotting site 209
- status
 - Agent Misses 204
 - Agent Retries 204
 - Resource Load 204
 - Returned as Default 204
 - Site Returned 204
- sticky
 - connection 346
 - connections 23, 92
 - network aggregation 92
 - time period 148
 - timer 347
- sticky connections, enabling 139
- sticky netmask 92
- sticky time period 139
- strikeout threshold 90
- stuffing cookie 140

- subdomain 347
- subnet 347
- summary
 - virtual cluster 199
- support information 116
- switch 347
- switch management 75
- SYN packet 92
- SYN/ACK 92, 347
- syslog 347
- system date and time 109
- System Event Log 198
- system information 116
- system log, displaying 198

T

- table, routing 345
- TCP 92, 160, 347
- TCP/IP 347
- TCP/UDP
 - packet 25
- Telnet 347
- telnet 82, 193, 194
- telnet session 82
- terminal 42
 - emulator 40
 - serial 40
- test machine 194
- test machine, external 193
- testing configuration 49, 194
- threshold, optimization 138
- time
 - server response 137
 - setting 44
- time and time zone 109
- Time option 44
- time period, sticky 139, 148
- Time Zone option 44
- time zone, setting 44
- timeout, stale connection 92
- timer, sticky 347
- traceroute 82, 194, 347
- tracert 82
- translation, address 339
- Transmission Control Protocol. See TCP.
- Transmission Control Protocol/Internet Protocol. See TCP/IP.
- transport layer. See Layer 4 (L4).
- trcert 194
- triangulation
 - ICMP 265
- triangulation, ICMP 256, 342
- troubleshooting techniques, network 325
- true 223, 227
- truth value 227
- TTL 347
- two-network configuration 194

Index

U

- UDP 49, 92, 144, 160, 348
- UDP-based Geographic Query Protocol 49
- unit
 - primary 95
- Upgrade option 45
- upgrading Equalizer 45
- URL 52
- user logins
 - permissions 56
- User Datagram Protocol. See UDP
- user logins 56
- using
 - ACV 144
 - Envoy with firewalled networks 260
 - Equalizer in single network environment 30, 31
 - second Equalizer as backup 36
 - server agents 138
- UTF characters 94
- utilities
 - Equalizer Configuration 341
- utility
 - Equalizer configuration 42

V

- verify once 130
- viewing
 - a site's graphical history 209
 - Equalizer information 196
 - geographic cluster's graphical history 209
- virtual
 - cluster 348
 - server address 348
 - web server 348
- virtual cluster 20, 193
 - adding 122
 - adding match rule to 231
 - adding server to 152
 - deleting 136
 - FTP services, providing 148
 - geographic 25
- virtual cluster summary 199
- virtualization 311
- VLAN 75
- VLB 311
 - and Smart Events 171
- VMware integration 311
- VT100 emulation 42

W

- WAP gateway 21
- WAP. See Wireless Application Protocol
- warranty 40
- web browser
 - Javascript-enabled 52
- web server, virtual 348
- weight 348

- adjusting server 157
- dynamic 138, 158
- initial 21
- oscillations 138
- server 158, 346
- spread coefficient 138
- static 342
- weight site parameter 269
- Weight Spread Coefficient option 138
- weighted round robin load balancing 137
- window
 - Configure Network Interfaces 42
 - Equalizer Configuration Menu 42, 44
 - Network Configuration 42
- Wireless Application Protocol (WAP) 348
- wireless application protocol (WAP) 21
- workload
 - relative 210
- writing
 - custom agents 274

X

- XCEL card 197
- XCEL SSL accelerator card 306

Z

- zone file 258