WebSphere Edge Components

Load Balancer for IPv4 Administration Guide

Version 8.0

WebSphere Edge Components

Load Balancer for IPv4 Administration Guide

Version 8.0

Note
Before using this information and the product it supports, be sure to read the general information under Appendix E, "Notices," on page 427.

First edition (May 2011)

This edition applies to:
WebSphere Edge Components, Version 8.0

and to all subsequent releases and modifications until otherwise indicated in new editions.

Order publications through your IBM representative or through the IBM branch office serving your locality.

Load Balancer for IPv4 Administration Guide

Tables

Figures

About this book

This book explains how to plan for, install, configure, use, and troubleshoot IBM® WebSphere® Application Server Load Balancer for AIX®, HP-UX, Linux, Solaris, and Windows operating systems. Previously, this product was called Edge Server Network Dispatcher, SecureWay™ Network Dispatcher, eNetwork Dispatcher, and Interactive Network Dispatcher.

Who should read this book

The ***Load Balancer Administration Guide*** is written for experienced network and system administrators who are familiar with their operating systems and with providing Internet services. Prior exposure to Load Balancer is not required.

This book is not intended to support previous releases of Load Balancer.

Reference Information

The Edge Components Information Center Web site links to the current version of this book in HTML and PDF formats.

For the most current updates about Load Balancer, visit the Web site support page and link to the Technote site.

To access these and related Web pages, go to the URLs listed in "Related documents and Web sites" on page xvii.

Accessibility

Accessibility features help a user who has a physical disability, such as restricted mobility or limited vision, to use software products successfully. These are the major accessibility features in Load Balancer:

- You can use screen-reader software and a digital speech synthesizer to hear what is displayed on the screen. You can also use voice recognition software, such as IBM ViaVoice®, to enter data and to navigate the user interface.
- You can operate features by using the keyboard instead of the mouse.
- You can configure and administer Load Balancer features by using standard text editors or command-line interfaces, instead of the provided graphical interfaces. For more information about the accessibility of particular features, refer to the documentation about those features.

How to send your comments

Your feedback is important in helping to provide the most accurate and high-quality information. If you have any comments about this book or any other Edge components document:

- Send your comments by e-mail to wasdoc@us.ibm.com. Be sure to include the name of the book, the part number of the book, the version, and if applicable the specific location of the text you are commenting on (for example, a page number or table number).

Related documents and Web sites

- *Concepts, Planning, and Installation for Edge Components* GC31-6918-00
- *Programming Guide for Edge Components* GC31-6919-00
- *Caching Proxy Administration Guide* GC31-6920-00
- *Load Balancer for IPv4 and IPv6 Administration Guide*
- IBM Web site home: *www.ibm.com/*
- IBM WebSphere Application Server product: *www.ibm.com/software/webservers/appserv/*
- IBM WebSphere Application Server library Web site: *www.ibm.com/software/webservers/appserv/was/library/*
- IBM WebSphere Application Server support Web site: *www.ibm.com/software/webservers/appserv/was/support/*
- IBM WebSphere Application Server Information Center: *www.ibm.com/software/webservers/appserv/infocenter.html*
- IBM WebSphere Application Server Edge Components Information Center: *www.ibm.com/software/webservers/appserv/was/library/*

Part 1. Introduction to Load Balancer

This part provides an overview of Load Balancer for IPv4 and its components, a high-level description of configuration features that are available, a list of hardware and software requirements, and installation instructions. It contains the following chapters:

Chapter 1. Overview of Load Balancer

This chapter gives an overview of Load Balancer and includes the following sections:

For a high-level list of configuration features provided by each of the Load Balancer components, to assist you in planning which features to use for managing your network, see Chapter 3, "Managing your network: Determining which Load Balancer features to use," on page 17.

What is Load Balancer?

Load Balancer is a software solution for distributing incoming client requests across servers. It boosts the performance of servers by directing TCP/IP session requests to different servers within a group of servers; in this way, it balances the requests among all the servers. This load balancing is transparent to users and other applications. Load Balancer is useful for applications such as e-mail servers, World Wide Web servers, distributed parallel database queries, and other TCP/IP applications.

When used with Web servers, Load Balancer can help maximize the potential of your site by providing a powerful, flexible, and scalable solution to peak-demand problems. If visitors to your site can not get through at times of greatest demand, use Load Balancer to automatically find the optimal server to handle incoming requests, thus enhancing your customers' satisfaction and your profitability.

Which components of Load Balancer can I use?

Load Balancer consists of the following five components that you can use separately or together to provide superior load-balancing results:

- You can use the **Dispatcher** component by itself to balance the load on servers within a local area network or wide area network using a number of weights and measurements that are dynamically set by Dispatcher. This component provides load balancing at a level of specific services, such as HTTP, FTP, SSL, NNTP, IMAP, POP3, SMTP, SIP, and Telnet. It does not use a domain name server to map domain names to IP addresses.

  For HTTP protocol, you can also use the Dispatcher's content-based routing feature to load balance based on the content of the client request. The chosen server is the result of matching the URL to a specified rule. Dispatcher's content-based routing (cbr forwarding method) does *not* require Caching Proxy.
- For both HTTP and HTTPS (SSL) protocol, you can use the **Content Based Routing** (CBR) component to load balance based on the content of the client request. A client sends a request to Caching Proxy, and Caching Proxy sends the request to the appropriate server. The chosen server is the result of matching the URL to a specified rule.

Note: The Content Based Routing (CBR) component is not available on platforms that run a 64-bit JVM, except for HP-UX ia64. On HP-UX ia64, the CBR component runs as a 32-bit application. You can use the CBR forwarding method of Load Balancer's Dispatcher component to provide content-based routing without the use of Caching Proxy. See "Dispatcher's content-based routing (cbr forwarding method)" on page 41 for more information.

- You can use the **Site Selector** component to balance the load on servers within a local or wide area network using a DNS round-robin approach or a more advanced user-specified approach. Site Selector works in conjunction with a name server to map DNS names to IP addresses.
- You can use the **Cisco CSS Controller** or the **Nortel Alteon Controller** component to generate server weights that are then sent to the Cisco CSS Switch or the Nortel Alteon Web Switch respectively for optimal server selection, load optimization, and fault tolerance.

For more information on the Dispatcher, CBR, Site Selector, Cisco CSS Controller, and Nortel Alteon Controller components, see "What are the components of Load Balancer?" on page 7.

What are the advantages to using Load Balancer?

The number of users and networks connected to the global Internet is growing exponentially. This growth is causing scalability problems that can limit users' access to popular sites.

Currently, network administrators are using numerous methods to try to maximize access. With some of these methods, you can choose a different server at random if an earlier choice is slow or not responding. This approach is cumbersome, annoying, and inefficient. Another method is standard round-robin, in which the domain name server selects servers in turn to handle requests. This approach is better, but still inefficient because it forwards traffic without any consideration of the server workload. In addition, even if a server fails, requests continue to be sent to it.

The need for a more powerful solution has resulted in Load Balancer. It offers numerous benefits over earlier and competing solutions:

Scalability

As the number of client requests increases, you can add servers dynamically, providing support for tens of millions of requests per day, on tens or even hundreds of servers.

Efficient use of equipment

Load balancing ensures that each group of servers makes optimum use of its hardware by minimizing the hot-spots that frequently occur with a standard round-robin method.

Easy integration

Load Balancer uses standard TCP/IP or UDP/IP protocols. You can add it to your existing network without making any physical changes to the network. It is simple to install and configure.

Low overhead

Using a simple mac level forwarding method, the Dispatcher component looks at the inbound client-to-server flows only. It does not need to see the outbound server-to-client flows. This significantly reduces its impact on the application compared with other approaches and can result in improved network performance.

High availability

The Dispatcher, Cisco CSS Controller, and Nortel Alteon Controller components offer built-in high availability, utilizing a backup machine that remains ready at all times to take over load balancing if the primary server machine fail. When one of the servers fails, requests continue to be serviced by the other server. This process eliminates any server as a single point of failure and makes the site highly available.

For more information, see "How can Load Balancer provide high availability?"

Content-based routing (using the CBR component or Dispatcher component)

In conjunction with Caching Proxy, the CBR component has the ability to proxy HTTP and HTTPS (SSL) requests to specific servers based on the content requested. For example, if a request contains the string "/cgi-bin/" in the directory portion of the URL, and the server name is a local server, CBR can direct the request to the best server in a set of servers specifically allocated to handle cgi requests.

Note: The Content Based Routing (CBR) component is not available on platforms that run a 64-bit JVM, except for HP-UX ia64. On HP-UX ia64, the CBR component runs as a 32-bit application. You can use the CBR forwarding method of Load Balancer's Dispatcher component to provide content-based routing without the use of Caching Proxy. See "Dispatcher's content-based routing (cbr forwarding method)" on page 41 for more information.

The Dispatcher component also provides content-based routing, but it does not require the Caching Proxy to be installed. Because the Dispatcher component's content-based routing is performed in the kernel as packets are received, it can provide *faster* content-based routing than the CBR component. The Dispatcher component performs content-based routing for HTTP (using the "content" type rule) and HTTPS (using SSL session ID affinity).

Note: Only the CBR component can use the content rule for HTTPS (SSL) when load-balancing traffic based upon the content of the HTTP request, which requires decrypting and re-encrypting messages.

How can Load Balancer provide high availability?

Dispatcher

The Dispatcher component offers a built-in high availability feature, eliminating Dispatcher as a single point of failure from your network. This feature involves the use of a second Dispatcher machine that monitors the main, or primary, machine and stands by to take over the task of load balancing should the primary machine fail at any time. The Dispatcher component also offers mutual high availability which allows two machines to be both primary and secondary (backup) for each other. See "Configure high availability" on page 164.

CBR

You can also achieve a level of high availability using the CBR component when using a two-tier configuration with a Dispatcher machine load balancing traffic across multiple servers that have CBR.

Cisco CSS Controller or Nortel Alteon Controller

The controllers have a high availability feature to eliminate the controller as a single point of failure. A controller on one machine may be configured as a primary and a controller on a different machine may be configured as a backup. The backup monitors the primary and stands by to take over the task of providing server weights to the switches should the primary fail. See "High availability" on page 197 for more information.

Chapter 2. Overview of the components of Load Balancer

This chapter gives an overview of Load Balancer components and includes the following sections:

For a high-level list of configuration features that are provided by each of the Load Balancer components, and to assist you in planning which features to use for managing your network, see Chapter 3, "Managing your network: Determining which Load Balancer features to use," on page 17.

What are the components of Load Balancer?

The five components of Load Balancer are: Dispatcher, Content Based Routing (CBR), Site Selector, Cisco CSS Controller, and Nortel Alteon Controller. Load Balancer gives you the flexibility of using the components separately or together depending on your site configuration. This section gives an overview of these components.

Overview of the Dispatcher component

The Dispatcher component balances traffic among your servers through a unique combination of load balancing and management software. Dispatcher can also detect a failed server and forward traffic around it. Dispatcher supports HTTP, FTP, SSL, SMTP, NNTP, IMAP, POP3, Telnet, SIP, and any other TCP or stateless UDP based application.

All client requests sent to the Dispatcher machine are directed to the "best" server according to weights that are set dynamically. You can use the default values for those weights or change the values during the configuration process.

Dispatcher offers three forwarding methods (specified on the port):

- MAC forwarding method (**mac**). With this forwarding method, Dispatcher load balances the incoming request to the server. The server returns the response directly to the client without any involvement of the Dispatcher.
- NAT/NAPT forwarding method (**nat**). Using Dispatcher's Network Address Translation (NAT)/ Network Address Port Translation (NAPT) capability removes the limitation for the backend servers to be located on a locally attached network. When you want to have servers located at remote locations, you can use the nat technique rather than using a Generic Routing Encapsulation (GRE)/Wide Area Network (WAN) technique. With the nat forwarding method, Dispatcher load balances the incoming request to the server. The server returns the response to Dispatcher. The Dispatcher machine then returns the response to the client.

- Content-based routing forwarding method (**cbr**). Without Caching Proxy, the Dispatcher component allows you to perform content-based routing for HTTP (using the "content" type rule) and HTTPS (using SSL session ID affinity). For HTTP and HTTPS traffic, the Dispatcher component can provide *faster* content-based routing than the CBR component. With the cbr forwarding method, Dispatcher load balances the incoming request to the server. The server returns the response to Dispatcher. The Dispatcher machine then returns the response to the client.

The Dispatcher component is the key to stable, efficient management of a large, scalable network of servers. With Dispatcher, you can link many individual servers into what seems to be a single, virtual server. Your site is presented as a single IP address to the world. Dispatcher functions independently of a domain name server; all requests are sent to the IP address of the Dispatcher machine.

Dispatcher brings distinct advantages in balancing traffic load to clustered servers, resulting in stable and efficient management of your site.

Managing local servers with Dispatcher

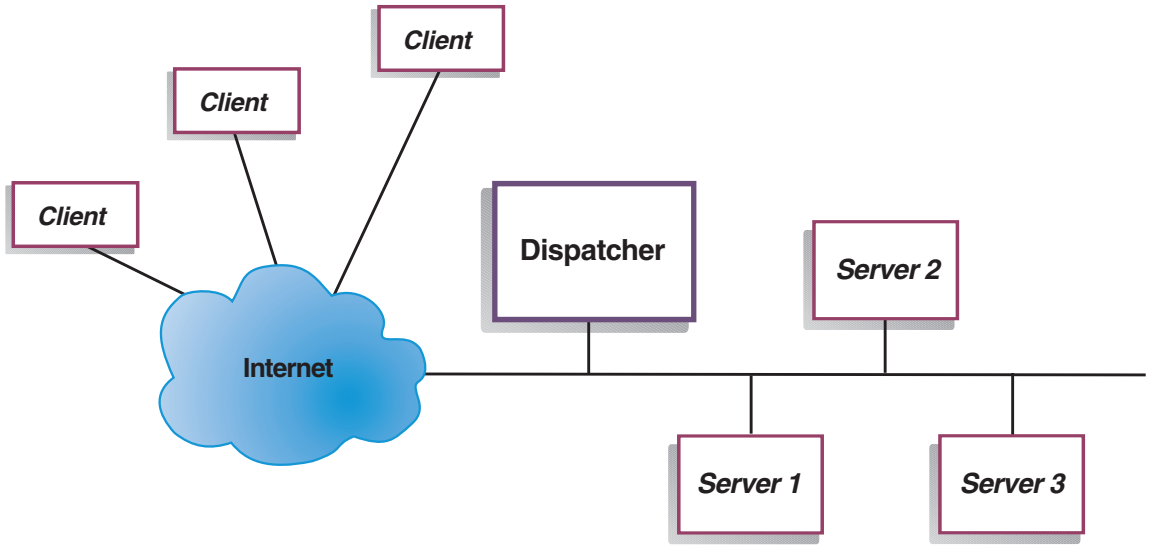


Figure 1. Example of a physical representation of a site using Dispatcher to manage local servers

Figure 1 shows a physical representation of the site using an Ethernet network configuration. The Dispatcher machine can be installed without making any physical changes to the network. After a client request is directed to the optimal server by the Dispatcher, the response is then sent directly from server to client with no involvement by the Dispatcher when using MAC forwarding method.

Managing servers using Dispatcher and Metric Server

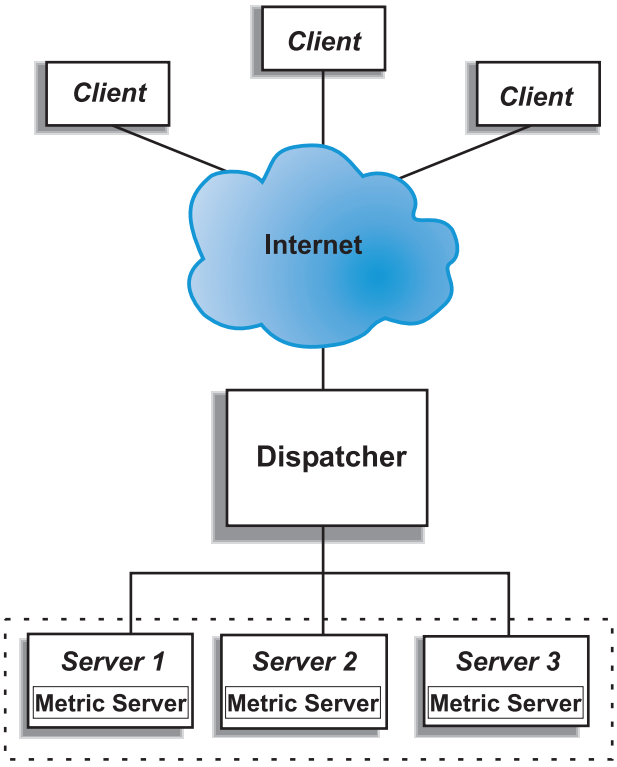


Figure 2. Example of a site using Dispatcher and Metric Server to manage servers

Figure 2 illustrates a site in which all servers are on a local network. The Dispatcher component is used to forward requests, and the Metric Server is used to provide system load information to the Dispatcher machine.

In this example, the Metric Server daemon is installed on each backend server. You can use Metric Server with the Dispatcher component or any of the other Load Balancer components.

Managing local and remote servers with Dispatcher

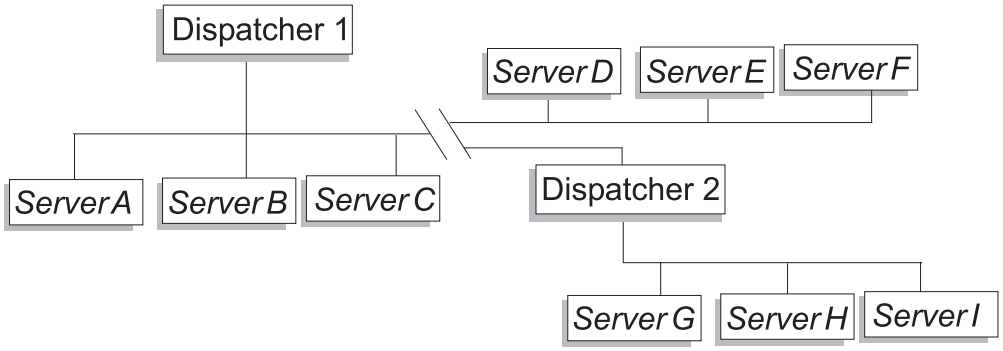


Figure 3. Example of a site using Dispatcher to manage local and remote servers

Wide area support in Dispatcher enables you to use both local and remote servers (servers on different subnets). Figure 3 shows a configuration where one local

Dispatcher (Dispatcher 1) serves as the entry point for all requests. It distributes these requests among its own local servers (ServerA, ServerB, ServerC) and to the remote Dispatcher (Dispatcher 2), which will load balance to its local servers (ServerG, ServerH, ServerI).

When using Dispatcher's NAT forwarding method or using GRE support, wide area support with Dispatcher can also be achieved without using a Dispatcher at the remote site (where ServerD, ServerE, and ServerF are located). See "Dispatcher's NAT/NAPT (nat forwarding method)" on page 39 and "GRE (Generic Routing Encapsulation) support" on page 189 for more information.

Overview of the Content Based Routing (CBR) component

CBR works with Caching Proxy to proxy client requests to specified HTTP or HTTPS (SSL) servers. It allows you to manipulate caching details for faster Web document retrieval with low network bandwidth requirements. CBR and Caching Proxy examines HTTP requests using specified rule types.

Note: The Content Based Routing (CBR) component is not available on platforms that run a 64-bit JVM, except for HP-UX ia64. On HP-UX ia64, the CBR component runs as a 32-bit application. You can use the CBR forwarding method of Load Balancer's Dispatcher component to provide content-based routing without the use of Caching Proxy. See "Dispatcher's content-based routing (cbr forwarding method)" on page 41 for more information.

CBR gives you the ability to specify a set of servers that handle a request based on regular expression matching of the content of the request. Because CBR allows you to specify multiple servers for each type of request, the requests can be load balanced for optimal client response. CBR also detects when one server in a set has failed, and stops routing requests to that server. The load-balancing algorithm used by the CBR component is identical to the proven algorithm used by the Dispatcher component.

When a request is received by Caching Proxy, it is checked against the rules that have been defined in the CBR component. If a match is found, then one of the servers associated with that rule is chosen to handle the request. Caching Proxy then performs its normal processing to proxy the request to the chosen server.

CBR has the same functions as Dispatcher with the exception of high availability, SNMP subagent, wide area, and a few other configuration commands.

Caching Proxy must be running before CBR can begin load balancing client requests.

Managing local servers with CBR

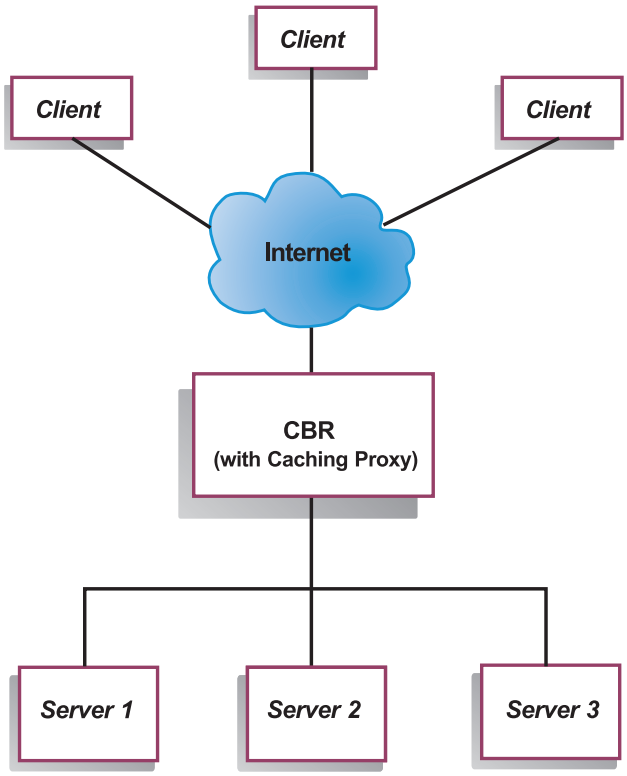


Figure 4. Example of a site using CBR to manage local servers

Figure 4 shows a logical representation of a site in which CBR is being used to proxy some content from local servers. The CBR component uses Caching Proxy to forward client requests (HTTP or HTTPS) to the servers based on the content of the URL.

Overview of the Site Selector component

Site Selector acts as a name server that works in conjunction with other name servers in a domain name system to load balance among a group of servers using measurements and weights that are gathered. You can create a site configuration to let you load balance traffic among a group of servers based on the domain name used for a client's request.

A client submits a request for resolution of a domain name to a name server within its network. Name server forwards the request to the Site Selector machine. Site Selector then resolves the domain name to the IP address of one of the servers that has been configured under the site name. Site Selector returns the IP address of the selected server to the name server. The name server returns the IP address to the client.

Metric Server is a system monitoring component of Load Balancer that must be installed in each load-balanced server within your configuration. Using Metric Server, Site Selector can monitor the level of activity on a server, detect when a server is the least heavily loaded, and detect a failed server. The load is a measure of how hard the server is working. By customizing system metric script files, you can control the type of measurements used to measure the load. You can configure Site Selector to suit your environment, considering such factors as frequency of access, the total number of users, and types of access (for example, short queries, long-running queries, or CPU-intensive loads).

Managing local and remote servers with Site Selector and Metric Server

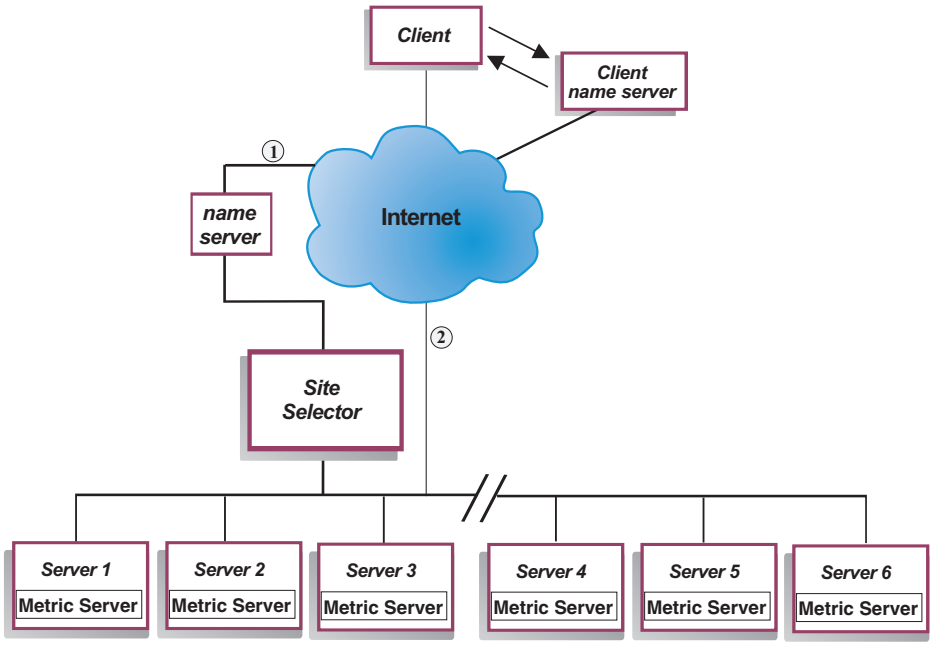


Figure 5. Example of a site using Site Selector and Metric Server to manage local and remote servers

Figure 5 illustrates a site in which the Site Selector component is used to answer requests. Server1, Server2, and Server3 are local. Server4, Server5, and Server6 are remote.

A client submits a request for resolution of a domain name to a client name server. The client name server forwards the request through the DNS to the Site Selector machine (Path 1). Site Selector then resolves the domain name to the IP address of one of the servers. Site Selector returns the IP address of the selected server to the client name server. The name server returns the IP address to the client.

After the client receives the IP address of the server, the client routes application requests directly to the selected server (Path 2).

Note: In this example, the Metric Server provides system load information to the Site Selector machine. The Metric Server agent is installed on each backend server. Use Metric Server in conjunction with Site Selector; otherwise Site Selector can only use a round-robin selection method for load balancing.

Overview of the Cisco CSS Controller component

Note: The Cisco CSS Controller component is shipped with Load Balancer for IPv4, but this component might not support newer hardware. Consult the prerequisites page for supported hardware: *http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921*

Cisco CSS Controller forms a complementary solution in conjunction with Cisco's CSS 11000 series switches. The combined solution blends the CSS 11000 series'

robust packet forwarding and content routing capabilities with Load Balancer's sophisticated awareness algorithms for determining load information and availability of the *service* (backend server application or database). The Cisco CSS Controller function utilizes Load Balancer's weight calculation algorithm, standard and custom advisors, and Metric Server to determine the metrics, health, and load of the service. With this information Cisco CSS Controller generates service weights, which it sends to the Cisco CSS Switch for optimal service selection, load optimization, and fault tolerance.

Cisco CSS Controller tracks many criteria, including:

- Active connections and connection rate (the number of new connections within a weight calculation cycle)
- Application and database availability, which is facilitated through the use of standard and customized advisors, and service-resident agents tailored to the specific application
- CPU utilization
- Memory utilization
- User-customizable system metrics

When a Cisco CSS Switch, without Cisco CSS Controller, is determining the health of a content-providing service, it uses response times for content requests or other network measures. With Cisco CSS Controller in place, these activities are offloaded from the Cisco CSS Switch to Cisco CSS Controller. Cisco CSS Controller influences the service's weight or ability to serve content, and activates or suspends a service as appropriate when the service regains or loses availability.

Cisco CSS Controller:

- Uses a published SNMP interface to obtain connection information from the Cisco CSS Switch
- Uses advisor input to analyze service availability and response time
- Uses Metric Server information to analyze system load
- Generates weights for each service in the configuration

Weights are applied to all services on a port. For any particular port, the requests are distributed between services based on their weights relative to each other. For example, if one service is set to a weight of 10, and the other to 5 the service set to 10 gets twice as many requests as the service set to 5. These weights are provided to the Cisco CSS Switch using SNMP. As the weight of any service is set higher, the Cisco CSS Switch directs more requests to that service.

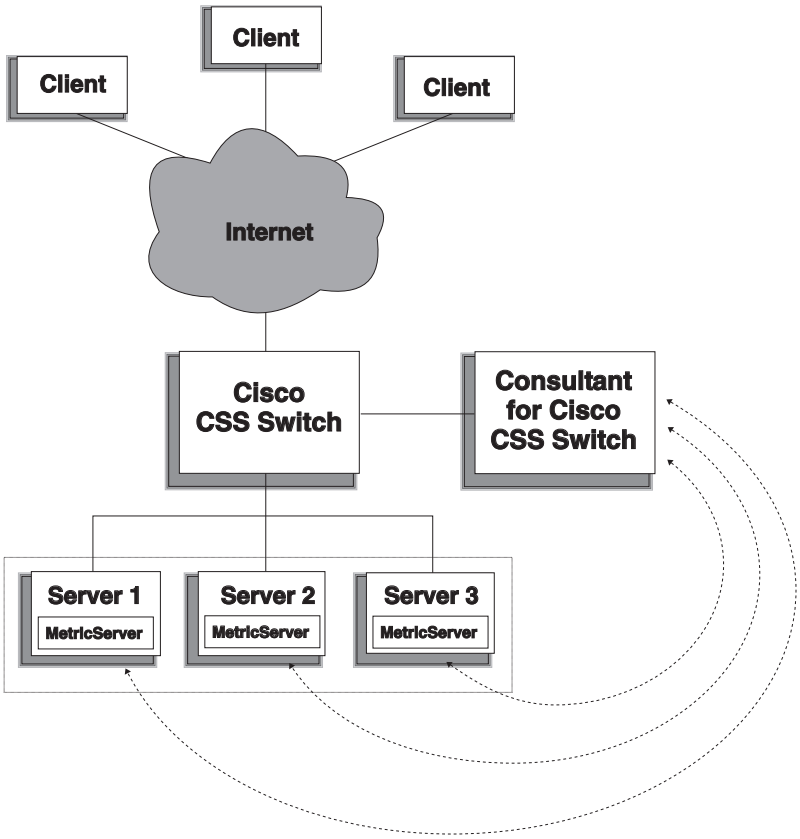


Figure 6. Example of a site using Cisco CSS Controller and Metric Server to manage local services

Cisco CSS Controller, in conjunction with the Cisco CSS Switch, delivers a "best of both worlds" solution that combines wire-speed content switching with sophisticated application awareness, fault tolerance, and service load optimization. Cisco CSS Controller is part of an overall complementary solution between the Cisco CSS Switch and IBM WebSphere Application Server Load Balancer.

Overview of Nortel Alteon Controller component

Note: The Nortel Alteon Controller component is shipped with Load Balancer for IPv4, but this component might not support newer hardware. Consult the prerequisites page for supported hardware: *http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921*

Nortel Alteon Controller in conjunction with the Nortel Alteon family of Web switches provides a complementary solution that combines the switches' packet forwarding speed and capacity with the Load Balancer's sophisticated awareness algorithms for determining server weights.

Nortel Alteon Controller allows you to develop custom advisors that are capable of performing more intelligent, application-aware assessments of the availability and load of applications used to deploy services.

The Metric Server provides system load information, such as CPU and memory utilization information, and a framework for you to develop custom system load measurements.

Nortel Alteon Controller collects many types of metric data to determine weights for servers being load-balanced by Nortel Alteon Web Switches, including:

- Active and new connections
- Application and database availability, which is facilitated through the use of standard and customized advisors, and server-resident agents tailored to the specific application
- CPU utilization
- Memory utilization
- User-customizable server metrics
- Reachability

Nortel Alteon Controller uses SNMP to communicate with the switch. Configuration , state and connection information is retrieved from the switch. When server weights are calculated by the controller, they are set on the switch. The switch uses the weights set by the controller to select the best server to handle client requests for a service.

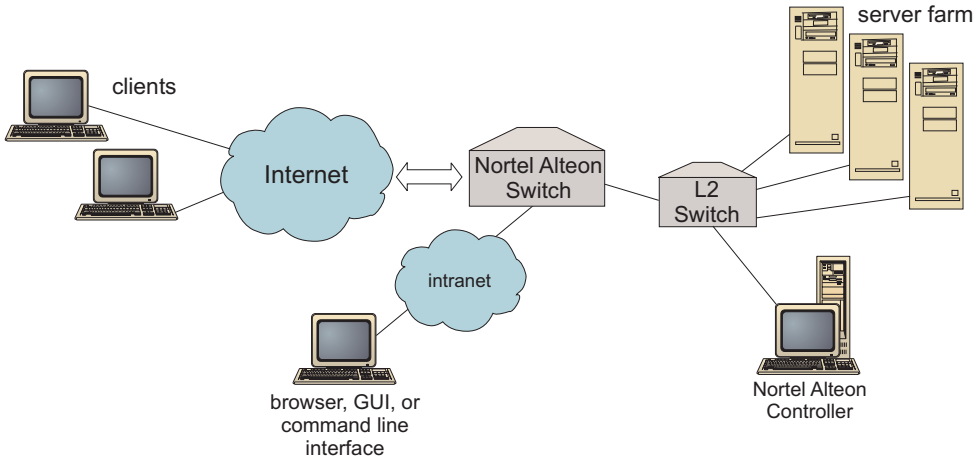


Figure 7. Example of a site using Nortel Alteon Controller to manage local servers

You can manage the controller using a browser, a remote GUI, or a remote command line interface.

Nortel Alteon Controller combined with the Nortel Alteon family of Web switches delivers a "best of both worlds" solution that combines wire-speed packet switching with sophisticated application awareness, fault tolerance and server load optimization. Nortel Alteon Controller is part of a complementary solution between the Nortel Alteon family of Web switches and IBM's WebSphere.

Chapter 3. Managing your network: Determining which Load Balancer features to use

This chapter lists the configuration features of the Load Balancer components so you can determine which features to use for managing your network:

- "Manager, Advisors, and Metric Server functions (for Dispatcher, CBR, and Site Selector components)"
- "Dispatcher component features"
- "Content Based Routing (CBR) component features" on page 20
- "Site Selector component features" on page 22
- "Cisco CSS Controller component features" on page 23
- "Nortel Alteon Controller component features" on page 24

Manager, Advisors, and Metric Server functions (for Dispatcher, CBR, and Site Selector components)

To optimize balancing the load across servers and ensure that the "right" server is chosen, see:

__ "Optimizing the load balancing provided by Load Balancer" on page 142

__ "Advisors" on page 146

__ "Metric Server" on page 157

Dispatcher component features

Dispatcher supports load balancing across your servers for HTTP, FTP, SSL, SMTP, NNTP, IMAP, POP3, Telnet, SIP, and any other TCP or stateless UDP based application.

Remote administration

__ To run Load Balancer configuration from a separate machine from the one on which the Load Balancer resides, see "Remote administration of Load Balancer" on page 213.

Collocation

__ To run Dispatcher on the same machine as a Web server that you are load balancing, see "Using collocated servers" on page 162.

High availability

__ To use Dispatcher to remove single point-of-failure limitations in your network, see "Simple high availability" on page 45 and "Mutual high availability" on page 46.

Client to server affinity

When load balancing SSL (HTTPS) traffic:

__ To ensure that the client uses the same SSL server for multiple connections, see "How affinity feature for Load Balancer works" on page 178.

- __ To ensure that the client uses the same server for HTTP and SSL traffic, see “Cross port affinity” on page 179.
- __ To ensure that the client uses the same server for multiple connections, see “How affinity feature for Load Balancer works” on page 178.
- __ To ensure that a group of clients use the same server for multiple connections, see “Affinity address mask (stickymask)” on page 179.
- __ To remove a server from your configuration (for example, for maintenance purposes) without disrupting client traffic, see “Quiesce server connection handling” on page 180.

Rules-based load balancing

In order to direct clients to different sets of servers for the same Web address, you can add "rules" to your Dispatcher configuration. For more information, see “Configure rules-based load balancing” on page 170.

- __ To direct clients to different sets of servers based on client source IP address, see “Using rules based on the client IP address” on page 171.
- __ To direct clients to different sets of servers based on client port, see “Using rules based on the client port” on page 171.
- __ To direct clients to different sets of servers based on time of day, see “Using rules based on the time of day” on page 172.
- __ To direct clients to servers based on Type of Service (TOS) bits in network packets, see “Using rules based on type of service (TOS)” on page 172.
- __ To direct clients to different sets of servers based on site traffic:
  - __ Using connections per second, see “Using rules based on the connections per second” on page 172.
  - __ Using total active connections, see “Using rules based on the total active connections” on page 173.
  - __ Reserving and sharing bandwidth for different Web addresses, see “Using rules based on reserved bandwidth and shared bandwidth” on page 173.
  - __ Ensuring traffic is measured correctly for each of your sets of servers, see “Server evaluation option for rules” on page 177.
- __ To direct overflow traffic to a default set of servers (for example, servers that will respond "site busy"), see “Using rules that are always true” on page 175.
- __ To override client affinity to ensure that a client does not "stick" to an overflow server, see “port affinity override” on page 176.

Content-based routing using Dispatcher's cbr forwarding method

To ensure SSL clients return to the same SSL server based on SSL ID in the client request, see the section on configuring content-based routing with SSL.

To direct HTTP clients to different sets of servers using rules based on matching the URL content of the client request, see “Dispatcher's content-based routing (cbr forwarding method)” on page 41 and “Using rules based on the request content” on page 176 for more information.

- __ To distinguish between particular URLs and their service applications, see “Server Partitioning: logical servers configured to one physical server (IP address)” on page 43.

__ To ensure clients return to the same server when requesting similar content in multiple connections using cookies created by your Web servers, see "Passive cookie affinity" on page 183.

__ To load balance Web traffic to caching-proxy servers that allow unique content to be cached on each server (thereby increasing the size of your site's cache by eliminating redundant caching of content on multiple machines), see "URI affinity" on page 184.

Comparison between the Dispatcher component's cbr forwarding method and the CBR component

The advantage of using Dispatcher's cbr forwarding method is that it provides a faster response to client requests than the CBR component. Also, Dispatcher's cbr forwarding does *not* require the installation and use of Caching Proxy.

If your network includes fully secure SSL (client through server) traffic, the advantage of using the CBR component (in conjunction with Caching Proxy) is that it can process the encryption and decryption required in order to perform content-based routing. For fully secure connections, the Dispatcher's cbr forwarding can only be configured with SSL ID affinity because it cannot process the encryption and decryption to perform true content-based routing on the client request's URL.

Wide area load balancing

Wide area load balancing can be achieved through several different methods.

__ To load balance to remote servers using the wide area feature of Dispatcher, see: "Configure wide area Dispatcher support" on page 184 and "GRE (Generic Routing Encapsulation) support" on page 189.

Note: An additional Dispatcher is required at the remote site if GRE is not supported at the remote site.

__ To load balance to remote servers using Dispatcher's nat forwarding method, see "Dispatcher's NAT/NAPT (nat forwarding method)" on page 39.

Note: *No* additional Dispatcher is required at the remote site if nat forwarding method is used.

Port mapping

__ To load balance one Web address to multiple server daemons on the same machine, where each daemon listens on a unique port see "Dispatcher's NAT/NAPT (nat forwarding method)" on page 39.

Setting up Dispatcher on a private network

__ To put Dispatcher traffic on a different network than client traffic (to improve performance by reducing contention on the external network), see "Using a private network configuration" on page 190.

Wildcard cluster and wildcard port

__ To combine multiple Web addresses into a single configuration, see "Use wildcard cluster to combine server configurations" on page 191.

__ To load balance firewalls, see "Use wildcard cluster to load balance firewalls" on page 191.

__ To direct traffic for all destination ports, see "Use wildcard port to direct unconfigured port traffic" on page 192.

"Denial of service" attack detection

__ To detect possible "denial of service" attacks, see “Denial of service attack detection” on page 193.

Binary logging

__ To analyze server traffic, see “Using binary logging to analyze server statistics” on page 194.

Alerts

__ To generate alerts when servers are marked up or down, see “Using scripts to generate an alert or record server failure” on page 145.

Content Based Routing (CBR) component features

CBR integrates load balancing with WebSphere Application Server's Caching Proxy to proxy client requests to specified HTTP or HTTPS (SSL) servers. To use CBR, Caching Proxy must be installed and configured on the same machine. For information on how to configure Caching Proxy in order to use CBR, see “Step 1. Configure Caching Proxy to use CBR” on page 79.

Note: The Content Based Routing (CBR) component is not available on platforms that run a 64-bit JVM, except for HP-UX ia64. On HP-UX ia64, the CBR component runs as a 32-bit application. You can use the CBR forwarding method of Load Balancer's Dispatcher component to provide content-based routing without the use of Caching Proxy. See “Dispatcher's content-based routing (cbr forwarding method)” on page 41 for more information.

With the CBR component (or the Dispatcher component's cbr forwarding method), you can provide the following advantages to your clients:

__ Load balance client requests for different types of content to sets of servers. (See “Load balancing requests for different types of content” on page 72.)

__ Improve response time by optimally dividing your site's content among your Web servers. (See “Dividing your site content for better response time” on page 72.)

__ Ensure uninterrupted client traffic during server failure by allowing multiple servers to be assigned to each type of content. (See “Providing backup of Web server content” on page 72.)

Comparison between the CBR component and the Dispatcher component's cbr forwarding method

If your network requires fully secure SSL traffic (client through server), the advantage of using the CBR component (in conjunction with Caching Proxy) is that it can process SSL encryption/decryption in order to perform content-based routing.

For fully secure SSL connections, the Dispatcher's cbr forwarding can only be configured with SSL ID affinity because it cannot process the encryption/decryption to perform true content-based routing on the client request's URL.

For HTTP traffic, the advantage of using Dispatcher's cbr forwarding method is that it provides a faster response to client requests than the CBR component. Also, Dispatcher's cbr forwarding does *not* require the installation and use of Caching Proxy.

Remote administration

__ To run Load Balancer configuration from a separate machine from the one on which the Load Balancer resides, see “Remote administration of Load Balancer” on page 213.

Collocation

__ CBR can run on the same machine as a server that you are load balancing. See “Using collocated servers” on page 162 for more information.

CBR with multiple instances of Caching Proxy

__ To improve CPU utilization by using multiple Caching Proxy processes, see “Using multiple Caching Proxy processes to improve CPU utilization” on page 72.

Provide content-based routing for SSL connections

To allow content-based routing of SSL traffic:

__ Using secure connection on both sides (client-to-proxy and proxy-to-server), see “Load balancing across fully secure (SSL) connections” on page 73.

__ Using secure connections only on the client-to-proxy side, see “Load balancing client-to-proxy in SSL and proxy-to-server in HTTP” on page 73.

Server partitioning

__ To distinguish between particular URLs and their service applications, see “Server Partitioning: logical servers configured to one physical server (IP address)” on page 43.

Rules-based load balancing

In order to direct clients to different sets of servers for the same Web address, you can add "rules" to your CBR configuration. For more information, see “Configure rules-based load balancing” on page 170.

__ To direct clients to different sets of servers based on the content of the requested URL, see “Using rules based on the request content” on page 176.

__ To direct clients to different sets of servers based on client source IP address, see “Using rules based on the client IP address” on page 171.

__ To direct clients to different sets of servers based on time of day, see “Using rules based on the time of day” on page 172.

__ To direct clients to different sets of servers based on site traffic:

  Using connections per second, see “Using rules based on the connections per second” on page 172.

  Using total active connections, see “Using rules based on the total active connections” on page 173.

__ To direct overflow traffic to a default set of servers (for example, server(s) that will respond "site busy"), see “Using rules that are always true” on page 175.

__ To override client affinity to ensure that a client does not "stick" to an overflow server, see “port affinity override” on page 176.

Client to server affinity

__ To ensure that a client returns to the same server for multiple connections, see “How affinity feature for Load Balancer works” on page 178.

- To remove a server from your configuration (for example, for maintenance purposes) without disrupting client traffic, see "Quiesce server connection handling" on page 180.
- To ensure that clients return to the same server when requesting similar content in multiple connections without relying on cookies created by your Web servers, see "Active cookie affinity" on page 181.
- To ensure that clients return to the same server when requesting similar content in multiple connections using cookies created by your Web servers, see "Passive cookie affinity" on page 183.
- To load balance Web traffic to caching-proxy servers that allow unique content to be cached on each server (thereby increasing the size of your site's cache by eliminating redundant caching of content on multiple machines), see "URI affinity" on page 184.

High availability using Dispatcher and CBR

- To remove single point of failure limitations in your network using Dispatcher in a two-tier configuration with CBR, see "How can Load Balancer provide high availability?" on page 5.

Binary logging

- To analyze server traffic, see "Using binary logging to analyze server statistics" on page 194.

Alerts

- To generate alerts when servers are marked up or down, see "Using scripts to generate an alert or record server failure" on page 145.

Site Selector component features

Site Selector load balances a name service request across a group of servers.

Remote administration

- To run Load Balancer configuration from a separate machine from the one on which the Load Balancer resides, see "Remote administration of Load Balancer" on page 213.

Collocation

- Site Selector can run on the same machine as a server that you are load balancing with no additional configuration steps required.

High availability

- High availability is inherently available through Domain Name System (DNS) methodologies using multiple redundant Site Selectors, assuming correct configuration of the parent name server and normal DNS recovery methods are in place. Examples of normal DNS recovery methods are: retransmission of queries and retrying zone transfers.
- To remove single point of failure limitations in your network using Dispatcher in a two-tier configuration with Site Selector, see "How can Load Balancer provide high availability?" on page 5.

Client to server affinity

__ To ensure that the client uses the same server for multiple name server requests, see “How affinity feature for Load Balancer works” on page 178.

__ To ensure client to server affinity using the standard DNS method of setting the Time To Live (TTL), see “TTL considerations” on page 93.

Rules-based load balancing

To direct client requests to different sets of servers for domain name resolution, you can add "rules" to your Site Selector configuration. For more information, see “Configure rules-based load balancing” on page 170.

__ To direct clients to different sets of servers based on client source IP address, see “Using rules based on the client IP address” on page 171.

__ To direct clients to different sets of servers based on time of day, see “Using rules based on the time of day” on page 172.

__ To direct clients to different sets of servers based on the metric load values of the server set, see:

“Metric all rule” on page 175

“Metric average rule” on page 175

__ To direct overflow traffic to a default set of servers (for example, server(s) that will respond "site busy"), see “Using rules that are always true” on page 175.

Wide area load balancing

Site Selector can run in both a local area network (LAN) or a wide area network (WAN).

In a WAN environment:

__ To load balance client name server requests using a weighted round-robin selection method, no additional configuration steps are required.

__ To consider the network proximity of the client name server to the servers providing the application requested (the destination servers), see “Using the Network Proximity feature” on page 93.

Alerts

__ To generate alerts when servers are marked up or down, see “Using scripts to generate an alert or record server failure” on page 145.

Cisco CSS Controller component features

Note: The Cisco CSS Controller component is shipped with Version 8.0 of Load Balancer for IPv4, but this component might not support newer hardware. Consult the prerequisites page for supported hardware: *http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921*

Cisco CSS Controller enhances the Cisco switches' server load-balancing capabilities with greater application and system awareness. The controller uses more application sensitive and system sensitive metrics to calculate server weights dynamically. The weights are provided to the switch using SNMP. The switch uses the weights when processing client requests resulting in server load optimization and improved fault tolerance.

To optimize balancing the load across servers and ensure that the "right" server is chosen, see:

__ "Optimizing the load balancing provided by Load Balancer" on page 200

__ "Advisors" on page 201 and "Create custom (customizable) advisors" on page 203

__ "Metric Server" on page 206

Remote administration

__ To run Load Balancer configuration from a separate machine from the one on which the Load Balancer resides, see "Remote administration of Load Balancer" on page 213.

Collocation

__ Cisco CSS Controller can run on the same machine as a server that you are load balancing with no additional configuration steps required.

High availability

__ To remove single point of failure limitations in your network, both the Cisco CSS Switch and the Cisco CSS Controller have high availability capabilities. For the switch, high availability capabilities are possible using the CSS redundancy protocol. For the Cisco CSS Controller, a proprietary protocol is used that allows the hot-standby configuration of two controllers.

For more information about configuring high availability, see "High availability" on page 110.

Binary logging

__ To analyze server traffic, see "Using binary logging to analyze server statistics" on page 209.

Alerts

__ To generate alerts when servers are marked up or down, see "Using scripts to generate an alert or record server failure" on page 210.

Nortel Alteon Controller component features

Note: The Nortel Alteon Controller component is shipped with Load Balancer for IPv4, but this component might not support newer hardware. Consult the prerequisites page for supported hardware: *http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921*

Nortel Alteon Controller enhances the Nortel Alteon switches' server load-balancing capabilities with greater application and system awareness. The controller uses more application sensitive and system sensitive metrics to calculate server weights dynamically. The weights are provided to the switch using SNMP. The switch uses the weights when processing client requests resulting in server load optimization and improved fault tolerance.

To optimize balancing the load across servers and ensure that the "right" server is chosen, see:

__ "Optimizing the load balancing provided by Load Balancer" on page 200

__ "Advisors" on page 201 and "Create custom (customizable) advisors" on page 203

__ "Metric Server" on page 206

Remote administration

__ To run Load Balancer configuration from a separate machine from the one on which the Load Balancer resides, see "Remote administration of Load Balancer" on page 213.

Collocation

__ Nortel Alteon Controller can run on the same machine as a server that you are load balancing with no additional configuration steps required.

High availability

__ To remove single point of failure limitations in your network, both the Nortel Alteon Web Switch and the Nortel Alteon Controller have high availability capabilities. For the switch, high availability is possible using redundancy protocol for connections to servers and for services. Nortel Alteon Controller provides high availability using a proprietary protocol that allows a hot-standby configuration of two controllers.

For more information about configuring high availability, see "High availability" on page 130.

Binary logging

__ To analyze server traffic, see "Using binary logging to analyze server statistics" on page 209.

Alerts

__ To generate alerts when servers are marked up or down, see "Using scripts to generate an alert or record server failure" on page 210.

Chapter 4. Installing Load Balancer

For instructions on installing Load Balancer, refer to the installation instructions in Concepts, Planning and Installation for IPv4.

Note: Throughout the administration guide, there are references to commands and file locations that include directory paths:

- For AIX, HP-UX, Linux, and Solaris operating systems, the installation path for the product is **/opt/ibm/edge/lb/**

  This path cannot be changed.

- For Windows operating systems, the default installation path for the product is **C:\Program Files\loadbalancer\ibm\edge\lb**.

  The beginning of this installation path can be changed, but be aware that Installation Manager requires you to install the product into an empty directory - in the default path it is **C:\Program Files\loadbalancer**. The **ibm\edge\lb** portion of the path is hard coded and cannot be modified.

  Since the initial section of the path can be modified, the installation location is referenced as <*install_root*>ibm\edge\lb\ throughout the administration guide, where <*install_root*> represents the empty directory that you specified during installation.

Part 2. Dispatcher component

This part provides information on a quick start configuration, planning considerations, and describes the methods of configuring Load Balancer's Dispatcher component. It contains the following chapters:

Chapter 5. Quick start configuration

This quick start example shows how to configure three locally attached workstations using the Dispatcher component's mac forwarding method to load-balance Web traffic between two Web servers. The configuration would be essentially the same for balancing any other TCP or stateless UDP application traffic.

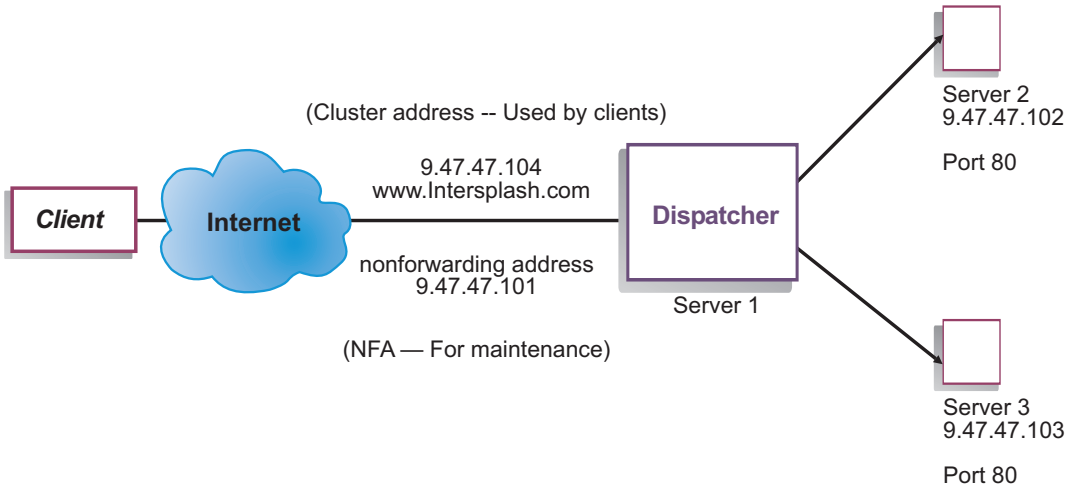


Figure 8. A simple local Dispatcher configuration

The mac forwarding method is the default forwarding method whereby Dispatcher load balances incoming requests to the server, and the server returns the response directly to the client. For more information on Dispatcher's MAC forwarding method, see "Dispatcher's MAC-level routing (mac forwarding method)" on page 38.

Note: You can complete the configuration using only two workstations with Dispatcher located on one of the Web server workstations. This setup represents a collocated configuration. Procedures for setting up more complex configurations can be found at "Setting up the Dispatcher machine" on page 50.

What do you need?

For the quick start example, you need three workstations and four IP addresses. One workstation is the Dispatcher machine; the other two workstations are the Web servers. Each Web server requires one IP address. The Dispatcher workstation requires two addresses: the nonforwarding address (NFA), and the cluster address (the address which is load balanced) that you provide to clients to access your Web site.

Note: The NFA is the address that is returned by the **hostname** command. This address is used for administrative purposes, such as remote configuration.

How do you prepare?

1. For this locally attached configuration example, set up your workstations on the same LAN segment. Ensure that network traffic between the three machines does not have to pass through any routers or bridges. (For setting up configurations with remote servers, see "Configure wide area Dispatcher support" on page 184.)
2. Configure the network adapters of the three workstations. For this example, we will assume you have the following network configuration:

Workstation	Name	IP Address
1	server1.Intersplashx.com	9.47.47.101
2	server2.Intersplashx.com	9.47.47.102
3	server3.Intersplashx.com	9.47.47.103
Netmask = 255.255.255.0		

   Each of the workstations contains only one standard Ethernet network interface card.
3. Ensure that server1.Intersplashx.com can ping both server2.Intersplashx.com and server3.Intersplashx.com.
4. Ensure that server2.Intersplashx.com and server3.Intersplashx.com can ping server1.Intersplashx.com.
5. Ensure that content is identical on the two Web servers (Server 2 and Server 3). This can be done by replicating data on both workstations, by using a shared file system such as NFS, AFS®, or DFS, or by any other means appropriate for your site.
6. Ensure that Web servers on server2.Intersplashx.com and server3.Intersplashx.com are operational. Use a Web browser to request pages directly from **http://server2.Intersplashx.com** and **http://server3.Intersplashx.com**.
7. Obtain another valid IP address for this LAN segment. This is the address you will provide to clients who wish to access your site. For this example we will use:

   ```
   Name= www.Intersplashx.com
   IP=9.47.47.104
   ```
8. Configure the two Web server workstations to accept traffic for www.Intersplashx.com.

   Add an alias for www.Intersplashx.com to the **loopback** interface on server2.Intersplashx.com and server3.Intersplashx.com.

   - For AIX systems:

     ifconfig lo0 alias www.Intersplashx.com netmask 255.255.255.255
   - For Solaris 9 systems:

     ifconfig lo0:1 plumb www.Intersplashx.com netmask 255.255.255.0 up
   - For other operating systems see Table 2 on page 56.
9. Delete any extra route that may have been created as a result of aliasing the loopback interface. See "Step 2. Check for an extra route" on page 59.

   You have now completed all configuration steps that are required on the two Web server workstations.

Configuring the Dispatcher component

With Dispatcher, you can create a configuration by using the command line, the configuration wizard, or the graphical user interface (GUI).

Note: The parameter values must be typed in English characters. The only exceptions are parameter values for host names and file names.

Configuring with the command line

If you are using the command line, follow these steps:

1. Start the dsserver on Dispatcher:
   - For AIX, HP-UX, Linux, or Solaris systems, run the following command as root user: **dsserver**
   - For Windows systems, dsserver runs as a service that starts automatically.
2. Start the executor function of Dispatcher:

   dscontrol executor start
3. Add the cluster address to the Dispatcher configuration:

   dscontrol cluster add www.Intersplashx.com
4. Add the HTTP protocol port to the Dispatcher configuration:

   dscontrol port add www.Intersplashx.com:80
5. Add each of the Web servers to the Dispatcher configuration:

   dscontrol server add www.Intersplashx.com:80:server2.Intersplashx.com

   dscontrol server add www.Intersplashx.com:80:server3.Intersplashx.com
6. Configure the workstation to accept traffic for the cluster address:

   dscontrol executor configure www.Intersplashx.com
7. Start the manager function of Dispatcher:

   dscontrol manager start

   Dispatcher will now do load balancing based on server performance.
8. Start the advisor function of Dispatcher:

   dscontrol advisor start http 80

   Dispatcher will now make sure that client requests are not sent to a failed Web server.

Your basic configuration with locally attached servers is now complete.

Testing your configuration

Test to see if the configuration is working:

1. From a Web browser, go to location **http://www.Intersplashx.com**. If a page is displayed, the configuration is working.
2. Reload the page in the Web browser.
3. Look at the results of the following command: **dscontrol server report www.Intersplashx.com:80:**. The total connections column of the two servers should add up to "2."

Configuring with the graphical user interface (GUI)

For information about using the Dispatcher GUI, see "GUI" on page 48 and Appendix A, "GUI: General instructions," on page 403.

Configuration Wizard

For information about using the configuration wizard, see “Configuring with the configuration wizard” on page 49.

Types of cluster, port, server configurations

There are many ways that you can configure Load Balancer to support your site. If you have only one host name for your site to which all of your customers will connect, you can define a single cluster of servers. For each of these servers, configure a port through which Load Balancer communicates. See Figure 9.

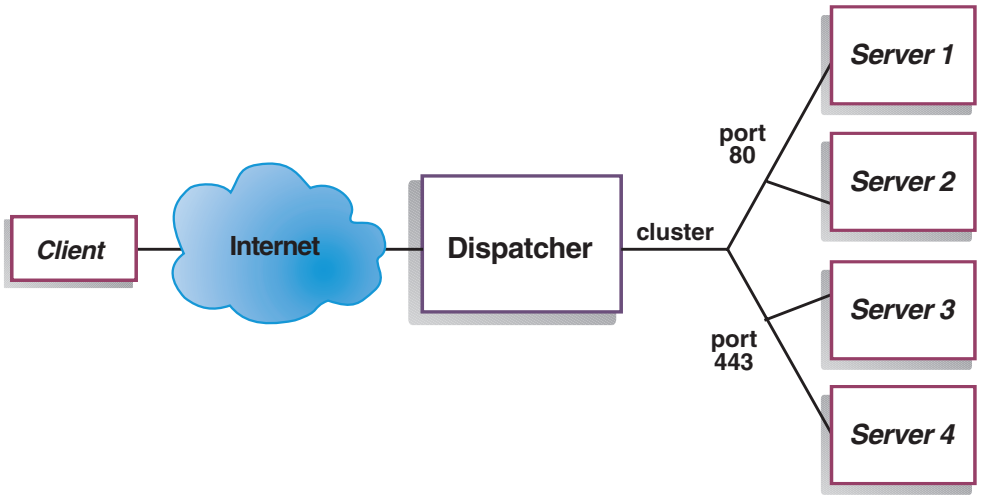


Figure 9. Example of Dispatcher configured with a single cluster and 2 ports

In this example for the Dispatcher component, one cluster is defined at www.productworks.com. This cluster has two ports: port 80 for HTTP and port 443 for SSL. A client making a request to http://www.productworks.com (port 80) goes to a different server than a client requesting https://www.productworks.com (port 443).

Another way of configuring Load Balancer might be appropriate if you have a very large site with many servers dedicated to each protocol supported. In this case, you might want to define a cluster for each protocol with a single port but with many servers, as shown in Figure 10 on page 35.

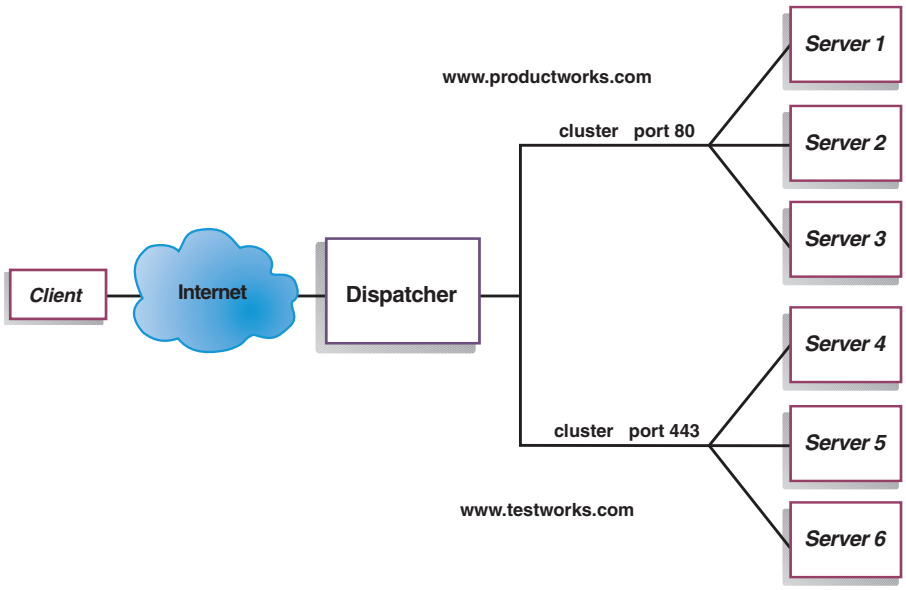


Figure 10. Example of Dispatcher configured with two clusters, each with one port

In this example for the Dispatcher component, two clusters are defined: www.productworks.com for port 80 (HTTP) and www.testworks.com for port 443 (SSL).

A third way of configuring Load Balancer might be necessary if your site does content hosting for several companies or departments, each one coming into your site with a different URL. In this case, you might want to define a cluster for each company or department and then define any ports to which you want to receive connections at that URL, as shown in Figure 11 on page 36.

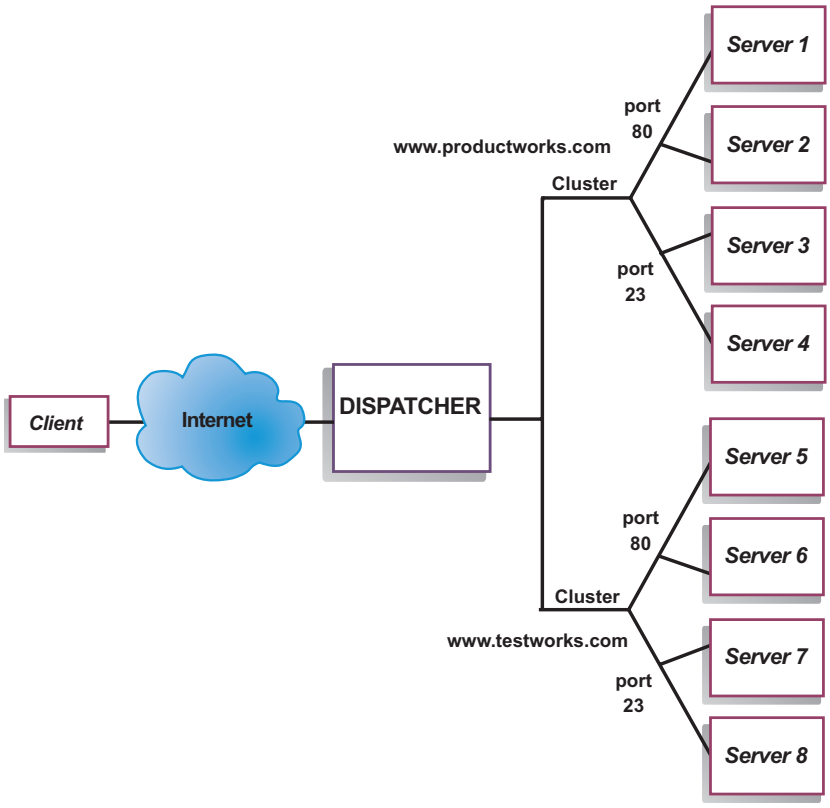


Figure 11. Example of Dispatcher configured with 2 clusters, each with 2 ports

In this example for the Dispatcher component, two clusters are defined with port 80 for HTTP and port 23 for Telnet for each of the sites at www.productworks.com and www.testworks.com.

Chapter 6. Planning for Dispatcher

This chapter describes what the network planner should consider before installing and configuring the Dispatcher component.

- See Chapter 3, "Managing your network: Determining which Load Balancer features to use," on page 17 for an overview of features that are available for managing your network.
- See Chapter 7, "Configuring Dispatcher," on page 47 for information on configuring the load-balancing parameters of Dispatcher.
- See Chapter 21, "Advanced features for Dispatcher, CBR, and Site Selector," on page 161 for information on how to set up Load Balancer for more advanced functions.
- See Chapter 23, "Operating and managing Load Balancer," on page 213 for information on remote authenticated administration, Load Balancer logs, and usage of the Load Balancer components.

This chapter includes the following sections:

Note: For previous versions, when the product was known as Network Dispatcher, the Dispatcher control command name was ndcontrol. The Dispatcher control command name is now **dscontrol**.

Planning considerations

Dispatcher consists of the following functions:

- **dsserver** handles requests from the command line to the executor, manager, and advisors.
- The **executor** supports port-based load balancing of TCP and UDP connections. It is able to forward connections to servers based on the type of request received (for example, HTTP, FTP, SSL, and so forth). The executor always runs when the Dispatcher component is being used for load balancing.
- The **manager** sets weights used by the executor based on:
  - Internal counters in the executor
  - Feedback from the servers provided by the advisors
  - Feedback from a system-monitoring program, such as Metric Server or WLM.

  Using the manager is optional. However, if the manager is not used, load balancing is performed using weighted round-robin scheduling based on the current server weights, and advisors are not available.

- The **advisors** query the servers and analyze results by protocol before calling the manager to set weights as appropriate. Currently there are advisors available for the following protocols: HTTP, FTP, SSL, SMTP, NNTP, IMAP, POP3, SIP, and Telnet.

  Dispatcher also offers advisors that do not exchange protocol-specific information, such as the DB2® advisor that reports on the health of DB2 servers and the ping advisor that reports whether the server responds to a ping. For a complete list of advisors, see "List of advisors" on page 149.

  You also have the option of writing your own advisors (see "Create custom (customizable) advisors" on page 154).

  Using the advisors is optional but recommended.
- To configure and manage the executor, advisors, and manager, use the command line (**dscontrol**) or the graphical user interface (**lbadmin**).
- A **sample configuration file** is provided to use for configuration and administration of the Dispatcher machine. See Appendix C, "Sample configuration files," on page 413. After you have installed the product, this file can be found in the following directory:
  - AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/samples**
  - Windows operating systems: **<***install_root***>ibm\edge\lb\servers\samples**
- The **SNMP subagent** allows an SNMP-based management application to monitor the status of the Dispatcher.

The three key functions of Dispatcher (executor, manager, and advisors) interact to balance and dispatch the incoming requests between servers. Along with load balancing requests, the executor monitors the number of new connections, active connections, and connections in a finished state. The executor also does garbage collection of completed or reset connections and supplies this information to the manager.

The manager collects information from the executor, the advisors, and a system-monitoring program, such as Metric Server. Based on the information the manager receives, it adjusts how the server machines are weighted on each port and gives the executor the new weighting for use in its balancing of new connections.

The advisors monitor each server on the assigned port to determine the server's response time and availability and then give this information to the manager. The advisors also monitor whether a server is up or down. Without the manager and the advisors, the executor does round-robin scheduling based on the current server weights.

Forwarding methods

With Dispatcher, you can select one of three forwarding methods specified at the port level: MAC forwarding, NAT/NAPT forwarding, or CBR (content-based routing) forwarding.

Dispatcher's MAC-level routing (mac forwarding method)

Using Dispatcher's MAC forwarding method (the default forwarding method), Dispatcher load balances the incoming request to the selected server and the server returns the response *directly* to the client without any involvement of the Dispatcher. With this forwarding method, Dispatcher only looks at the inbound

client-to-server flows. It does not need to see the outbound server-to-client flows. This significantly reduces its impact on the application and can result in improved network performance.

The forwarding method can be selected when adding a port using the **dscontrol port add** *cluster:port* **method** *value* command. The default forwarding method value is **mac**. You can specify the method parameter only when the port is added. When you add the port, you cannot change the setting of the forwarding method. See "dscontrol port — configure ports" on page 319 for more information.

Linux limitation: Linux systems employ a host-based model of advertising hardware addresses to IP addresses using ARP. This model is incompatible with the backend server or the high availability collocation server requirements for Load Balancer's mac forwarding method. See "Linux loopback aliasing alternatives when using Load Balancer's mac forwarding" on page 60, which describes a number of solutions to alter the Linux system's behavior to make it compatible with Load Balancer's mac forwarding.

Linux limitation when using zSeries or S/390 servers: There are limitations when using zSeries or S/390 servers that have Open System Adapter (OSA) cards. See "Problem: On Linux, Dispatcher configuration limitations when using zSeries or S/390 servers that have Open System Adapter (OSA) cards" on page 264, for possible workarounds.

Dispatcher's NAT/NAPT (nat forwarding method)

Using Dispatcher's Network Address Translation (NAT) or Network Address Port Translation (NAPT) capability removes the limitation for load-balanced servers to be located on a locally attached network. When you want to have servers located at remote locations, you can use the NAT forwarding method technique rather than using a GRE/WAN encapsulation technique. You can also use the NAPT feature to access multiple server daemons residing on each load-balanced server machine, where each daemon listens on a unique port.

You can configure a server with multiple daemons in two different ways:

- With NAT, you can configure multiple server daemons to respond to requests to different IP addresses. This is also known as binding a server daemon to an IP address.
- With NAPT, you can configure multiple server daemons (running on the same physical server) to listen on different port numbers.

This application works well with upper-level application protocols such as HTTP, SSL, IMAP, POP3, NNTP, SMTP, Telnet, and so on.

Limitations:

- Dispatcher's implementation of NAT/NAPT is a *simple* implementation of this feature. It analyzes and operates upon only the contents of TCP/IP packet headers. It does not analyze the contents of the data portion of the packets. For Dispatcher, NAT/NAPT will not work with application protocols, such as FTP, which imbed the addresses or port numbers in the data portion of the messages. This is a well-known limitation of header-based NAT/NAPT.
- Dispatcher's NAT/NAPT cannot work in conjunction with the wildcard cluster or wildcard port feature.

You will need three IP addresses for the Dispatcher machine – nfa, cluster, and return address. To implement NAT/NAPT, do the following (see also "Sample steps for configuring Dispatcher's nat or cbr forwarding methods" on page 42):

- Set the **clientgateway** parameter on the **dscontrol executor set** command. Clientgateway is an IP address that is used as the router address through which traffic in the return direction is forwarded from Load Balancer to clients. This value must be set to a nonzero IP address before you can use NAT/NAPT. See "dscontrol executor — control the executor" on page 299 for more information.
- Add a port using the **dscontrol port add** *cluster:port* **method** *value* command. The forwarding method value should be set to **nat**. You can specify the method parameter only when the port is added. After you add the port, you cannot change the setting of the forwarding method. See "dscontrol port — configure ports" on page 319 for more information.

  Note: If you do not set client gateway address to a nonzero value, then the forwarding method can only be **mac** (MAC based forwarding method).
- Add a server using the mapport, returnaddress, and router parameters using the **dscontrol** command. For example:

  dscontrol server add *cluster:port:server* **mapport** *value* **returnaddress** *rtrnaddress* **router** *rtraddress*

  – **mapport** (optional)

    This maps the client request's destination port number (which is for Dispatcher) to the server's port number that Dispatcher uses to load balance the client's request. Mapport allows Load Balancer to receive a client's request on one port and to transmit it to a different port on the server machine. With mapport you can load balance a client's requests to a server machine that might have multiple server daemons running. The default for mapport is the client request's destination port number.

  – **returnaddress**

    The return address is a unique address or host name that you configure on the Dispatcher machine. Dispatcher uses the return address as its source address when load balancing the client's request to the server. This ensures that the server returns the packet to the Dispatcher machine rather than sending the packet directly to the client. (Dispatcher will then forward the IP packet to the client.) You must specify the return address value when adding the server. You cannot modify the return address unless you remove the server and then add it again. The return address cannot be the same as the cluster, server, or NFA address.

    When you use nat or cbr forwarding methods, you must define a return address for communication between Load Balancer and the backend servers. The number of connections that Load Balancer can keep active with the backend server is limited by the number of return addresses that are defined. Load Balancer uses ports that are based upon the return address only; not the return address and server combination. When all the available ports are in use, additional connections fail. In a busy environment, use multiple return addresses to prevent a shortage of available ports.

  – **router**

    The address of the router to the remote server. If this is a locally attached server, enter the server address, unless the server is located on the same machine as Load Balancer. In that case, continue to use the real router address.

For more information on the **dscontrol server** command using the mapport, returnaddress, and router parameters, see “dscontrol server — configure servers” on page 330.

Dispatcher's content-based routing (cbr forwarding method)

The Dispatcher component allows you to perform content-based routing for HTTP (using the "content" type rule) and HTTPS (using SSL session ID affinity) without having to use Caching Proxy. For HTTP and HTTPS traffic, the Dispatcher component's cbr forwarding method can provide faster content-based routing than the CBR component, which requires Caching Proxy.

For HTTP: Server selection for Dispatcher's content-based routing is based upon the contents of a URL or an HTTP header. It is configured using the "content" type rule. When configuring the content rule, specify the search string "pattern" and a set of servers to the rule. When processing a new incoming request, this rule compares the specified string with the client's URL or with the specified HTTP header in the client request.

If Dispatcher finds the string in the client request, Dispatcher forwards the request to one of the servers within the rule. Dispatcher then relays the response data from the server to the client ("cbr" forwarding method).

If Dispatcher does not find the string in the client request, Dispatcher does *not* select a server from the set of servers within the rule.

Note: The content rule is configured in the Dispatcher component the same way it is configured in the CBR component. Dispatcher can use the content rule for HTTP traffic. However, the CBR component can use the content rule for *both* HTTP and HTTPS (SSL) traffic.

For HTTPS (SSL): Dispatcher's content-based routing load balances based on the SSL ID session field of the client request. With SSL, a client request contains the SSL session ID of a prior session, and servers maintain a cache of their prior SSL connections. Dispatcher's SSL ID session affinity allows the client and server to establish a new connection using the security parameters of the previous connection with the server. By eliminating the renegotiation of SSL security parameters, such as shared keys and encryption algorithms, the servers save CPU cycles and the client gets a quicker response. In order to enable SSL session ID affinity: the **protocol** type specified for the port must be **SSL** and port **stickytime** must be set to a nonzero value. When stickytime has been exceeded, the client may be sent to a different server from the previous.

You will need three IP addresses for the Dispatcher machine – nfa, cluster, and return address. To implement Dispatcher's content-based routing (see also “Sample steps for configuring Dispatcher's nat or cbr forwarding methods” on page 42):

- Set the **clientgateway** parameter on the **dscontrol executor set** command. Clientgateway is an IP address that is used as the router address through which traffic in the return direction is forwarded from Dispatcher to clients. The clientgateway value defaults to zero. This value must be set to a nonzero IP address before you can add a content-based routing forwarding method. See “dscontrol executor — control the executor” on page 299 for more information.
- Add a port using the **method** parameter and the **protocol** parameter on the **dscontrol port add** command. The forwarding method value should be set to **cbr**. The port protocol type can be either HTTP or SSL. See “dscontrol port — configure ports” on page 319 for more information.

Note: If you do not set client gateway address to a nonzero value, then the forwarding method can only be the **mac** forwarding method.

- Add a server using the mapport, returnaddress, and router parameters

  dscontrol server add *cluster:port:server* **mapport** *value* **returnaddress** *rtrnaddress* **router** *rtraddress*

  Note: For information on configuring the server using mapport (optional), returnaddress and router parameters, see page 40.

- **For HTTP:** Configure using rules based on the client request content (rule type **content**). For example,

  dscontrol rule 125.22.22.03:80:contentRule1 **type** content **pattern** *pattern*

  Where *pattern* specifies the pattern to be used for the content type rule. For more information on the content rule type, see "Using rules based on the request content" on page 176. For more information on valid expressions for *pattern*, see Appendix B, "Content rule (pattern) syntax," on page 409.

Note: The connection record replication feature of high availability (which ensures that a client's connection will not drop when a backup Dispatcher machine takes over for the primary machine) is *not* supported with Dispatcher's content-based routing.

Sample steps for configuring Dispatcher's nat or cbr forwarding methods

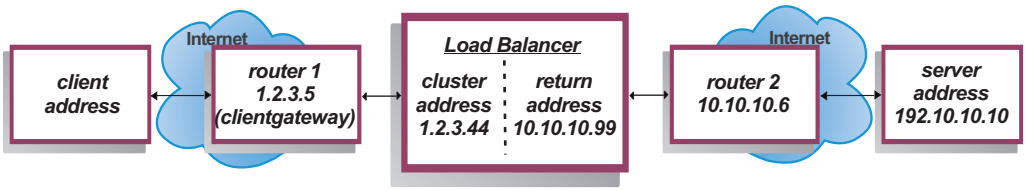


Figure 12. Example for using Dispatcher's nat or cbr forwarding methods

You will need at least three IP addresses for the Dispatcher machine. For Figure 12, the following are the necessary steps to minimally configure Dispatcher's nat or cbr forwarding methods:

```
1.Start the executor
  dscontrol executor start

2.Define the client gateway
  dscontrol executor set clientgateway 1.2.3.5
  NOTE: If your subnet does not have a local router, then you must
        configure a machine to do IP forwarding and use that as the
        clientgateway. Consult your operating system documentation
        to determine how to enable IP forwarding.

3.Define the cluster address
  dscontrol cluster add 1.2.3.44

4.Configure the cluster address
  dscontrol executor configure 1.2.3.44

5.Define the port with a method of nat or cbr
  dscontrol port add 1.2.3.44:80 method nat
  or
  dscontrol port add 1.2.3.44:80 method cbr protocol http
```

```
6.Configure an alias return address on Load Balancer (using ethernet card 0)
  NOTE: On Linux systems, you do not need to alias the return address if using
  nat forwarding on a collocated machine.

  dscontrol executor configure 10.10.10.99

  or use the ifconfig command (for Linux or UNIX only):
  AIX: ifconfig en0 alias 10.10.10.99 netmask 255.255.255.0
  HP-UX: ifconfig lan0:1 10.10.10.99 netmask 255.255.255.0 up
  Linux: ifconfig eth0:1 10.10.10.99 netmask 255.255.255.0 up
  Solaris: ifconfig eri0 addif 10.10.10.99 netmask 255.255.255.0 up

7.Define the backend servers
  dscontrol server add 1.2.3.4:80:192.10.10.10
    router 10.10.10.6 returnaddress 10.10.10.99
```

The client gateway (1.2.3.5) is the router 1 address between Load Balancer and the client. The router (10.10.10.6) is the router 2 address between Load Balancer and the backend server. If you are unsure of the clientgateway or router 2 address, you can use a traceroute program with the client (or server) address to determine the router address. The exact syntax of this program will differ based on the operating system you are using. You should consult your operating system documentation for more information regarding this program.

If the server is on the same subnet as Load Balancer (that is, no routers are returned using traceroute) enter the server address as the router address. However, if the server is located on the same machine as Load Balancer, the router address should be entered in the router field instead of the server address. The router address is the address used in the "server add" command on the Load Balancer machine in step 7.

Server Partitioning: logical servers configured to one physical server (IP address)

With server partitioning, you can further distinguish between particular URLs and their specific applications. For example, one Web server can serve JSP pages, HTML pages, GIF files, database requests, and so on. Load Balancer now provides the ability to partition one cluster and port specific server into several logical servers. This allows you to advise on a particular service on the machine to detect if a servlet engine or a database request is running faster, or not running at all.

Server partitioning allows Load Balancer to detect, for example, that the HTML service is serving pages rapidly, but the database connection has gone down. This allows you to distribute load based on more granular service-specific workload, rather than server-wide weighting alone.

Server partitioning using HTTP or HTTPS advisors

Server partitioning can be useful when used in conjunction with the HTTP and HTTPS advisors. For example, when you have an HTML server that handles HTML, GIF, and JSP pages, if you define (by adding) the server once under port 80, you receive just one load value for the whole HTTP server. This might be misleading because it is possible that the GIF service might not be functioning on the server. Dispatcher still forwards GIF pages to the server, but the client sees a timeout or a failure.

If you define the server three times (for example, ServerHTML, ServerGIF, ServerJSP) under the port and define the server **advisorrequest** parameter with a

different string for each logical server, then you can query the health of the particular service on the server. ServerHTML, ServerGIF and ServerJSP represent three logical servers that have been partitioned from one physical server. For ServerJSP, you can define the advisorrequest string to query the service on the machine that handles JSP pages. For ServerGIF, you can define the advisorrequest string to query the GIF service. And for ServerHTML, you define the advisorrequest to query the HTML service. So, if the client gets no response from the advisorrequest to query the GIF service, Dispatcher will mark that logical server (ServerGIF) as down, while the other two logical servers may be healthy. Dispatcher does not forward any more GIFs to the physical server, but it can still send JSP and HTML requests to the server.

For more information on the **advisorrequest** parameter, see "Configuring the HTTP or HTTPS advisor using the request and response (URL) option" on page 152.

Example for configuring a physical server into logical servers

Within the Dispatcher configuration, you can represent a physical server or a logical server using the *cluster:port:**server*** hierarchy. The server can be a unique IP address of the machine (physical server) in either a symbolic name or IP address format. Or, if you define the server to represent a partitioned server, then you must provide a resolvable server address for the physical server on the **address** parameter of the **dscontrol server add** command. See "dscontrol server — configure servers" on page 330 for more information.

Following is an example of partitioning physical servers into logical servers to handle different types of requests.

```
Cluster: 1.1.1.1
        Port:  80
             Server: A (IP address 1.1.1.2)
                       HTML server
             Server: B (IP address 1.1.1.2)
                       GIF server
             Server: C (IP address 1.1.1.3)
                       HTML server
             Server: D (IP address 1.1.1.3)
                       JSP server
             Server: E (IP address 1.1.1.4)
                       GIF server
             Server: F (IP address 1.1.1.4)
                       JSP server
        Rule1: /*.htm
             Server: A
             Server: C
        Rule2: /*.jsp
             Server: D
             Server: F
        Rule3: /*.gif
             Server: B
             Server: E
```

In this example, server 1.1.1.2 is partitioned into 2 logical servers: "A" (handling HTML requests) and "B" (handling GIF requests). Server 1.1.1.3 is partitioned into 2 logical servers: "C" (handling HTML requests) and "D" (handling JSP requests). Server 1.1.1.4 is partitioned into 2 logical servers: "E" (handling GIF requests) and "F" (handling JSP requests).

High availability

Simple high availability

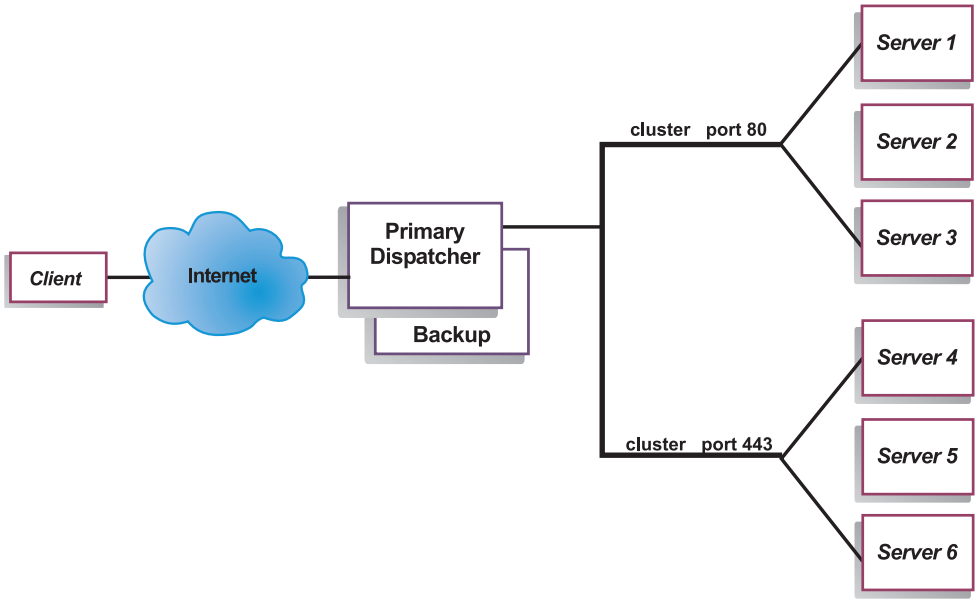


Figure 13. Example of a Dispatcher using simple high availability

The high availability feature involves the use of a second Dispatcher machine. The first Dispatcher machine performs load balancing for all the client traffic as it does in a single Dispatcher configuration. The second Dispatcher machine monitors the "health" of the first, and takes over the task of load balancing if it detects that the first Dispatcher machine has failed.

Each of the two machines is assigned a specific role, either *primary* or *backup*. The primary machine sends connection data to the backup machine on an ongoing basis. While the primary is *active* (load balancing), the backup is in a *standby* state, continually updated and ready to take over, if necessary.

The communication sessions between the two machines are referred to as *heartbeats*. The heartbeats allow each machine to monitor the health of the other.

If the backup machine detects that the active machine has failed, it will take over and begin load balancing. At that point the *statuses* of the two machines are reversed: the backup machine becomes *active* and the primary becomes *standby*.

In the high availability configuration, both primary and backup machines must be on the same subnet with identical configuration.

For information about configuring high availability, see "High availability" on page 164.

Mutual high availability

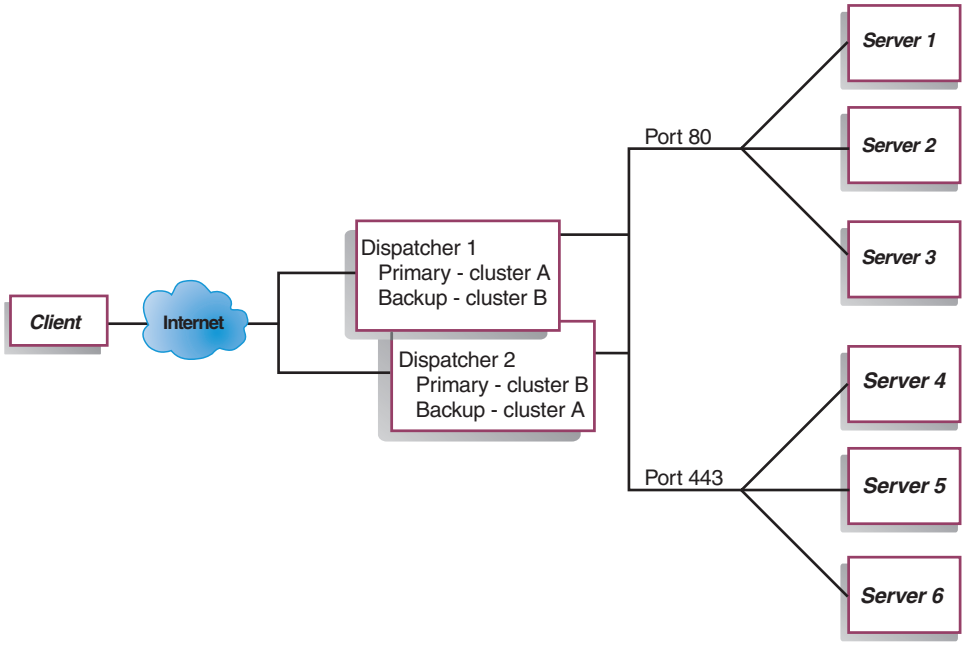


Figure 14. Example of a Dispatcher using mutual high availability

The mutual high availability feature involves the use of two Dispatcher machines. Both machines actively perform load balancing of client traffic, and both machines provide backup for each other. In a simple high availability configuration, only one machine performs load balancing. In a mutual high availability configuration, both machines load balance a portion of the client traffic.

For mutual high availability, client traffic is assigned to each Dispatcher machine on a cluster address basis. Each cluster can be configured with the NFA (nonforwarding address) of its primary Dispatcher. The primary Dispatcher machine normally performs load balancing for that cluster. In the event of a failure, the other machine performs load balancing for both its own cluster and for the failed Dispatcher's cluster.

For an illustration of a mutual high availability configuration with shared “cluster set A" and shared “cluster set B," see Figure 14. Each Dispatcher can actively route packets for its *primary* cluster. If either Dispatcher were to fail and could no longer actively route packets for its primary cluster, then the other Dispatcher could take over routing packets for its *backup* cluster.

Note: Both machines must configure their shared cluster sets the same. That is, the ports used and the servers under each port must be identical in the two configurations.

For information about configuring high availability and mutual high availability, see “High availability” on page 164.

Chapter 7. Configuring Dispatcher

Before following the steps in this chapter, see Chapter 6, "Planning for Dispatcher," on page 37. This chapter explains how to create a basic configuration for the Dispatcher component of Load Balancer.

- See Chapter 20, "Manager, Advisors, and Metric Server functions for Dispatcher, CBR, and Site Selector," on page 141 and Chapter 21, "Advanced features for Dispatcher, CBR, and Site Selector," on page 161 for more complex configurations of Load Balancer.
- See Chapter 23, "Operating and managing Load Balancer," on page 213 for information on remote authenticated administration, Load Balancer logs, and usage of the Load Balancer components.

Note: For previous versions, when the product was known as Network Dispatcher, the Dispatcher control command name was ndcontrol. The Dispatcher control command name is now **dscontrol**.

Overview of configuration tasks

Before you begin the configuration steps in this table, ensure that your Dispatcher machine and all server machines are connected to the network, have valid IP addresses, and are able to ping one another.

Table 1. Configuration tasks for the Dispatcher function

Task	Description	Related information
Set up the Dispatcher machine.	Set up your load balancing configuration.	"Setting up the Dispatcher machine" on page 50
Set up machines to be load-balanced.	Alias the loopback device, check for an extra route, and delete any extra routes.	"Setting up server machines for load balancing" on page 55

Methods of configuration

There are four basic methods of configuring the Dispatcher:

- Command line
- Scripts
- Graphical user interface (GUI)
- Configuration wizard

Command line

This is the most direct means of configuring the Dispatcher. The command parameter values must be entered in English characters. The only exceptions are host names (used in cluster, server, and highavailability commands) and file names (used in file commands).

To start the Dispatcher from the command line:

1. Issue the **dsserver** command from the command prompt. To stop the service, type: **dsserver stop**

Note: For Windows systems, click **Start** > **Control Panel** > **Administrative Tools** > **Services**. Right-click **IBM Dispatcher** and select **Start**. To stop the service, follow the same steps and select **Stop**.

2. Next, issue Dispatcher control commands you want in order to set up your configuration. The procedures in this manual assume use of the command line. The command is **dscontrol**. For more information about commands, see Chapter 26, "Command reference for Dispatcher and CBR," on page 287.

You can use a minimized version of the dscontrol command parameters by typing the unique letters of the parameters. For example, to get help on the file save command, you can type **dscontrol he f** instead of **dscontrol help file**.

To start up the command line interface: issue **dscontrol** to receive an dscontrol command prompt.

To end the command line interface: issue **exit** or **quit**.

Scripts

You can enter commands for configuring Dispatcher into a configuration script file and run them together. See "Sample Load Balancer configuration files" on page 413.

Note: To quickly run the content of a script file (for example, myscript), use either of the following commands:

- To update the current configuration, run the following executable commands from your script file:

  dscontrol file appendload *myscript*

- To completely replace the current configuration, run the following executable commands from your script file:

  dscontrol file newload *myscript*

To save the current configuration into a script file (for example, savescript), run the following command:

dscontrol file save *savescript*

This command will save the configuration script file in the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/configurations/dispatcher**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\configurations\dispatcher**

GUI

For general instructions and an example of the graphical user interface (GUI), see Figure 38 on page 403.

To start the GUI, follow these steps:

1. Ensure dsserver is running
   - For AIX, HP-UX, Linux, or Solaris systems, run the following command as root:

     dsserver

   - For Windows systems, dsserver runs as a service that starts automatically
2. Do one of the following actions, depending on your operating system:

- For AIX, HP-UX, Linux, or Solaris systems: type **lbadmin**
- For Windows systems: click **Start** > **Programs** > **IBM WebSphere** > **Edge Components** > **IBM Load Balancer** > **Load Balancer**

To configure the Dispatcher component from the GUI, you must first select **Dispatcher** in the tree structure. You can start the executor and manager after you connect to a Host. You can also create clusters containing ports and servers, and start advisors for the manager.

The GUI can be used to do anything that you would do with the **dscontrol** command. For example, to define a cluster using the command line, you would enter **dscontrol cluster add** *cluster* command. To define a cluster from the GUI, right-click Executor, then in the pop-up menu, left-click **Add Cluster**. Enter the cluster address in the pop-up window, then click **OK**.

Pre-existing Dispatcher configuration files can be loaded using the **Load New Configuration**(for completely replacing the current configuration) and **Append to Current Configuration** (for updating the current configuration) options presented in the **Host** pop-up menu. You should save your Dispatcher configuration to a file periodically using the **Save Configuration File As** option also presented in the **Host** pop-up menu. The **File** menu located at the top of the GUI will allow you to save your current host connections to a file or restore connections in existing files across all Load Balancer components.

The configuration commands can also be run remotely. For more information, see "Remote Method Invocation (RMI)" on page 213.

In order to run a command from the GUI: highlight the Host node from the GUI tree and select **Send command....** from the Host pop-up menu. In the command entry field, type the command that you want to run, for example: **executor report**. The results and history of the commands run in the current session and appear in the window provided.

You can access **Help** by clicking the question mark icon in the upper right corner of the Load Balancer window.

- **Help: Field level —** describes each field, default values
- **Help: How do I —** lists tasks that can be done from that screen
- **InfoCenter —** provides centralized access to product information

For more information about using the GUI, see Appendix A, "GUI: General instructions," on page 403.

Configuring with the configuration wizard

If you are using the configuration wizard, follow these steps:

1. Start the dsserver on Dispatcher:
   - For AIX, HP-UX, Linux, or Solaris systems, run the following as root user: **dsserver**
   - For Windows systems, dsserver runs as a service that starts automatically.
2. Start the wizard function of Dispatcher, **dswizard.**

The wizard guides you step by step through the process of creating a basic configuration for the Dispatcher component. You will be asked questions about

your network. You will be guided through the setup of a cluster for Dispatcher to load balance traffic between a group of servers.

Setting up the Dispatcher machine

Before setting up the Dispatcher machine, you must be the root user (for AIX, HP-UX, Linux, or Solaris systems) or the Administrator on Windows systems.

On all supported platforms, the Load Balancer can have a **collocated** server. Collocation means that Load Balancer can physically reside on a server machine which it is load balancing.

For the Dispatcher machine, when using the mac forwarding method, you will need at least two valid IP addresses. For cbr or nat forwarding method, you will need at least three valid IP addresses:

- An IP address specifically for the Dispatcher machine

  This IP address is the primary IP address of the Dispatcher machine and is called the nonforwarding address (NFA). This is by default the same address as that returned by the **hostname** command. Use this address to connect to the machine for administrative purposes, such as doing remote configuration using Telnet or accessing the SNMP subagent. If the Dispatcher machine can already ping other machines on the network, you do not need to do anything further to set up the nonforwarding address.
- One IP address for each cluster

  A cluster address is an address that is associated with a host name (such as www.yourcompany.com). This IP address is used by a client to connect to the servers in a cluster. This is the address that is load balanced by the Dispatcher.
- For cbr or nat forwarding, an IP address for the return address

  Dispatcher uses the return address as its source address when load balancing the client's request to the server. This ensures that the server returns the packet to the Dispatcher machine rather than sending the packet directly to the client. (Dispatcher will then forward the IP packet to the client.) You must specify the return address value when adding the server. You cannot modify the return address unless you remove the server and then add it again.

  The number of connections that Load Balancer can keep active with the backend server is limited by the number of return addresses that are defined. Load Balancer uses ports that are based upon the return address only; not the return address and server combination. When all the available ports are in use, additional connections fail. In a busy environment, use multiple return addresses to prevent a shortage of available ports.

Solaris systems only:

- By default, Dispatcher is configured to load balance traffic on 100Mbps Ethernet network interface cards. The default 100Mbps Ethernet adapter is specified in the ibmlb.conf file as `eri`. However, support is also provided for other types of interface cards, which include: le, ce, ge, hme, eri, bge, vge, qfe, dfme, fjgi and fjge.

  For example, to change the default setting, edit the **/opt/ibm/edge/lb/servers/ibmlb.conf** file as follows:
  - To use a 10 Mbps Ethernet adapter, replace `eri` with `le`.
  - To use a 1Gbps Ethernet adapter, replace `eri` with `ge`.
  - To use a multi-port adapter, replace `eri` with `qfe`.

To support multiple types of adapters, replicate the line in the ibmlb.conf file and modify each line to match your device type.

For example, if you plan to use two 100Mbps Ethernet adapters, you need a single line in the ibmlb.conf file specifying the `eri` device.

If you plan to use one 10Mbps Ethernet adapter and one 100Mbps Ethernet adapter, you need to specify two lines in the ibmlb.conf file: one line specifying the `le` device and one line specifying the `eri` device.

Note: The **ibmlb.conf** file provides input to the Solaris **autopush** command and must be compatible with the autopush command.

- To determine the type of Ethernet network interface in use on your machine, issue the following command from the Solaris command prompt:

  ```
  ifconfig -a
  ```

  If the following output results:

  ```
  lo0: flags=2001000849<UP,LOOPBACK,RUNNING,MULTICAST,IPv4,VIRTUAL>
       mtu 8232 index 1 inet 127.0.0.1 netmask ff000000
  eri0: flags=1000843<UP,BROADCAST,RUNNING,MULTICAST,IPv4>
        mtu 1500 index 2 inet 9.42.93.208
        netmask fffffc00 broadcast 9.42.95.255 ether 0:3:ba:2d:24:45
  ```

  Then you would edit the ibmlb.conf file as follows:

  ```
  eri -1 0 ibmlb
  ```

- Starting or stopping the Dispatcher executor will unconfigure all aliases on the adapters listed in the ibmlb.conf file. To automatically reconfigure aliases on those adapters (except those for use by the Dispatcher component of Load Balancer) use the **goAliases** script file. A sample script is located in the following directory:
  - AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/samples**
  - Windows operating systems: **<***install_root***>ibm\edge\lb\servers\samples**

  This script *must* be moved to the following directory before it will run:
  - AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/bin**
  - Windows operating systems: **<***install_root***>ibm\edge\lb\servers\bin**

  The goAliases script is automatically run when the Dispatcher executor starts or stops.

  For example, if clusters X and Y are configured for use by the CBR component on any of the adapters listed in ibmlb.conf, clusters X and Y are unconfigured when the **dscontrol executor start** or **dscontrol executor stop** commands are issued. This may not be the desired result. When clusters X and Y are configured in the goAliases script, the clusters are automatically reconfigured after the Dispatcher executor starts or stops.

Ensure that IP forwarding is not enabled for the TCP/IP protocol.

Figure 15 on page 52 shows an example of Dispatcher set up with a single cluster, two ports, and three servers.

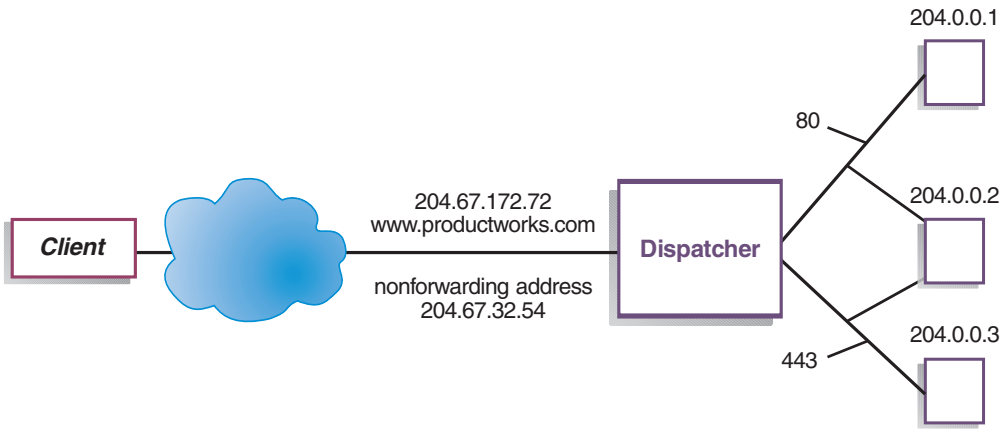


Figure 15. Example of the IP addresses needed for the Dispatcher machine

For help with commands used in this procedure, see Chapter 26, "Command reference for Dispatcher and CBR," on page 287.

For a sample configuration file, see "Sample Load Balancer configuration files" on page 413.

Step 1. Start the server function

AIX, HP-UX, Linux, or Solaris systems: To start the server function, type **dsserver**.

Windows systems: The server function starts automatically as a service.

Note: A default configuration file (default.cfg) gets automatically loaded when starting dsserver. If the user decides to save the Dispatcher configuration in default.cfg, then everything saved in this file is automatically loaded next time dsserver gets started.

Step 2. Start the executor function

To start the executor function, enter the **dscontrol executor start** command. You may also change various executor settings at this time. See Chapter 26, "Command reference for Dispatcher and CBR," on page 287.

Step 3. Define the nonforwarding address (if different from hostname)

The nonforwarding address is used to connect to the machine for administrative purposes, such as using Telnet or SMTP to this machine. By default, this address is the hostname.

To define the nonforwarding address, enter the **dscontrol executor set nfa** *IP_address* command or edit the sample configuration file. *IP_address* is either the symbolic name or the IP address.

Step 4. Define a cluster and set cluster options

Dispatcher will balance the requests sent to the cluster address to the servers configured on the ports for that cluster.

The cluster is either the symbolic name, the dotted decimal address, or the special address 0.0.0.0 that defines a wildcard cluster. To define a cluster, issue the

command **dscontrol cluster add**. To set cluster options, issue the command **dscontrol cluster set** or you can use the GUI to issue commands. Wildcard clusters can be used to match multiple IP addresses for incoming packets to be load balanced. See "Use wildcard cluster to combine server configurations" on page 191, "Use wildcard cluster to load balance firewalls" on page 191, and "Use wildcard cluster with Caching Proxy for transparent proxy" on page 192 for more information.

Step 5. Alias the network interface card

When the cluster has been defined, you normally must configure the cluster address on one of the network interface cards of the Dispatcher machine. To do this, issue the command **dscontrol executor configure** *cluster_address***.** This will look for an adapter with an existing address that belongs on the same subnet as the cluster address. It will then issue the operating system's adapter configuration command for the cluster address, using the adapter found and the netmask for the existing address found on that adapter. For example:

```
dscontrol executor configure 204.67.172.72
```

Circumstances where you do not want to configure the cluster address include clusters added to a standby server in high-availability mode, or clusters added to a wide-area dispatcher acting as a remote server. You also do not need to run the executor configure command if, in stand-alone mode, you use the sample **goIdle** script. For information on the goIdle script, see "Using scripts" on page 167.

In rare cases you might have a cluster address that does not match any subnet for existing addresses. In this case, use the second form of the executor configure command and explicitly provide the interface name and netmask. Use **dscontrol executor configure** *cluster_address interface_name netmask*.

Some examples include:

```
dscontrol executor configure 204.67.172.72 en0 255.255.0.0
(AIX systems)
dscontrol executor configure 204.67.172.72 eth0:1 255.255.0.0
(Linux systems)
dscontrol executor configure 204.67.172.72 eri0 255.255.0.0
(Solaris systems)
dscontrol executor configure 204.67.172.72 en1 255.255.0.0
(Windows systems)
```

Windows systems

To use the second form of the `executor configure` command on Windows systems, you must determine the interface name to use. If you have only one Ethernet card in your machine, the interface name is en0. If you have only one Token Ring card, the interface name is tr0. If you have multiple cards of either type, you will need to determine the mapping of the cards. Use the following steps:

1. From the command line start the executor: `dscontrol executor start`
2. Run the command: `dscontrol executor xm 1`

Output will be displayed to the screen. To determine the interface name to use for your Load Balancer configuration, look for the IP address of your Load Balancer Machine in the lines that follow `Number of NIC records`.

The IP addresss of your Load Balancer machine will be listed as: `ia->ia_addr`. The associated interface name will be listed as: `ifp->if_name`.

The interface names assigned by the `executor configure` command map to the interface names listed in this command.

After you obtain this mapping information, you can create an alias on the network interface to the cluster address.

Using ifconfig commands to configure cluster aliases

On Linux or UNIX systems, the executor configure command runs ifconfig commands.

Solaris and HP-UX systems: When using bind-specific server applications that bind to a list of IP addresses that do not contain the server's IP, use **arp publish** command instead of ifconfig to dynamically set an IP address on the Load Balancer machine. For example:

```
arp -s <cluster> <Load Balancer MAC address> pub
```

Step 6. Define ports and set port options

To define a port, enter the **dscontrol port add** *cluster*:*port* command, edit the sample configuration file, or use the GUI. *Cluster* is either the symbolic name or the IP address. *Port* is the number of the port you are using for that protocol. You may also change various port settings at this time. You must define and configure all servers for a port. See Chapter 26, “Command reference for Dispatcher and CBR,” on page 287.

Port number 0 (zero) is used to specify a wildcard port. This port will accept traffic for a port that is not destined for any of the defined ports on the cluster. The wildcard port is used to configure rules and servers for any port. This function could also be used if you have an identical server and rule configuration for multiple ports. The traffic on one port could then affect the load-balancing decisions for traffic on other ports. See “Use wildcard port to direct unconfigured port traffic” on page 192 for more information about when you might want to use a wildcard port.

Step 7. Define load-balanced server machines

To define a load-balanced server machine, enter the **dscontrol server add** *cluster*:*port*:*server* command, edit the sample configuration file, or use the GUI. *Cluster* and *server* are either the symbolic name or the IP address. *Port* is the number of the port you are using for that protocol. You must define more than one server to a port on a cluster in order to perform load balancing.

Bind-specific servers: If the Dispatcher component is load balancing to bind-specific servers, then the servers *must* be configured to bind to the cluster address. Because the Dispatcher forwards packets without changing the destination IP address, when the packets reach the server, the packets will still contain the cluster address as the destination. If a server has been configured to bind to an IP address other than the cluster address, then the server will be unable to accept requests destined for the cluster.

To determine if the server is bind specific, issue the `netstat -an` command and look for the server:port. If the server is not bind specific, the result from this command will be 0.0.0.0:80. If the server is bind specific, you will see an address such as 192.168.15.103:80.

Note: For Solaris and Linux systems: When using advisors, bind-specific servers must not be collocated.

Multiple address collocation: In a collocated configuration, the address of the collocated server machine does *not* have to be identical to the nonforwarding address (NFA). You can use another address if your machine has been defined with multiple IP addresses. For the Dispatcher component, the collocated server machine must be defined as **collocated** using the **dscontrol server** command. For more information on collocated servers, see "Using collocated servers" on page 162.

For more information on dscontrol server command syntax, see "dscontrol server — configure servers" on page 330.

Step 8. Start the manager function (optional)

The manager function improves load balancing. To start the manager, enter the **dscontrol manager start** command, edit the sample configuration file, or use the GUI.

Step 9. Start the advisor function (optional)

The advisors give the manager more information about the ability of the load-balanced server machines to respond to requests. An advisor is specific to a protocol. For example, to start the HTTP advisor, issue the following command:

```
dscontrol advisor start http port
```

For a list of advisors along with their default ports, see Chapter 26, "Command reference for Dispatcher and CBR," on page 287. For a description of each advisor, see "List of advisors" on page 149.

Step 10. Set cluster proportions as required

If you start advisors, you may modify the proportion of importance given to advisor information being included in the load balancing decisions. To set the cluster proportions, issue the **dscontrol cluster set** *cluster* **proportions** command. For more information, see "Proportion of importance given to status information" on page 142.

Setting up server machines for load balancing

Perform the following steps if one of these conditions is true:

- If you are using mac forwarding method and this is a backend server machine.
- If you are using mac forwarding method and this is a collocated server that is configured as the high availability standby machine.

  Notes:

  1. Procedures to delete the aliasing of the loopback will need to be put in the go* scripts in case the machine changes to active.
  2. If this is configured as the high availability active machine, procedures to alias the loopback device will need to be put in the go* scripts in case the machine changes to standby.

When using mac forwarding method, Dispatcher will only load balance across servers that allow the loopback adapter to be configured with an additional IP address, for which the backend server will never respond to ARP (address resolution protocol) requests. Follow the steps in this section to set up the load-balanced server machines.

Step 1. Alias the loopback device

For the load-balanced server machines to work, you must set (or preferably alias) the loopback device (often called lo0) to the cluster address. When using the mac forwarding method, the Dispatcher component does not change the destination IP address in the TCP/IP packet before forwarding the packet to a TCP server machine. By setting or aliasing the loopback device to the cluster address, the load balanced server machines will accept a packet that was addressed to the cluster address.

If you have an operating system that supports network interface aliasing (such as AIX, HP-UX, Linux, Solaris, or Windows systems), you should alias the loopback device to the cluster address. The benefit of using an operating system that supports aliases is that you have the ability to configure the load-balanced server machines to serve multiple cluster addresses.

IMPORTANT: For Linux systems, see "Linux loopback aliasing alternatives when using Load Balancer's mac forwarding" on page 60.

If you have a server with an operating system that does not support aliases you must set the loopback device to the cluster address.

Use the command for your operating system as shown in Table 2 to set or alias the loopback device.

Table 2. Commands to alias the loopback device (lo0) for Dispatcher

AIX	**AIX 4.3 or earlier:**
	`ifconfig lo0 alias cluster_address netmask netmask`
	
	Note: Use the netmask of the primary adapter.
	AIX 5.x or later:
	`ifconfig lo0 alias cluster_address netmask 255.255.255.255`

HP-UX	`ifconfig lo0:1 cluster_address up`
	
	Note: If you use bind-specific server applications that bind to a list of IP addresses that do not contain the server's IP, use the arp publish command instead of ifconfig to dynamically set an IP address on the Load Balancer machine.
	
	For example:
	`arp -s cluster_address Load Balancer's_MAC_address pub`

Linux	Choose one of the following commands:
	• Use the ip command:
	`ip -4 addr add cluster_address/32 dev lo`
	• Use the ifconfig command:
	`ifconfig lo:1 cluster_address netmask 255.255.255.255 up`
	IMPORTANT: Once you issue one of the configuration commands on your machine, consistently use the same configuration command (**ip** or **ifconfig**), or results can occur that are not expected.

OS/2	**ifconfig lo** *cluster_address*

Table 2. Commands to alias the loopback device (lo0) for Dispatcher (continued)

OS/390®	Configuring a loopback alias on OS/390 system
	• In the IP parameter member (file), an Administrator will need to create an entry in the Home address list. For example
	`HOME`
	`;Address Link`
	`192.168.252.11 tr0`
	`192.168.100.100 ltr1`
	`192.168.252.12 loopback`
	• Several addresses can be defined for the loopback.
	• The loopback address of 127.0.0.1 is configured by default.

Solaris	**Solaris 7**
	`ifconfig lo0:1 cluster_address 127.0.0.1 up`
	Solaris 8 and later
	`ifconfig lo0:1 plumb cluster_address netmask netmask up`

Windows NT	1. Click **Start**, then click **Settings**. 2. Click **Control Panel**, then double-click **Network**. 3. If you have not done so already, add the MS Loopback Adapter Driver. a. In the Network window, click **Adapters**. b. Select **MS Loopback Adapter**, then click **OK**. c. When prompted, insert your installation CD or disks. d. In the Network window, click **Protocols**. e. Select **TCP/IP Protocol**, then click **Properties**. f. Select **MS Loopback Adapter**, then click **OK**. 4. Set the loopback address to your cluster address. Accept the default subnet mask (255.0.0.0), and do not enter a gateway address. **Note:** You may have to exit and reenter Network Settings before the MS Loopback Driver shows up under TCP/IP Configuration.

Table 2. Commands to alias the loopback device (lo0) for Dispatcher (continued)

Windows, other versions (including Windows Server 2003 and later)	1. Use the ipconfig command to determine the interface name for the loopback device:
	`ipconfig /all`
	
	This command locates the connection with a description of the Microsoft Loopback Adapter. The following example is the output from the ipconfig /all command, where the Microsoft Loopback Adapter is Ethernet adapter Local Area Connection 2, so the connection is Local Area Connection 2:
	`Windows IP Configuration`
	
	`Host Name . . . . . : ndserv10`
	`Primary Dns Suffix . . . . . : rtp.somewhere.ibm.com`
	`Node Type . . . . . : Unknown`
	`IP Routing Enabled. . . . . : No`
	`WINS Proxy Enabled. . . . . : No`
	`DNS Suffix Search List. . . . . : rtp.somewhere.ibm.com`
	
	`Ethernet adapter Local Area Connection 2:`
	
	`Connection-specific DNS Suffix . :`
	`Description . . . . . : Microsoft Loopback Adapter`
	`Physical Address. . . . . : 02-00-4C-4F-4F-50`
	`DHCP Enabled. . . . . : No`
	`IP Address. . . . . : 9.42.92.158`
	`Subnet Mask . . . . . : 255.255.252.0`
	`IP Address. . . . . : 9.42.92.159`
	`Subnet Mask . . . . . : 255.255.252.0`
	`IP Address. . . . . : 9.42.92.160`
	`IP Address. . . . . : 9.42.92.159`
	`Default Gateway . . . . . :`
	`DNS Servers . . . . . : 127.0.0.1`
	2. Add the cluster address to the loopback using the netsh command. For example:
	`netsh interface ipv4 add address "Local Area Connection 2" 9.42.92.161`
	
	Note: If you are using a high-availability configuration, and the machine is running as the primary machine, do not alias to the loopback device, because this scenario prevents traffic to the cluster address from being routed by the Load Balancer machine.
	3. Issue the following ipconfig command again, and you should see the address added on the loopback adapter. For example, issue the following command:
	`ipconfig /all`
	
	You should see output that is similar to the following:
	`Ethernet adapter Local Area Connection 2:`
	
	`Connection-specific DNS Suffix . :`
	`Description . . . . . : Microsoft Loopback Adapter`
	`Physical Address. . . . . : 02-00-4C-4F-4F-50`
	`DHCP Enabled. . . . . : No`
	`IP Address. . . . . : 9.42.92.158`
	`Subnet Mask . . . . . : 255.255.252.0`
	`IP Address. . . . . : 9.42.92.159`
	`Subnet Mask . . . . . : 255.255.252.0`
	`IP Address. . . . . : 9.42.92.161`
	`IP Address. . . . . : 9.42.92.160`
	`IP Address. . . . . : 9.42.92.159`
	`Default Gateway . . . . . :`
	`DNS Servers . . . . . : 127.0.0.1`
	4. Enable forwarding for all the interfaces in the machine using the following command:
	`netsh interface ipv4 show interface`

Step 2. Check for an extra route

On some operating systems, a default route may have been created and needs to be removed.

- Check for an extra route on Windows operating systems with the following command:

  ```
  route print
  ```

  IMPORTANT: Any extra routes should be ignored on Windows 2003. If problems are encountered with routing after aliasing, remove the alias and add it back using a different netmask.

- Check for an extra route on all AIX, HP-UX, Linux, and Solaris operating systems with the following command:

  ```
  netstat -nr
  ```

Windows Example:

1. After **route print** is entered, a table similar to the following example will be displayed. (This example shows finding and removing an extra route to cluster 9.67.133.158 with a default netmask of 255.0.0.0.)

   ```
   Active Routes:

   Network Address Netmask         Gateway Address  Interface      Metric
   0.0.0.0         0.0.0.0         9.67.128.1       9.67.133.67     1
   9.0.0.0         255.0.0.0       9.67.133.158     9.67.133.158    1
   9.67.128.0      255.255.248.0   9.67.133.67      9.67.133.67     1
   9.67.133.67     255.255.255.255 127.0.0.1        127.0.0.1       1
   9.67.133.158    255.255.255.255 127.0.0.1        127.0.0.1       1
   9.255.255.255   255.255.255.255 9.67.133.67      9.67.133.67     1
   127.0.0.0       255.0.0.0       127.0.0.1        127.0.0.1       1
   224.0.0.0       224.0.0.0       9.67.133.158     9.67.133.158    1
   224.0.0.0       224.0.0.0       9.67.133.67      9.67.133.67     1
   255.255.255.255 255.255.255.255 9.67.133.67      9.67.133.67     1
   ```

2. Find your cluster address under the “Gateway Address” column. If you have an extra route, the cluster address will appear twice. In the example given, the cluster address (9.67.133.158) appears in row 2 and row 8.
3. Find the network address in each row in which the cluster address appears. You need one of these routes and will need to delete the other route, which is extraneous. The extra route to be deleted is the one whose network address begins with the first digit of the cluster address, followed by three zeroes. In the example shown, the extra route is the one in row two, which has a network address of **9.0.0.0**:

   ```
   9.0.0.0    255.0.0.0   9.67.133.158  9.67.133.158     1
   ```

Step 3. Delete any extra route

You must delete the extra route. Use the command for your operating system shown in Table 3 to delete the extra route.

Example: To delete the extra route as shown in the "Active Routes" example table for Step 2, enter:

```
route delete 9.0.0.0 9.67.133.158
```

Table 3. Commands to delete any extra route for Dispatcher

HP-UX	**route delete** *cluster_address cluster_address*

Table 3. Commands to delete any extra route for Dispatcher (continued)

Windows	**route delete** *network_address cluster_address* (at an MS-DOS prompt) **Note:** You must delete the extra route every time you reboot the server. On Windows 2003, it is not possible to delete routes. Any extra routes should be ignored on Windows 2003. If problems are encountered with routing after aliasing, remove the alias and add it back using a different netmask.

Using the example shown in Figure 15 on page 52, and setting up a server machine that is running and AIX system, the command would be:

```
route delete -net  204.0.0.0  204.67.172.72
```

Step 4. Verify server is properly configured

To verify if a backend server is properly configured, perform the following steps from a different machine on the same subnet when the Load Balancer is not running and *cluster* is unconfigured:

1. Issue the command:

   ```
   arp -d cluster
   ```

2. Issue the command:

   ```
   ping cluster
   ```

   There should be no response. If there is a response to the ping, ensure that you did not ifconfig the cluster address to the interface. Ensure that no machine has a published arp entry to the cluster address.

3. Ping the backend server, then immediately issue the command:

   ```
   arp -a
   ```

   In the output from the command, you should see the MAC address of your server. Issue the command:

   ```
   arp -s cluster server_mac_address
   ```

4. Ping the cluster. You should get a response. Issue a http, telnet, or other request that is addressed to the cluster that you expect your backend server to handle. Ensure that it works properly.
5. Issue the command:

   ```
   arp -d cluster
   ```

6. Ping the cluster. There should be no response.

   Note: If there is a response, issue an **arp** *cluster* instruction to get the MAC address of the misconfigured machine. Then, repeat steps 1 through 6.

Linux loopback aliasing alternatives when using Load Balancer's mac forwarding

Some versions of Linux systems issue ARP responses for any IP address configured on the machine on any interface present on the machine. It may also choose an ARP source IP address for ARP `who-has` queries based on all IP addresses present on the machine, regardless of the interfaces on which those addresses are configured. This causes all cluster traffic to be directed to a single server in an indeterminate manner.

When using Dispatcher's mac forwarding method, a mechanism must be employed to ensure that cluster-addressed traffic can be accepted by the stacks of the backend servers, including the collocated high availability standby machine, when both high availability and collocation are in use.

In most cases, you must alias the cluster address on the loopback; therefore, backend servers must have the cluster aliased on the loopback, and if you use high availability and collocation, the standby load-balancing servers must have clusters aliased on the loopback.

To ensure that Linux systems do not advertise addresses on the loopback, you can use any one of the following four solutions to make Linux systems compatible with Dispatcher's mac forwarding.

1. Use a kernel that does not advertise the addresses. This is the preferred option, as it does not incur a per-packet overhead and it does not require per-kernel reconfiguration.
   - United Linux 1 / SLES8 with SP2(x86) or SP3 (all other architectures) and higher contains the Julian ARP hidden patch. Ensure that it is always in effect before aliasing the cluster address with the command:

     ```
     # sysctl -w  net.ipv4.conf.all.hidden=1 net.ipv4.conf.lo.hidden=1
     ```

     Clusters can then be aliased in the normal way, such as:

     ```
     # ifconfig lo:1 $CLUSTER_ADDRESS netmask 255.255.255.255 up
     ```
   - Use the arp_ignore sysctl available in 2.4.25 and 2.6.5 and higher, but note that distributions sometimes backport features. Ensure that it is enabled before aliasing the cluster addresses with the commands:

     ```
     # sysctl -w   net.ipv4.conf.all.arp_ignore=3
     net.ipv4.conf.all.arp_announce=2
     ```

     Clusters must then be aliased with the following command:

     ```
     # ip addr add $CLUSTER_ADDRESS/32 scope host dev lo
     ```

     A similar command must be in the go* scripts in high availability collocation configurations.
   - Note: When using `sysctl`, ensure that these settings survive reboot by adding the settings to `/etc/sysctl.conf`.
2. Use IP tables to redirect all incoming cluster traffic to the localhost. If you use this method, do not configure the loopback adapter with an alias. Instead, use the command:

   ```
   # iptables -t nat -A PREROUTING -d $CLUSTER_ADDRESS -j REDIRECT
   ```

   This command causes Linux systems to do destination NAT on each packet, converting the cluster address to the interface address. This method has about a 6.4% connections-per-second throughput penalty. This method works on any supported stock distribution; no kernel module or kernel patch+build+install is needed.
3. Apply the noarp module version 1.2.0 or higher. The kernel source must be available and properly configured, and development tools (gcc, gnu make, and so forth) must be available. You must build and install the module every time the kernel is upgraded. It is available at http://www.masarlabs.com/noarp/. Because the kernel code itself is not modified, it is much less intrusive than solution #4 (listed below), and is much less prone to error. It also must be configured before any cluster address is aliased on the loopback. For example:

```
# modprobe noarp
# noarpctl add $CLUSTER_ADDRESS nic-primary-addr
```

where *nic-primary-addr* is an address in the same subnet as the cluster address. Clusters can then be aliased in the normal way, such as:

```
# ifconfig lo:1 cluster address netmask 255.255.255.255 up
```

Note: For high availability collocation configurations, `noarpctl adds` and `dels` must be placed in the go* scripts. This ensures that the active Load Balancer can use ARP for the cluster address and that the standby Load Balancer, which is acting as a server, does not accidentally (that is, indeterminately) begin to receive all cluster traffic.

4. Obtain the Julian patch from the following Web site: *http://www.ssi.bg/~ja/#hidden*. Follow your distribution instructions for patching and compiling a kernel suitable for use with that distribution. If this is a collocated high availability Load Balancer, ensure that the `uname -r` matches the distribution-supplied kernel, and ensure that you start with the distribution kernel `.config` file. After you build, install, and run your kernel with the Julian hidden patch, following the instructions under the first solution listed for enabling the patch.

   Note: Distribution support implications might exist for running a custom kernel.

Part 3. Content Based Routing (CBR) component

This part provides information on a quick start configuration, planning considerations, and describes the methods of configuring Load Balancer's CBR component. It contains the following chapters:

Chapter 8. Quick start configuration

This quick start example shows how to configure three locally attached workstations using CBR along with Caching Proxy to load-balance Web traffic between two Web servers. (For simplicity, this example illustrates servers on the same LAN segment, however with CBR there is no restriction for using servers on the same LAN.)

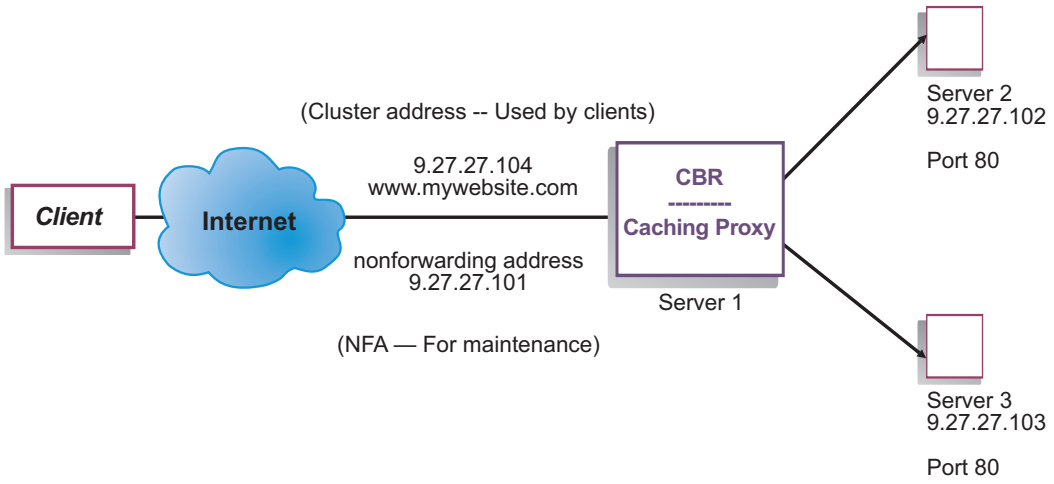


Figure 16. A simple local CBR configuration

What do you need?

For the quick start example, you will need three workstations and four IP addresses. One workstation is used as the CBR machine; the other two workstations are used as Web servers. Each Web server requires one IP address. The CBR workstation requires one actual address, and one address to be load balanced.

Note: The Content Based Routing (CBR) component is not available on platforms that run a 64-bit JVM, except for HP-UX ia64. On HP-UX ia64, the CBR component runs as a 32-bit application. You can use the CBR forwarding method of Load Balancer's Dispatcher component to provide content-based routing without the use of Caching Proxy. See "Dispatcher's content-based routing (cbr forwarding method)" on page 41 for more information.

To use CBR, Caching Proxy must be installed on the same server. To configure Caching Proxy for CBR, see "Step 1. Configure Caching Proxy to use CBR" on page 79.

How do you prepare?

1. For this example, set up your workstations on the same LAN segment. Ensure that network traffic between the three machines does not have to pass through any routers or bridges.
2. Configure the network adapters of the three workstations. For this example, we will assume you have the following network configuration:

Workstation	Name	IP Address
1	server1.mywebsite.com	9.27.27.101
2	server2.mywebsite.com	9.27.27.102
3	server3.mywebsite.com	9.27.27.103
Netmask = 255.255.255.0		

Each of the workstations contains only one standard Ethernet network interface card.

3. Ensure that server1.mywebsite.com can ping both server2.mywebsite.com and server3.mywebsite.com.
4. Ensure that server2.mywebsite.com and server3.mywebsite.com can ping server1.mywebsite.com.
5. Ensure that Web servers on server2.mywebsite.com and server3.mywebsite.com are operational. Use a Web browser to request pages directly from **http://server2.mywebsite.com** (for example, .../member/index.html) and **http://server3.mywebsite.com** (for example, .../guest/index.html).
6. Obtain another valid IP address for this LAN segment. This is the cluster address you will provide to clients who wish to access your site. For this example we will use:

```
Name= www.mywebsite.com
IP=9.27.27.104
```

Configuring the CBR component

With CBR, you can create a configuration by using the command line, the configuration wizard, or the graphical user interface (GUI). For this quick start example, configuration steps are demonstrated using the command line.

Note: The parameter values must be typed in English characters. The only exceptions are parameter values for host names and file names.

Configuring with the command line

From a command prompt, follow these steps:

1. Start the cbrserver. Run the following command as root user or Administrator:
   cbrserver

   Note: For Windows platform: Start cbrserver (Content Based Routing) from the Services panel: **Start** > **Control Panel** > **Administrative Tools** > **Services**.
2. Start the executor function of CBR:

   cbrcontrol executor start
3. Start Caching Proxy. (Caching Proxy can be started any time after you start the executor function):

   ibmproxy

   Note: For Windows platform: You can also start Caching Proxy from the Services panel: **Start** > **Control Panel** > **Administrative Tools** > **Services**.
4. Add the cluster (the host name, Web site, to which clients connect) to the CBR configuration:

   cbrcontrol cluster add www.mywebsite.com

5. Add the cluster address (9.27.27.104) for the Web site to the network interface card on the CBR machine. See"Step 5. Alias the network interface card (optional)" on page 81 for more information.
6. Add the http protocol port to the CBR configuration:

   cbrcontrol port add www.mywebsite.com:80
7. Add each of the Web servers to the CBR configuration:

   cbrcontrol server add www.mywebsite.com:80:server2.mywebsite.com

   cbrcontrol server add www.mywebsite.com:80:server3.mywebsite.com
8. Add content rules to your CBR configuration. (A content rule defines how a URL request will be distinguished and sent to one of the servers or server sets):

   cbrcontrol rule add www.mywebsite.com:80:memberRule type content pattern uri=*/member/*

   cbrcontrol rule add www.mywebsite.com:80:guestRule type content pattern uri=*/guest/*

   In this example, using the content rule, client requests to Web site www.mywebsite.com are sent to a different server based on a directory in their URI request path. See Appendix B, "Content rule (pattern) syntax," on page 409 for more information.
9. Add servers to your rules:

   cbrcontrol rule useserver www.mywebsite:80:memberRule server2.mywebsite.com

   cbrcontrol rule useserver www.mywebsite:80:guestRule server3.mywebsite.com

   CBR will now do load balancing based on content-based rule. A client with a URL request containing **/member/** will be directed to server2.mywebsite.com. A client with a URL request containing **/guest/** will be directed to server3.mywebsite.com.
10. Start the manager function of CBR:

    cbrcontrol manager start
11. Start the advisor function of CBR:

    cbrcontrol advisor start http 80

    CBR will now make sure that client requests are not sent to a failed Web server.

Your basic configuration with locally attached servers is now complete.

Testing your configuration

Test to see if the configuration is working:

1. From a Web browser, go to location **http://www.mywebsite.com/member/index.htm**. If a page is displayed, the configuration is working.
2. Reload the page in the Web browser.
3. Look at the results of the following command:

   ```
   cbrcontrol server report www.mywebsite.com:80:
   ```

   The total connections column of the two servers should add up to "2."

Configuring with the graphical user interface (GUI)

For information on using the CBR GUI, see "GUI" on page 77 and see Appendix A, "GUI: General instructions," on page 403.

Configuring with the configuration wizard

For information on using the CBR wizard, see "Configuration wizard" on page 78.

Types of cluster, port, server configurations

There are many ways that you can configure CBR to support your site. If you have only one host name for your site to which all of your customers will connect, you can define a single cluster of servers. For each of these servers, you configure a port through which CBR communicates. See Figure 9 on page 34.

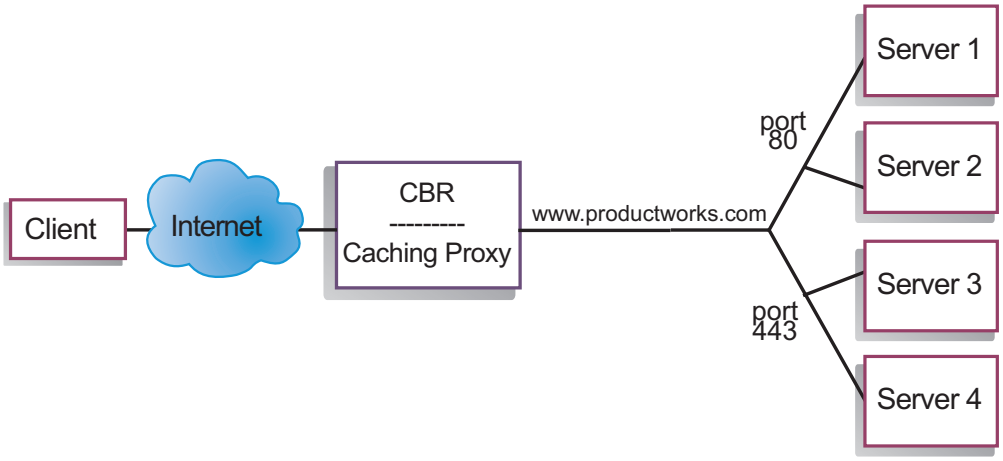


Figure 17. Example of CBR configured with a single cluster and 2 ports

In this example for the CBR component, one cluster is defined at www.productworks.com. This cluster has two ports: port 80 for HTTP and port 443 for SSL. A client making a request to http://www.productworks.com (port 80) would go to a different server than a client requesting https://www.productworks.com (port 443).

Another way of configuring CBR would be appropriate if you have a very large site with many servers dedicated to each protocol supported. In this case, you might want to define a cluster for each protocol with a single port but with many servers, as shown in Figure 10 on page 35.

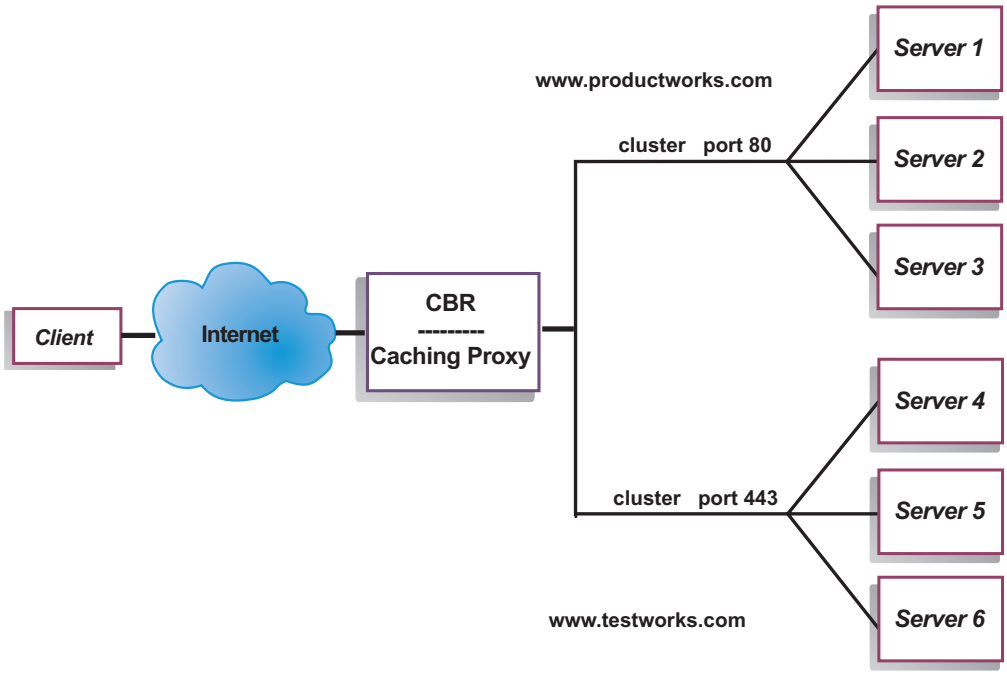


Figure 18. Example of CBR configured with two clusters, each with one port

In this example for the CBR component, two clusters are defined: www.productworks.com for port 80 (HTTP) and www.testworks.com for port 443 (SSL).

A third way of configuring CBR would be necessary if your site does content hosting for several companies or departments, each one coming into your site with a different URL. In this case, you might want to define a cluster for each company or department and then define any ports to which you want to receive connections at that URL, as shown in Figure 11 on page 36.

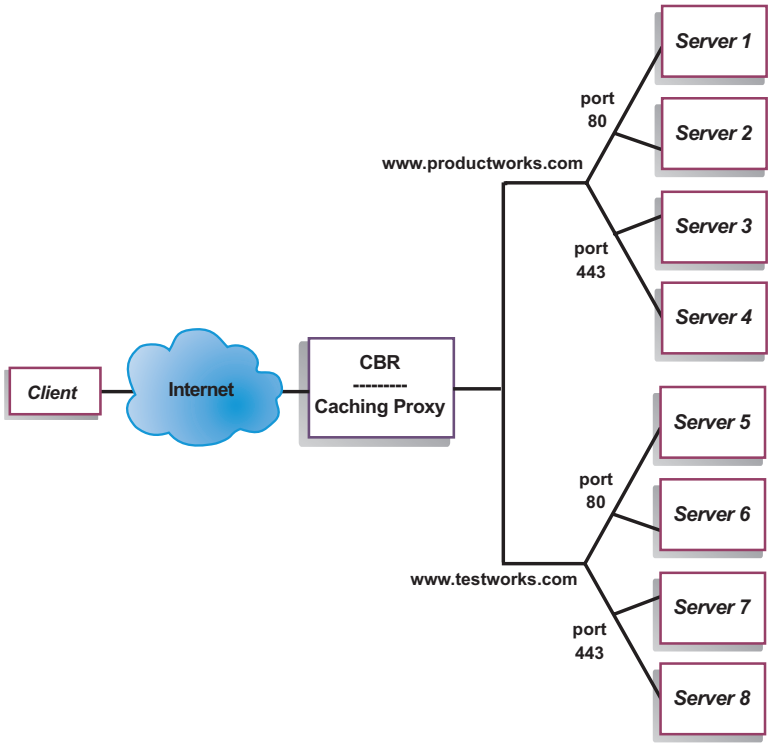


Figure 19. Example of CBR configured with 2 clusters, each with 2 ports

In this example for the CBR component, two clusters are defined with port 80 (HTTP) and port 443 (SSL) for each of the sites at www.productworks.com and www.testworks.com.

Chapter 9. Planning for Content Based Routing

This chapter describes what the network planner should consider before installing and configuring the CBR component with Caching Proxy.

- See Chapter 3, "Managing your network: Determining which Load Balancer features to use," on page 17 for an overview of features that are available for managing your network.
- See Chapter 10, "Configuring Content Based Routing," on page 75 for information on configuring the load-balancing parameters of CBR.
- See Chapter 21, "Advanced features for Dispatcher, CBR, and Site Selector," on page 161 for information on how to set up Load Balancer for more advanced functions.
- See Chapter 23, "Operating and managing Load Balancer," on page 213 for information on remote authenticated administration, Load Balancer logs, and usage of the Load Balancer components.

This chapter includes the following section:

Planning considerations

The CBR component allows you to load balance HTTP and SSL traffic using Caching Proxy to proxy the request. With CBR, you can load balance servers that you configure from your CBR configuration file using cbrcontrol commands.

Note: The Content Based Routing (CBR) component is not available on platforms that run a 64-bit JVM, except for HP-UX ia64. On HP-UX ia64, the CBR component runs as a 32-bit application. You can use the CBR forwarding method of Load Balancer's Dispatcher component to provide content-based routing without the use of Caching Proxy. See "Dispatcher's content-based routing (cbr forwarding method)" on page 41 for more information.

CBR is very similar to Dispatcher in its component structure. CBR consists of the following functions:

- **cbrserver** handles requests from the command line to the executor, manager, and advisors.
- The **executor** supports load balancing of client requests. The executor must be started in order to use the CBR component.
- The **manager** sets weights used by the executor based on:
  - Internal counters in the executor
  - Feedback from the servers provided by the advisors
  - Feedback from a system-monitoring program, such as Metric Server.

  Using the manager is optional. However, if the manager is not used, load balancing is performed using weighted round-robin scheduling based on the current server weights, and advisors will not be available.

- The **advisors** query the servers and analyze results by protocol before calling the manager to set weights as appropriate. It may not make sense to use some of these advisors in a typical configuration. You also have the option of writing your own advisors. Using the advisors is optional but recommended. Load Balancer provides a Caching Proxy (cachingproxy) advisor. See "Advisors" on page 146 for more information.
- To configure and manage the executor, advisors, and manager, use the command line (**cbrcontrol**) or the graphical user interface (**lbadmin**).

The three key functions of CBR (executor, manager, and advisors) interact to balance and dispatch the incoming requests between servers. Along with load balancing requests, the executor monitors the number of new connections and active connections and supplies this information to the manager.

Load balancing requests for different types of content

The CBR component gives you the ability to specify a set of servers that will handle a request based on regular expression matching the content of the client request. CBR allows you to partition your site so that different content or application services can be served by different sets of servers. This partitioning is transparent to clients accessing your site.

Dividing your site content for better response time

One way to divide your site would be to assign some servers to handle only cgi requests, and another set of servers to handle all other requests. This would stop compute intensive cgi scripts from slowing down the servers for normal HTML traffic, allowing clients to get better overall response time. Using this scheme, you could also assign more powerful workstations for normal requests. This would give clients better response time without the expense of upgrading all your servers. You could also assign more powerful workstations for cgi requests.

Another possibility for partitioning your site could be to direct clients who are accessing pages requiring registration to one set of servers, and all other requests to a second set of servers. This would keep casual browsers of your site from tying up resources that could be used by clients who have committed to your registration. It would also allow you to use more powerful workstations to service those clients who have registered.

You could of course combine the methods above for even more flexibility, and improved service.

Providing backup of Web server content

Because CBR allows you to specify multiple servers for each type of request, the requests can be load balanced for optimal client response. By allowing multiple servers to be assigned to each type of content, you are protected if one workstation or server fails. CBR will recognize the failure and continue to load balance client requests to the other servers in the set.

Using multiple Caching Proxy processes to improve CPU utilization

Caching Proxy communicates with a CBR process through its plug-in interface. CBR must be running on the local machine for this to work. Because these are two separate processes, multiple instances of the Caching Proxy can be running and working with a single instance of CBR. This setup might be configured in order to

segregate addresses or functionality between the Caching Proxies, or to improve the resource utilization of the machine by having several Caching Proxies handling client traffic. The proxy instances can be listening on different ports or binding to unique IP addresses on the same port, depending on what best suits the traffic requirements.

Using rules-based load balancing with CBR

CBR along with Caching Proxy examines HTTP requests using specified rule types. When running, Caching Proxy accepts client requests and queries the CBR component for the best server. Upon this query, CBR matches the request to a set of prioritized rules. When a rule is matched, an appropriate server is chosen from a preconfigured server set. Finally, CBR informs Caching Proxy which server was chosen and the request gets proxied there.

After you define a cluster to be load balanced, you must make sure that all requests to that cluster have a rule that will choose a server. If no rule is found that matches a particular request, the client will receive an error page from Caching Proxy. The easiest way to ensure that all requests will match some rule is to create an "always true" rule at a very high priority number. Make sure that the servers used by this rule can handle all the requests not explicitly handled by the rules that have a lower-numbered priority. (Note: The lower-numbered priority rules are evaluated first.)

For more information see “Configure rules-based load balancing” on page 170.

Load balancing across fully secure (SSL) connections

CBR with Caching Proxy can receive SSL transmission from the client to the proxy (client-to-proxy side) as well as support transmission from the proxy to an SSL server (proxy-to-server side). By defining an SSL port on a server in the CBR configuration to receive the SSL request from the client, you have the ability to maintain a fully secure site, using CBR to load balance across secure (SSL) servers.

In addition to other ibmproxy.conf file changes for CBR, another configuration statement needs to be added to the ibmproxy.conf file for Caching Proxy to enable SSL encryption on the proxy-to-server side. The format must be:

```
proxy uri_pattern url_pattern address
```

where *uri_pattern* is a pattern to match (for example: /secure/*), *url_pattern* is a replacement URL (for example: https://clusterA/secure/*), and *address* is the cluster address (for example: clusterA).

Load balancing client-to-proxy in SSL and proxy-to-server in HTTP

CBR with Caching Proxy can also receive SSL transmission from the client and then decrypt the SSL request before proxying the request to an HTTP server. For CBR to support client-to-proxy in SSL and proxy-to-server in HTTP, there is an optional keyword **mapport** on the cbrcontrol server command. Use this keyword when you need to indicate that the port on the server is different from the incoming port from the client. The following is an example of adding a port using the mapport keyword, where the client's port is 443 (SSL) and the server's port is 80 (HTTP):

```
cbrcontrol server add cluster:443 mapport 80
```

The port number for mapport can be any positive integer value. The default is the port number value of the incoming port from the client.

Because CBR must be able to advise on an HTTP request for a server configured on port 443 (SSL), a special advisor *ssl2http* is provided. This advisor starts on port 443 (the incoming port from the client) and advises on the server(s) configured for that port. If there are two clusters configured and each cluster has port 443 and servers configured with a different mapport, then a single instance of the advisor can open the appropriate port accordingly. The following is an example of this configuration:

```
Executor
    Cluster1
       Port:443
           Server1 mapport 80
           Server2 mapport 8080
    Cluster2
       Port:443
           Server3 mapport 80
           Server4 mapport 8080
    Manager
      Advisor ssl2http 443
```

Chapter 10. Configuring Content Based Routing

Before following the steps in this chapter, see Chapter 9, "Planning for Content Based Routing," on page 71. This chapter explains how to create a basic configuration for the CBR component of Load Balancer.

- See Chapter 20, "Manager, Advisors, and Metric Server functions for Dispatcher, CBR, and Site Selector," on page 141 and Chapter 21, "Advanced features for Dispatcher, CBR, and Site Selector," on page 161 for more complex configurations of Load Balancer.
- See Chapter 23, "Operating and managing Load Balancer," on page 213 for information on remote authenticated administration, Load Balancer logs, and usage of the Load Balancer components.

Overview of configuration tasks

Before you begin the configuration steps in this table, ensure that your CBR machine and all server machines are connected to the network, have valid IP addresses, and are able to ping one another.

Note: The Content Based Routing (CBR) component is not available on platforms that run a 64-bit JVM, except for HP-UX ia64. On HP-UX ia64, the CBR component runs as a 32-bit application. You can use the CBR forwarding method of Load Balancer's Dispatcher component to provide content-based routing without the use of Caching Proxy. See "Dispatcher's content-based routing (cbr forwarding method)" on page 41 for more information.

Table 4. Configuration tasks for the CBR component

Task	Description	Related information
Set up the CBR machine.	Finding out about the requirements.	"Setting up the CBR machine" on page 79
Set up machines to be load-balanced.	Set up your load balancing configuration.	"Step 7. Define load balanced server machines" on page 82

Methods of configuration

To create a basic configuration for the CBR component of Load Balancer, there are four basic methods:

- Command line
- Scripts
- Graphical user interface (GUI)
- Configuration wizard

To use CBR, Caching Proxy must be installed.

Note: Caching Proxy is a service that starts automatically by default after installation. You must stop Caching Proxy before starting the CBR server function (cbrserver) and modify the Caching Proxy service to start manually rather than automatically.

- For Linux or UNIX systems: Stop Caching Proxy by finding its process identifier using `ps -ef | grep ibmproxy` command and then ending the process using `kill process_id` command.
- For Windows systems: Stop Caching Proxy from the Services panel.

Command line

This is the most direct means of configuring CBR. The command parameter values must be entered in English characters. The only exceptions are host names (used, for example, in cluster and server commands) and file names.

To start CBR from the command line:

- On Linux or UNIX systems: As root user, issue **cbrserver** command from the command prompt. (To stop the service, issue the following: **cbrserver stop**.

  On Windows systems: Click **Start** > **Control Panel** > **Administrative Tools** > **Services**. Right-click **IBM Content Based Routing** and select **Start**. To stop the service, follow the same steps and select **Stop**.
- Next, issue the CBR control commands you want in order to set up your configuration. The procedures in this manual assume use of the command line. The command is **cbrcontrol.** For more information about commands, see Chapter 26, "Command reference for Dispatcher and CBR," on page 287.
- Start Caching Proxy. Issue **ibmproxy** command from the command prompt. (You must start the executor prior to starting Caching Proxy.)

  Note: For Windows platforms: Start Caching Proxy from the Services panel: **Start** > **Control Panel** > **Administrative Tools** > **Services**.

You can enter an abbreviated version of the cbrcontrol command parameters. You only need to enter the unique letters of the parameters. For example, to get help on the file save command, you can type **cbrcontrol he f** instead of **cbrcontrol help file**.

To start up the command line interface: issue **cbrcontrol** to receive a cbrcontrol command prompt.

To end the command line interface: issue **exit** or **quit**.

Notes:

1. On Windows platform, the Dispatcher component's dsserver starts automatically. If you are using only CBR and not the Dispatcher component, you can stop dsserver from starting automatically as follows:
   a. In the Services window, right-click IBM Dispatcher.
   b. Select Properties.
   c. In the **Startup type** field, select Manual.
   d. Click OK, and close the Services window.
2. When you configure Content Based Routing (CBR) from the operating system's command prompt rather than from the cbrcontrol>> prompt, take care using these characters:

   ( ) right and left parentheses

   & ampersand

   | vertical bar

   ! exclamation point

   * asterisk

The operating system's shell may interpret these as special characters and convert them to alternate text before cbrcontrol evaluates them.

The special characters in the above list are optional characters on the **cbrcontrol rule add** command, and are used when specifying a pattern for a content rule. For example, the following command might be valid only when using the cbrcontrol>> prompt.

```
rule add 10.1.203.4:80:cbr_prod_rule_ek type content
  pattern uri=/nipoek/*
```

For this same command to work at the operating system's prompt, double quotation marks (" ") must be placed around the pattern as follows:

```
cbrcontrol rule add 10.1.203.4:80:cbr_prod_rule_ek type content
  pattern "uri=/nipoek/*"
```

If the quotation marks are not used, some of the pattern might be truncated when the rule is saved in CBR. Note that quotation marks are not supported when using the cbrcontrol>> command prompt.

Scripts

You can enter the commands for configuring CBR into a configuration script file and run them together.

Note: To quickly run the content of a script file (for example, myscript), use either of the following commands:

- To update the current configuration, run the following executable commands from your script file:

  cbrcontrol file appendload *myscript*

- To completely replace the current configuration, run the following executable commands from your script file:

  cbrcontrol file newload *myscript*

To save the current configuration into a script file (for example, savescript), run the following command:

cbrcontrol file save *savescript*

This command will save the configuration script file in the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/configurations/cbr**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\configurations\cbr**

GUI

For general instructions and an example of the graphical user interface (GUI), see Figure 38 on page 403.

To start the GUI, follow these steps

1. Ensure cbrserver is running. As root user or administrator, issue the following from a command prompt: **cbrserver**
2. Do one of the following actions, depending on your operating system:
   - For AIX, HP-UX, Linux, or Solaris systems: enter **lbadmin**
   - For Windows systems: click **Start** > **Programs** > **IBM WebSphere** > **Edge Components** > **IBM Load Balancer** > **Load Balancer**

3. Start Caching Proxy. (From the GUI, you must first connect to the Host and start the Executor for the CBR component prior to starting Caching Proxy.) Do one of the following:
   - For AIX, HP-UX, Linux, or Solaris systems: To start Caching Proxy, enter **ibmproxy**
   - For Windows systems: To start Caching Proxy, go to the Services panel: **Start** > **Control Panel** > **Administrative Tools** > **Services**

In order to configure the CBR component from the GUI, you must first select **Content Based Routing** in the tree structure. You can start the manager after you connect to a Host. You can also create clusters containing ports and servers, and start advisors for the manager.

The GUI can be used to do anything that you would do with the **cbrcontrol** command. For example, to define a cluster using the command line, you would enter **cbrcontrol cluster add** *cluster* command. To define a cluster from the GUI, right-click Executor, then in the pop-up menu, left-click **Add Cluster.** Enter the cluster address in the pop-up window, then click **OK.**

Pre-existing CBR configuration files can be loaded using the **Load New Configuration** (for completely replacing the current configuration) and **Append to Current Configuration** (for updating the current configuration) options presented in the **Host** pop-up menu. You should save your CBR configuration to a file periodically using the **Save Configuration File As** option also presented in the **Host** pop-up menu. The **File** menu located at the top of the GUI will allow you to save your current host connections to a file or restore connections in existing files across all Load Balancer components.

You can access **Help** by clicking the question mark icon in the upper right corner of the Load Balancer window.

- **Help: Field level —** describes each field, default values
- **Help: How do I —** lists tasks that can be done from that screen
- **InfoCenter —** provides centralized access to product information

In order to run a command from the GUI: highlight the Host node from the GUI tree and select **Send command...** from the Host pop-up menu. In the command entry field, type the command that you want to run, for example: **executor report**. The results and history of the commands run in the current session appear in the window provided.

For more information about using the GUI, see Appendix A, “GUI: General instructions,” on page 403.

Configuration wizard

If you are using the configuration wizard, follow these steps:

1. Start the cbrserver: issue **cbrserver** on the command prompt as root user or administrator.
2. Start the wizard function of CBR:

   Launch the wizard from the command prompt by issuing the **cbrwizard**. Or, select the Configuration Wizard from the CBR component menu as presented in the GUI.
3. Start Caching Proxy in order to load balance HTTP or HTTPS (SSL) traffic.

For AIX, HP-UX, Linux, or Solaris systems: To start Caching Proxy, enter **ibmproxy**

For Windows systems: To start Caching Proxy, go to the Services panel: **Start** > **Control Panel** > **Administrative Tools** > **Services**

The CBR wizard guides you step-by-step through the process of creating a basic configuration for the CBR component. It asks you questions about your network and guides you as you set up a cluster that enables CBR to load balance traffic between a group of servers.

Setting up the CBR machine

Before setting up the CBR machine, you must be the root user (for AIX, HP-UX, Linux, or Solaris systems) or the Administrator (for Windows systems).

You will need one IP address for each cluster of servers that is set up. A cluster address is an address that is associated with a host name (such as www.company.com). This IP address is used by a client to connect to the servers in a cluster. Specifically, this address is found in the URL request from the client. All requests made to the same cluster address are load balanced by CBR.

For Solaris systems only: Before using the CBR component, the system defaults for IPCs (Inter-process Communication) must be modified. The maximum size of a shared memory segment and the number of semaphore identifiers need to be increased. To tune your system to support CBR, edit the **/etc/system** file on your system to add the following statements and then reboot:

```
set shmsys:shminfo_shmmax=0x02000000
set semsys:seminfo_semmap=750
set semsys:seminfo_semmni=30
set semsys:seminfo_semmns=750
set semsys:seminfo_semmnu=30
set semsys:seminfo_semume=30
```

If you do not increase the shared memory segment to the values shown above, **cbrcontrol executor start** command will fail.

Step 1. Configure Caching Proxy to use CBR

To use CBR, Caching Proxy must be installed.

Note: Caching Proxy is a service that starts automatically by default after installation. You must stop Caching Proxy before starting the CBR server function and modify the Caching Proxy service to start manually rather than automatically.

- For AIX, HP-UX, Linux, and Solaris systems: Stop Caching Proxy by finding its process identifier using `ps -ef | grep ibmproxy` command and then ending the process using `kill` *`process_id`* command.
- For Windows systems: Stop Caching Proxy from the Services panel.

You must make the following modifications to the Caching Proxy configuration file (ibmproxy.conf):

Verify that the incoming URL directive **CacheByIncomingUrl** is "off" (default).

In the mapping rules section of the configuration file, for every cluster, add a mapping rule similar to:

```
Proxy    /*  http://cluster.domain.com/*    cluster.domain.com
```

Note: CBR sets the protocol, server, and target port at a later time.

There are four entries that must be edited for the CBR Plug-in:

- ServerInit
- PostAuth
- PostExit
- ServerTerm

Each entry must be on a single line. There are several instances of "ServerInit" in the ibmproxy.conf file, one for each plug-in. The entries for the "CBR Plug-in" should be edited and uncommented.

The specific additions to the configuration file for each of the operating systems follow:

Table 5. Necessary additions to the CBR configuration file, by operating system

Operating system	**Additions to CBR configuration file**
AIX, HP-UX, Linux, and Solaris systems	ServerInit /opt/ibm/edge/lb/servers/lib/liblbcbr.so:ndServerInit PostAuth /opt/ibm/edge/lb/servers/lib/liblbcbr.so:ndPostAuth PostExit /opt/ibm/edge/lb/servers/lib/liblbcbr.so:ndPostExit ServerTerm /opt/ibm/edge/lb/servers/lib/liblbcbr.so:ndServerTerm
Windows systems	ServerInit <*install_root*>ibm\edge\lb\servers\lib\liblbcbr.dll:ndServerInit PostAuth <*install_root*>ibm\edge\lb\servers\lib\liblbcbr.dll:ndPostAuth PostExit <*install_root*>ibm\edge\lb\servers\lib\liblbcbr.dll:ndPostExit ServerTerm <*install_root*>ibm\edge\lb\servers\lib\liblbcbr.dll:ndServerTerm

Step 2. Start the server function

To start the CBR server function, type **cbrserver** on the command line.

A default configuration file (default.cfg) gets automatically loaded when starting cbrserver. If you decide to save the CBR configuration in default.cfg, then everything saved in this file is automatically loaded next time cbrserver gets started.

Step 3. Start the executor function

To start the executor function, enter the **cbrcontrol executor start** command. You may also change various executor settings at this time. See “dscontrol executor — control the executor” on page 299.

Step 4. Define a cluster and set cluster options

CBR will balance the requests sent for the cluster to the corresponding servers configured on the ports for that cluster.

The cluster is the symbolic name located in the host portion of the URL and should match the name used in the Proxy statement of the ibmproxy.conf file.

Clusters defined in CBR should be defined to match the incoming request. A cluster must be defined using the same hostname or IP address that the incoming request will contain. For example, if the request will enter as the IP address, the cluster must be defined as the IP address. If there is more than one hostname that resolves to a single IP address (and requests can arrive with any one of those hostnames) then all the hostnames should be defined as clusters.

To define a cluster, issue the following command:

```
cbrcontrol cluster add cluster
```

To set cluster options, issue the following command:

```
cbrcontrol cluster set cluster option value
```

For more information, see Chapter 26, "Command reference for Dispatcher and CBR," on page 287.

Step 5. Alias the network interface card (optional)

If you are running Caching Proxy configured as a reverse proxy, when load balancing for multiple Web sites, you must add the cluster address for each Web site to at least one of the network interface cards of the Load Balancer machine. Otherwise, this step can be omitted.

For **AIX, HP-UX, Linux, or Solaris systems**: To add the cluster address to the network interface, use the ifconfig command. Use the command for your operating system as shown in Table 6.

Table 6. Commands to alias the NIC

AIX	**ifconfig** *interface_name* **alias** *cluster_address* **netmask** *netmask*
HP-UX	**ifconfig** *interface_name cluster_address* **netmask** *netmask* **up**
Linux	**ifconfig** *interface_name cluster_address* **netmask** *netmask* **up**
Solaris 9, and Solaris 10	**ifconfig** *interface_name* **addif** *cluster_address* **netmask** *netmask* **up**

Note: For Linux and HP-UX systems, *interface_name* must have a unique number for each cluster address that is added, for example: eth0:1, eth0:2, and so on.

For **Windows 2003**: To add the cluster address to the network interface, do the following:

1. Click **Start** > **Control Panel** > **Network Connections** > *Local Area Connection*
2. Click **Properties**.
3. Select **Internet Protocol (TCP/IP)** and click **Properties**.
4. Select **Use the following IP address** and click **Advanced**.
5. Click **Add** and then type the IP address and subnet mask for the cluster.

Step 6. Define ports and set port options

The port number is the port on which the server applications are listening. For CBR with Caching Proxy running HTTP traffic, this is typically port 80.

To define a port to the cluster you defined in the previous step, issue the following command:

```
cbrcontrol port add cluster:port
```

To set port options, issue the following command:

```
cbrcontrol port set cluster:port option value
```

For more information, see Chapter 26, "Command reference for Dispatcher and CBR," on page 287.

Step 7. Define load balanced server machines

The server machines are the machines running the applications that you want load balanced. The *server* is the symbolic name or dotted decimal address of the server machine. To define a server on the cluster and port, issue the following command:

```
cbrcontrol server add cluster:port:server
```

You must define more than one server per port on a cluster in order to perform load balancing.

Step 8. Add rules to your configuration

This is the key step in configuring CBR with Caching Proxy. A rule defines how a URL request will be distinguished and sent to one of the appropriate set of servers. The special rule type used by CBR is called a content rule. To define a content rule, issue the following command:

```
cbrcontrol rule add cluster:port:rule type content pattern pattern
```

The value *pattern* is the regular expression that is compared to the URL in each client request. For more information on how to configure the pattern, see Appendix B, "Content rule (pattern) syntax," on page 409.

Some other rule types defined in Dispatcher can also be used in CBR. For more information, see "Configure rules-based load balancing" on page 170.

Step 9. Add servers to your rules

When a rule is matched by a client request, the rule's set of servers is queried for which server is best. The rule's server set is a subset of the servers defined in the port. To add servers to a rule's server set, issue the following command:

```
cbrcontrol rule useserver cluster:port:rule server
```

Step 10. Start the manager function (optional)

The manager function improves load balancing. To start the manager, issue the following command:

```
cbrcontrol manager start
```

Step 11. Start the advisor function (optional)

The advisors give the manager more information about the ability of the load balanced server machines to respond to requests. An advisor is specific to a protocol. For example, to start the HTTP advisor, issue the following command:

```
cbrcontrol advisor start http port
```

Step 12. Set cluster proportions as required

If you start advisors, you may modify the proportion of importance given to advisor information being included in the load balancing decisions. To set the cluster proportions, issue the **cbrcontrol cluster set** *cluster* **proportions** command. For more information, see "Proportion of importance given to status information" on page 142.

Step 13. Start Caching Proxy

- AIX systems: Add to your LIBPATH environment variable:

  ```
  /opt/ibm/edge/lb/servers/lib
  ```

- Linux, HP-UX or Solaris systems: Add to your LD_LIBRARY_PATH environment variable:

  ```
  /opt/ibm/edge/lb/servers/lib
  ```

- Windows systems: Add to your PATH environment variable:

  ```
  <install_root>ibm\edge\lb\servers\lib
  ```

In the new environment, start Caching Proxy:

- From the command prompt, issue **ibmproxy**
- For Windows systems, you can start Caching Proxy from the Services panel. Click **Start** -> **Control Panel** -> **Administrative Tools** -> **Services**.

CBR configuration example

To configure CBR follow these steps:

1. Start CBR: issue the **cbrserver** command.
2. Start up the command line interface: issue the **cbrcontrol** command.
3. The **cbrcontrol** prompt is displayed. Issue the following commands. *(cluster(c),port(p),rule(r),server(s))*
   - executor start
   - cluster add c
   - port add c:p
   - server add c:p:s
   - rule add c:p:r type content pattern uri=*
   - rule useserver c:p:r s
4. Start Caching Proxy: Issue the **ibmproxy** command. (For Windows platform, start Caching Proxy from the Services panel.
5. Remove all proxy configurations from the browser.
6. Load `http://c/` into your browser where "'c" is the cluster that you configured previously.
   - Server "s" is invoked
   - The following Web page is displayed `http://s/`

Part 4. Site Selector component

This part provides information on a quick start configuration, planning considerations, and describes the methods of configuring Load Balancer's Site Selector component. It contains the following chapters:

Chapter 11. Quick start configuration

This quick start example shows how to create a site name configuration using Site Selector to load balance traffic among a set of servers based on the domain name used on a client request.

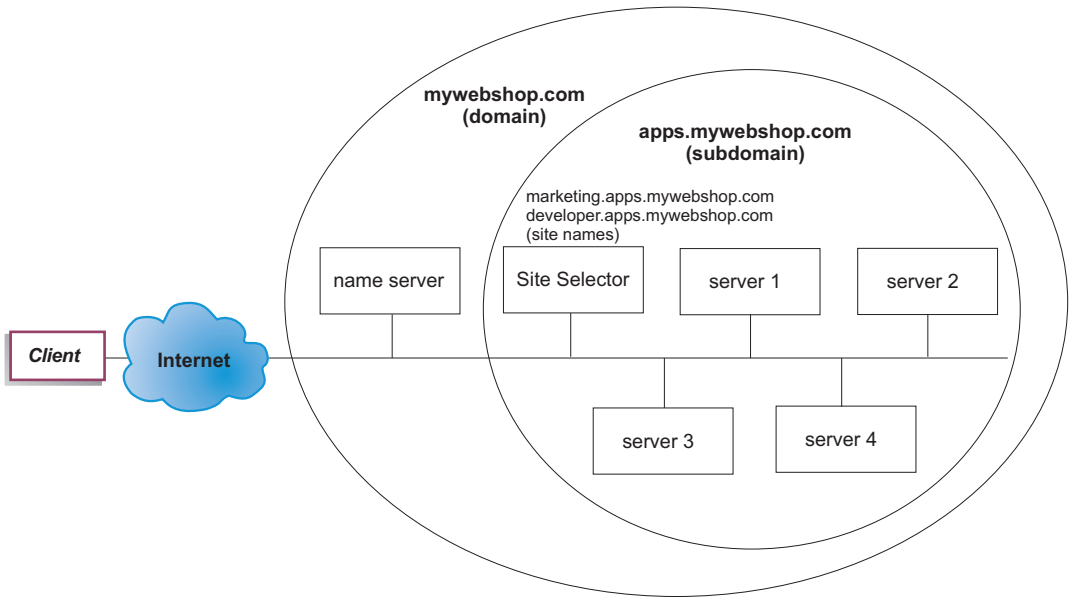


Figure 20. A simple Site Selector configuration

What do you need?

For this quick start configuration example, you will need the following:

- Administrative access to your site's name server
- Four servers (server1, server2, server3, server4) configured to the network and an additional server with the installed Site Selector component

Note: If you collocate Site Selector on one of the load balanced servers, then you will need four servers instead of five. However, collocation will impact the performance of the load balanced servers.

How do you prepare?

For this quick start example, the company's site domain is mywebshop.com. Site Selector is responsible for a subdomain within mywebshop.com. Therefore, you need to define a subdomain within mywebshop.com. For example: apps.mywebshop.com. Site Selector is not a fully implemented DNS, such as BIND, and acts as a leafnode in a DNS hierarchy. Site Selector is authoritative for the apps.mywebshop.com subdomain. Subdomain apps.mywebshop.com will include the following site names: marketing.apps.mywebshop.com and developer.apps.mywebshop.com.

1. Update the company site's domain name server (see Figure 20 on page 87). Create a name server record in the named.data file for the subdomain (apps.mywebshop.com) where Site Selector is the authoritative name server:

   apps.mywebshop.com. IN NS siteselector.mywebshop.com
2. Ensure that the fully qualified hostname or site does not resolve in the current domain name system.
3. Install Metric Server on the servers (server1, server2, server3, server4) that you intend to have Site Selector load balance. See "Metric Server" on page 157 for more information.

Configuring the Site Selector component

With Site Selector, you can create a configuration by using the command line, the configuration wizard, or the graphical user interface (GUI). For this quick start example, configuration steps are demonstrated using the command line.

Note: The parameter values must be typed in English characters. The only exceptions are parameter values for host names and file names.

Configuring with the command line

From a command prompt, follow these steps:

1. Start the ssserver on the machine hosting Site Selector. As root user or administrator, issue the following from a command prompt: **ssserver**

   Note: For Windows platform: Start ssserver (IBM Site Selector) from the Services panel: **Start** > **Control Panel** > **Administrative Tools** > **Services**.
2. Start the name server on the Site Selector configuration:

   sscontrol nameserver start
3. Configure the site names (marketing.apps.mywebshop.com and developer.apps.mywebshop.com) on Site Selector:

   sscontrol sitename add marketing.apps.mywebshop.com

   sscontrol sitename add developer.apps.mywebshop.com
4. Add the servers to the Site Selector configuration. (Configure server1 and server2 to site name marketing.apps.mywebshop.com. Configure server3 and server4 to site name developer.apps.myeebshop.com):

   sscontrol server add marketing.apps.mywebshop.com:server1+server2

   sscontrol server add developer.apps.mywebshop.com:server3+server4
5. Start the manager function of Site Selector:

   sscontrol manager start
6. Start the advisor function of Site Selector (HTTP advisor for marketing.apps.mywebshop.com and FTP advisor for developer.apps.mywebshop):

   sscontrol advisor start http marketing.apps.mywebshop.com:80

   sscontrol advisor start ftp developer.apps.mywebshop.com:21

   Site Selector will now make sure that client requests are not sent to a failed server.
7. Ensure the Metric Server has been started on each of the load-balanced servers.

Your basic Site Selector configuration is now complete.

Testing your configuration

Test to see if the configuration is working:

1. From a client, that has a primary DNS configured as the nameserver responsible for mywebshop.com, try to ping one of the configured site names.
2. Connect to the application. For example:
   - Open a browser, request marketing.apps.mywebshop.com and a valid page should be served
   - Open an FTP client to developer.apps.mywebshop.com and enter a valid user and password
3. Look at the results of the following command:

   sscontrol server status marketing.apps.mywebshop.com:

   sscontrol server status developer.apps.mywebshop.com:

   The total hits entry of each server should add up to the ping and application request

Configuring with the graphical user interface (GUI)

For information on using the Site Selector GUI, see "GUI" on page 96 and Appendix A, "GUI: General instructions," on page 403.

Configuring with the configuration wizard

For information on using the Site Selector wizard, see "Configuration wizard" on page 97.

Chapter 12. Planning for Site Selector

This chapter describes what the network planner should consider before installing and configuring the Site Selector component.

- See Chapter 3, "Managing your network: Determining which Load Balancer features to use," on page 17 for an overview of features that are available for managing your network.
- See Chapter 13, "Configuring Site Selector," on page 95 for information on configuring the load-balancing parameters of Site Selector.
- See Chapter 21, "Advanced features for Dispatcher, CBR, and Site Selector," on page 161 for information on how to set up Load Balancer for more advanced functions.
- See Chapter 23, "Operating and managing Load Balancer," on page 213 for information on remote authenticated administration, Load Balancer logs, and usage of the Load Balancer components.

This chapter includes the following sections:

Planning Considerations

Site Selector works in conjunction with a domain name server to load balance among a group of servers using measurements and weights that are gathered. You can create a site configuration to let you load balance traffic among a group of servers based on the domain name used for a client's request.

Limitations: The DNS queries that Site Selector supports are Type A queries only. Any other query types will result in a return code of NOTIMPL (Not Implemented). If an entire domain is delegated to Site Selector, ensure that the domain receives only Type A queries.

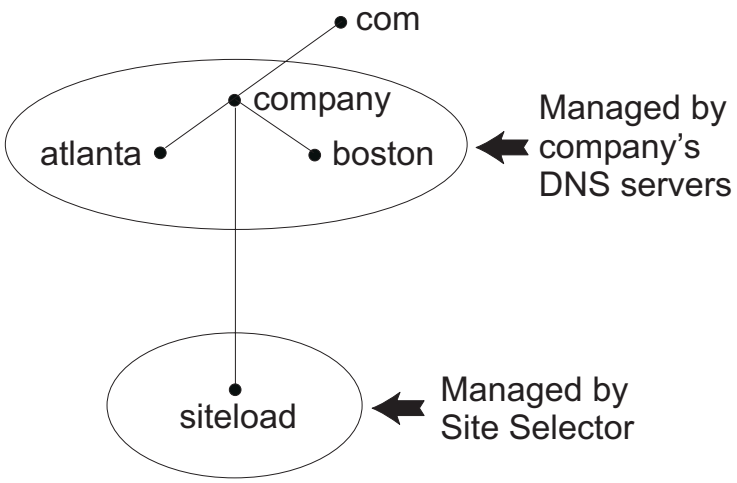


Figure 21. Example of a DNS environment

When setting up a subdomain for Site Selector within your DNS environment, Site Selector should have authority over its own subdomain. For example (see Figure 21 on page 91), your company has been assigned authority over **company.com** domain. Within the company, there are several subdomains. Site Selector would have authority for **siteload.company.com**, while the DNS server(s) would still maintain authority for **atlanta.company.com** and **boston.company.com**.

In order for company's name server to recognize Site Selector as having authority for the siteload subdomain, a name server entry will need to be added to its named data file. For example, on AIX systems, a name server entry would look like the following:

```
siteload.company.com. IN NS siteselector.company.com.
```

Where **siteselector.company.com** is the hostname of the Site Selector machine. Equivalent entries would need to be made in any other named database files for use by DNS servers.

A client submits a request for resolution of a domain name to a name server within its network. Name server forwards the request to the Site Selector machine. Site Selector then resolves the domain name to the IP address of one of the servers that has been configured under the site name. Site Selector returns the IP address of the selected server to the name server. Name server returns the IP address to the client. (Site Selector acts as a non-recursive (leaf node) name server, and it will return an error if it does not resolve the domain name request.)

Refer to Figure 5 on page 12 which illustrates a site in which Site Selector is used in conjunction with a DNS system to load balance across local and remote servers.

Site Selector consists of the following functions:

- **ssserver** handles request from the command line to the Name Server, manager, and advisors.
- The **name server** function supports the load balancing of incoming name server requests. You must start the name server function for Site Selector to begin providing DNS resolution. Site Selector listens on port 53 for incoming DNS requests. If the requesting site name is configured, then Site Selector returns a single server address (from a set of server addresses) associated with the site name.
- The **manager** sets weights used by the name server based on:
  - Feedback from the servers provided by the advisors
  - Feedback from a system-monitoring program, such as Metric Server.

  Using the manager is optional. However, if the manager is not used, load balancing is performed using weighted round-robin scheduling based on the current server weights, and advisors will not be available.
- The **Metric Server** is a system monitoring component of Load Balancer that you install on the backend server machine. (If you collocate Load Balancer on a server machine that is being load balanced, then you would install Metric Server on the Load Balancer machine.)

  With Metric Server, Site Selector can monitor the level of activity on a server, detect when a server is the least heavily loaded, and detect a failed server. The load is a measure of how hard the server is working. The system Site Selector administrator controls the type of measurement used to measure the load. You can configure Site Selector to suit your environment, considering such factors as frequency of access, the total number of users, and types of access (for example, short queries, long-running queries, or CPU-intensive loads).

Load balancing is based on server weights. For Site Selector, there are four proportions which the manager uses to determine weights:

- CPU
- memory
- port
- system

CPU and memory values are all supplied by Metric Server. Consequently, use of Metric Server is *recommended* with the Site Selector component.

See "Metric Server" on page 157 for more information.

- The **advisors** query the servers and analyze results by protocol before calling the manager to set weights as appropriate. It may not make sense to use some of these advisors in a typical configuration. You also have the option of writing your own advisors. Using the advisors is optional but recommended. See "Advisors" on page 146 for more information.
- To configure and manage the name server, advisors, Metric Server, and manager, use the command line (**sscontrol**) or the graphical user interface (**lbadmin**).

The four key functions of Site Selector (name server, manager, Metric Server, and advisors) interact to balance and resolve the incoming requests between servers.

TTL considerations

Using DNS-based load balancing requires that caching of name resolutions be disabled. The TTL (time to live) value determines the effectiveness of DNS-based load balancing. TTL determines how long another nameserver will cache the resolved response. Small TTL values allow for subtle changes in the server or network load to be realized more quickly. However, disabling caching requires that clients contact the authoritative name server for every name resolution request, thus potentially increasing the client latency. When choosing a TTL value, careful consideration should be given to the impact that disabled-caching has on an environment. Also be aware that DNS-based load balancing is potentially limited by client-side caching of name resolutions.

TTL can be configured using the **sscontrol sitename [add | set]** command. See "sscontrol sitename — configure a sitename" on page 362 for more information.

Using the Network Proximity feature

Network proximity is the calculation of each server's nearness to the requesting client. To determine network proximity, the Metric Server agent (which must reside on each load-balanced server) sends a ping to the client IP address and returns the response time to Site Selector. Site Selector uses the proximity response in the load-balancing decision. Site Selector combines the network proximity response value with the weight from the manager to create a combined final weight value for the server.

Use of the network proximity feature with Site Selector is optional.

The Site Selector provides the following network proximity options that can be set per site name:

- Cache life: The amount of time a proximity response is valid and saved in the cache.
- Proximity percent: The importance of the proximity response versus the health of the server (as input from the manager weight).

- Wait for all: Determines whether to wait for all proximity (ping) responses from the servers before responding to the client request.

  If set to **yes**, the Metric Server pings the client to obtain the proximity response time. Name server waits for all Metric Servers to respond or for a time-out to occur. Then, for each server, the name server combines the proximity response time with the weight the manager calculated to create a "combined weight" value for each server. Site Selector will supply the client with the server IP address with the best combined weight. (It is expected that most client name servers have a 5 second time-out. Site Selector tries to respond before that time-out is exceeded.)

  If set to **no**, a name resolution is provided to the client based on the current manager weights. Then, the Metric Server pings the client to obtain the proximity response time. The name server caches the response time it receives from the Metric Server. When the client returns for a second request, the name server combines the current manager weight with the cached ping response value for each server to obtain the server with the best "combined weight." Site Selector returns this server's IP address to the client for its second request.

Network proximity options can be set on the **sscontrol sitename [add | set]** command. See Chapter 27, “Command reference for Site Selector,” on page 341 for more information.

Chapter 13. Configuring Site Selector

Before following the steps in this chapter, see Chapter 12, "Planning for Site Selector," on page 91. This chapter explains how to create a basic configuration for the Site Selector component of Load Balancer.

- See Chapter 20, "Manager, Advisors, and Metric Server functions for Dispatcher, CBR, and Site Selector," on page 141 and Chapter 21, "Advanced features for Dispatcher, CBR, and Site Selector," on page 161 for more complex configurations of Load Balancer.
- See Chapter 23, "Operating and managing Load Balancer," on page 213 for information on remote authenticated administration, Load Balancer logs, and usage of the Load Balancer components.

Overview of configuration tasks

Note: Before you begin the configuration steps in this table, ensure that your Site Selector machine and all server machines are connected to the network, have valid IP addresses, and are able to ping one another.

Table 7. Configuration tasks for the Site Selector component

Task	Description	Related information
Set up the Site Selector machine.	Finding out about the requirements.	"Setting up the Site Selector machine" on page 98
Set up machines to be load-balanced.	Set up your load balancing configuration.	"Step 4. Define load balanced server machines" on page 98

Methods of configuration

To create a basic configuration for the Site Selector component of Load Balancer, there are four basic methods of configuring the Site Selector component:

- Command line
- Scripts
- Graphical user interface (GUI)
- Configuration wizard

Command line

This is the most direct means of configuring Site Selector. The command parameter values must be entered in English characters. The only exceptions are host names (used, for example, in site name and server commands) and file names.

To start Site Selector from the command line:

1. Issue the **ssserver** command from the command prompt. To stop the service, type: **ssserver stop**

   Note: For Windows systems, click **Start** > **Control Panel** > **Administrative Tools** > **Services**. Right-click **IBM Site Selector** and select **Start**. To stop the service, follow the same steps and select **Stop**.

2. Next, issue Site Selector control commands you want in order to set up your configuration. The procedures in this manual assume use of the command line. The command is **sscontrol**. For more information about commands, see Chapter 27, "Command reference for Site Selector," on page 341.

You can enter a minimized version of the sscontrol command parameters. You only need to enter the unique letters of the parameters. For example, to get help on the file save command, you can type **sscontrol he f** instead of **sscontrol help file**.

To start up the command line interface: issue **sscontrol** to receive an sscontrol command prompt.

To end the command line interface: issue **exit** or **quit**.

Note: On Windows platform, the Dispatcher component's dsserver starts automatically. If you are using only Site Selector and not the Dispatcher component, you can stop dsserver from starting automatically as follows:

1. From Windows Services, right-click IBM Dispatcher.
2. Select Properties.
3. In the **Startup type** field, select Manual.
4. Click OK, and close the Services window.

Scripts

The commands for configuring Site Selector can be entered into a configuration script file and run together.

Note: To quickly run the content of a script file (for example, myscript), use either of the following commands:

- For updating the current configuration, run the executable commands from your script file using —

  sscontrol file appendload *myscript*

- For completely replacing the current configuration, run the executable commands from your script file using —

  sscontrol file newload *myscript*

To save the current configuration into a script file (for example, savescript), run the following command:

sscontrol file save *savescript*

This command will save the configuration script file in the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/configurations/ss**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\configurations\ss**

GUI

For general instructions and an example of the GUI, see Figure 38 on page 403.

To start the GUI, follow these steps

1. Ensure ssserver is running. As root user or administrator, issue the following from a command prompt: **ssserver**
2. Next, do one of the following:

- For AIX, HP-UX, Linux, or Solaris systems: enter **lbadmin**
- For Windows systems: click **Start** > **Programs IBM WebSphere** > **Edge Components** > **IBM Load Balancer** > **Load Balancer**

In order to configure the Site Selector component from the GUI, you must first select **Site Selector** in the tree structure. After you connect to a host running ssserver, you can create site names containing servers, start the manager, and start advisors.

The GUI can be used to do anything that you would do with the **sscontrol** command. For example, to define a site name using the command line, you would enter **sscontrol sitename add** *sitename* command. To define a site name from the GUI, right-click Name Server, then in the pop-up menu, left-click **Add Site Name.** Enter the site name in the pop-up window, then click **OK.**

Pre-existing Site Selector configuration files can be loaded using the **Load New Configuration** (for completely replacing the current configuration) and **Append to Current Configuration** (for updating the current configuration) options presented in the **Host** pop-up menu. You should save your Site Selector configuration to a file periodically using the **Save Configuration File As** option also presented in the **Host** pop-up menu. The **File** menu located at the top of the GUI will allow you to save your current host connections to a file or restore connections in existing files across all the Load Balancer components.

To run a command from the GUI: highlight the Host node from the GUI tree and select **Send command....** from the Host pop-up menu. In the command entry field, type the command that you want to run, for example: **nameserver status**. The results and history of the commands run in the current session appear in the window provided.

You can access **Help** by clicking the question mark icon in the upper right corner of the Load Balancer window.

- **Help: Field level —** describes each field, default values
- **Help: How do I —** lists tasks that can be done from that screen
- **InfoCenter —** provides centralized access to product information

For more information about using the GUI, see Appendix A, "GUI: General instructions," on page 403.

Configuration wizard

If you are using the configuration wizard, follow these steps:

1. Start the ssserver on Site Selector:
   - Run the following as root use or Administrator:

     ssserver
2. Start the wizard function of Site Selector, **sswizard.**

   You can launch this wizard from the command prompt by issuing the **sswizard**. Or, select the Configuration Wizard from the Site Selector component menu as presented in the GUI.

The Site Selector wizard guides you step-by-step through the process of creating a basic configuration for the Site Selector component. It asks you questions about your network and guides you as you setup a site name that enables Site Selector to load balance traffic between a group of servers.

Setting up the Site Selector machine

Before setting up the Site Selector machine, you must be the root user (for AIX, HP-UX, Linux, or Solaris systems) or the Administrator (for Windows systems).

You will need an unresolvable fully qualified hostname to use as a site name for a group of servers that you set up. The site name is the name that the clients use to access your site (such as www.yourcompany.com). Site Selector will load-balance traffic for this site name among the group of servers using DNS.

Step 1. Start the server function

To start the Site Selector server function, type **ssserver** on the command line.

Note: A default configuration file (default.cfg) gets automatically loaded when starting ssserver. If you decide to save the configuration in default.cfg, then everything saved in this file is automatically loaded next time ssserver gets started.

Step 2. Start the Name Server

To start the Name Server, enter the **sscontrol nameserver start** command.

Optionally, start the Name Server using the bindaddress keyword to bind only to the specified address.

Step 3. Define a site name and set site name options

Site Selector will balance the requests sent for the site name to the corresponding servers configured to it.

The site name is an unresolvable host name that the client will request. The site name must be a fully qualified domain name (for example, www.dnsdownload.com). When a client requests this site name, one of the server IP addresses associated with the site name is returned.

To define a site name, issue the following command:

```
sscontrol sitename add sitename
```

To set site name options, issue the following command:

```
sscontrol sitename set sitename option value
```

For more information, see Chapter 27, “Command reference for Site Selector,” on page 341.

Step 4. Define load balanced server machines

The server machines are the machines running the applications that you want load balanced. The *server* is the symbolic name or dotted decimal address of the server machine. To define a server on the site name from step 3, issue the following command:

```
sscontrol server add sitename:server
```

You must define more than one server under a site name in order to perform load balancing.

Step 5. Start the manager function (optional)

The manager function enhances load balancing. Prior to starting the manager function, ensure that the metric server is installed in all the load-balanced machines.

To start the manager, issue the following command:

```
sscontrol manager start
```

Step 6. Start the advisor function (optional)

The advisors give the manager more information about the ability of the load balanced server machines to respond to requests. An advisor is specific to a protocol. The Load Balancer supplies many advisors. For example, to start the HTTP advisor for a specific site name, issue the following command:

```
sscontrol advisor start http sitename:port
```

Step 7. Define system metric (optional)

See "Metric Server" on page 157 for information on using system metrics and Metric Server.

Step 8. Set site name proportions as required

If you start advisors, you may modify the proportion of importance given to advisor (port) information being included in the load balancing decisions. To set the site name proportions, issue the **sscontrol sitename set** *sitename* **proportions** command. For more information, see "Proportion of importance given to status information" on page 142.

Setting up server machines for load balancing

Use Metric Server with the Site Selector component. Refer to "Metric Server" on page 157 for information on setting up Metric Server on all server machines that Site Selector is load balancing.

Part 5. Cisco CSS Controller component

This part provides information on a quick start configuration, planning considerations, and describes the methods of configuring Load Balancer's Cisco CSS Controller component. It contains the following chapters:

Chapter 14. Quick start configuration

This quick start example shows how to create a configuration using the Cisco CSS Controller component. Cisco CSS Controller provides server weight information that assists Cisco CSS Switch in determining optimal server selection for load balancing decisions.

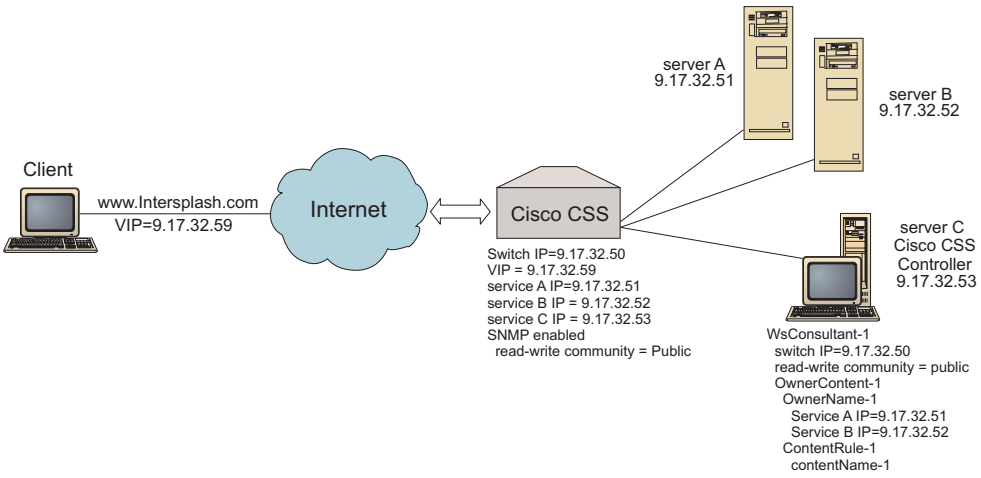


Figure 22. A simple Cisco CSS Controller configuration

What do you need?

For this quick start configuration example, you will need the following:

- A Cisco CSS Switch
- A server machine with the Cisco CSS Controller component
- Two Web server machines
- This configuration example requires five IP addresses:
  - An IP address that you provide to clients to access your Web site, www.Intersplashx.com (9.17.32.59)
  - An IP address for an interface (gateway) to Cisco CSS Switch (9.17.32.50)
  - An IP address for server A (9.17.32.51)
  - An IP address for server B (9.17.32.52)
  - An IP address for the Cisco CSS Controller server C (9.17.32.53)

How do you prepare?

Ensure the following steps are complete before beginning configuration for this example:

- Ensure that your Cisco CSS Switch is properly configured. For configuration information, refer to *Cisco Content Services Switch Getting Started Guide*.
- Ensure that the Cisco CSS Controller machine can ping the Cisco CSS Switch (9.17.32.50), server A (9.17.32.51), and server B (9.17.32.52).
- Ensure that the client machine can ping the VIP (9.17.32.59)

Configuring the Cisco CSS Controller component

With Cisco CSS Controller, you can create a configuration by using the command line or the graphical user interface (GUI). For this quick start example, configuration steps are demonstrated using the command line.

Note: The parameter values must be typed in English characters. The only exceptions are parameter values for host names and file names.

Configuring with the command line

From a command prompt, follow these steps:

1. Start the ccoserver on Load Balancer. As root user or administrator, issue the following from a command prompt: **ccoserver**
2. Add a switch consultant to the Cisco CSS Controller configuration, specifying the Cisco CSS Switch IP interface address and the read-write community name. These values must match the corresponding attributes on the Cisco CSS Switch:

   ccocontrol consultant add SwConsultant-1 address 9.17.32.50 community public

   This will check connectivity to Cisco CSS Switch and will verify that the SNMP read-write community name is working properly.
3. Add ownercontent (OwnerContent-1) to the switch consultant, specifying ownername (OwnerName-1) and contentrule (ContentRule-1):

   ccocontrol ownercontent add SwConsultant-1:OwnerContent-1 ownername OwnerName-1 contentrule ContentRule-1

   These values must match the corresponding attributes on the Cisco CSS Switch.

   Cisco CSS Controller can now communicate with the switch over SNMP and will obtain the necessary configuration information from the switch. After this step, you should see information in the Cisco CSS Controller about which services are configured on Cisco CSS Switch for the specified ownercontent.
4. Configure the type of metrics to collect (active connection, connection rate, HTTP) and proportion for each metric on the ownercontent:

   ccocontrol ownercontent metrics SwConsultant-1:OwnerContent-1 activeconn 45 connrate 45 http 10

   This command will configure which metric information and proportion you want to collect from the services to be used for weight calculation. The total proportion of all the metrics must equal 100.
5. Start the switch consultant function of Cisco CSS Controller:

   ccocontrol consultant start SwConsultant-1

   With this command, all the metric collectors will start, and service weight calculations will begin. Cisco CSS Controller communicates the results of its service weight calculations to Cisco CSS Switch using SNMP.

Your basic Cisco CSS Controller configuration is now complete.

Testing your configuration

Test to see if the configuration is working:

1. From the Client Web browser, go to location **http://www.Intersplashx.com**. If a page is displayed, the configuration is working.
2. Reload the page in the Web browser.

3. Look at the results of the following command: **ccocontrol service report SwConsultant-1:OwnerContent-1:Service-1**. The total connections column of the two Web servers should add up to "2."

Configuring with the graphical user interface (GUI)

For information on using the Cisco CSS Controller GUI, see "GUI" on page 115 and Appendix A, "GUI: General instructions," on page 403.

Chapter 15. Planning for Cisco CSS Controller

Note: The Cisco CSS Controller component is shipped with Version 8.0 of Load Balancer for IPv4, but this component might not support newer hardware. Consult the prerequisites page for supported hardware: *http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921*

This chapter describes what a network planner should consider before installing and configuring the Cisco CSS Controller component.

- See Chapter 16, "Configuring Cisco CSS Controller," on page 113 for information on configuring the load-balancing parameters of the Cisco CSS Controller component.
- See Chapter 22, "Advanced features for Cisco CSS Controller and Nortel Alteon Controller," on page 197 for information on how to set up Load Balancer for more advanced functions.
- See Chapter 23, "Operating and managing Load Balancer," on page 213 for information on remote authenticated administration, Load Balancer logs, and usage of the Load Balancer components.

This chapter includes:

- "System requirements"
- "Planning considerations"
  - "Placement of the consultant in the network" on page 108
  - "High availability" on page 110
  - "Calculating weights" on page 110
  - "Problem determination" on page 111

System requirements

For hardware and software requirements, refer to the following Web page:
`http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921`

You will also need

- A system on which to run the Cisco CSS Controller.
- An installed and configured Cisco CSS 11000 series content services switch

Planning considerations

The Cisco CSS Controller manages a set of switch consultants. Each consultant determines weights for services that are load balanced by a single switch. The switch for which the consultant provides weights is configured for content load balancing. The consultant uses the SNMP protocol to send the calculated weights to the switch. The switch uses the weights to select a service for the content rule it is load balancing when the load balancing algorithm is weighted round-robin. To determine weights, the consultant uses one or more of the following pieces of information:

- Availability and response times, determined through the use of application **advisors** that communicate with applications running on the service.
- System load information, determined by retrieving a metric value from **metric server agents** running on the service.

- Connection information about the service, obtained from the switch.
- Reachability information, obtained by pinging the service.

See the *Cisco Content Services Switch Getting Started Guide* for a description of content load balancing and for detailed information on configuring the switch.

For a consultant to obtain the information it needs to determine service weights, you must have:

- IP connectivity between the consultant and the services for which weights are calculated.
- IP connectivity between the consultant and the switch that is load balancing the servers for which weights are calculated.
- SNMP enabled on the switch. Both read and write capabilities must be enabled.

Placement of the consultant in the network

As indicated in Figure 23 on page 109, the consultant can be connected to the network behind the switch or switches for which it provides weights. Some parameters must be configured on the switch and some on the controller to enable connectivity between the controller, the switch, and the services.

In Figure 23 on page 109:

- A consultant is connected to the network behind the switches for which it is providing weights.
- The network consists of two VLANs.
- For the consultant to communicate with services in both VLANs, IP forwarding must be enabled on the interfaces through which the services are connected, and on the interface through which the consultant is connected.
- The IP address of the switch must be configured as the default gateway on the consultant and the service systems.

Refer to the *Cisco Content Services Switch Getting Started Guide* for detailed information about configuring VLANs and IP routing on the switch.

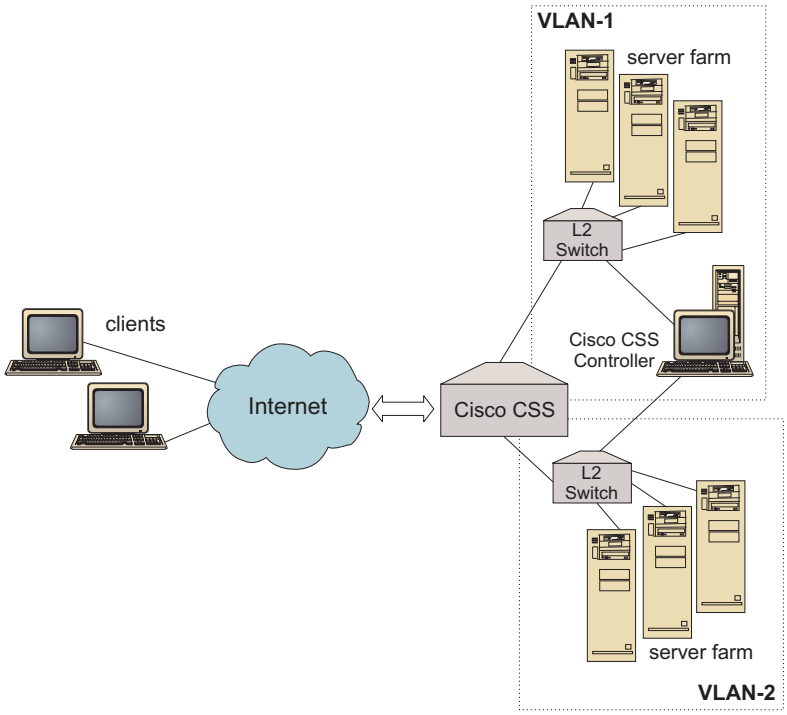


Figure 23. Example of a consultant connected behind the switches

You can manage the Cisco CSS Controller using any of the following interfaces:

- A browser
- A GUI (remote or local)
- A command line (remote or local)

For remote management, in Figure 24 on page 110 :

- The consultant is connected behind the switch for which it is providing weights.
- The user interface is running on a remote system in front of the switch.
- The switch must be configured to allow the remote system to communicate through it to the controller system.

Refer to *Cisco Content Services Switch Getting Started Guide* for detailed information.

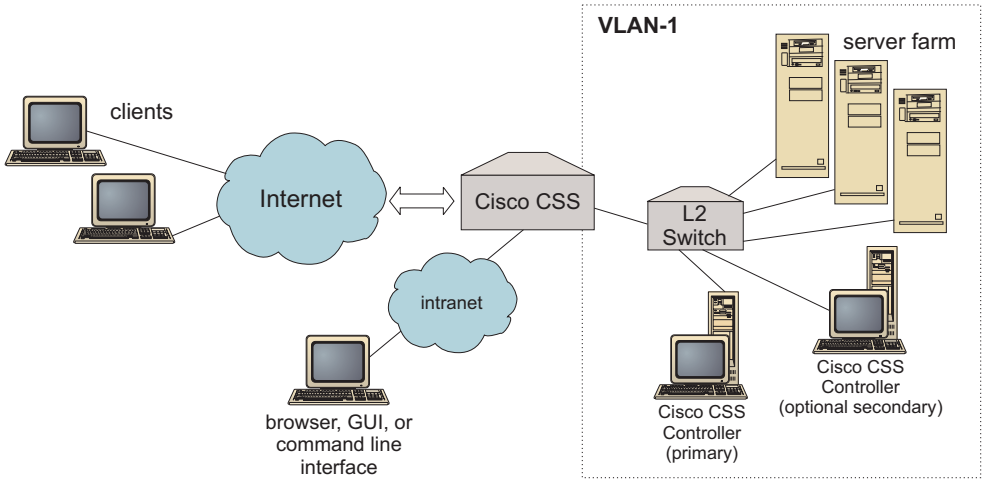


Figure 24. Example of consultant (with optional high availability partner), configured behind switch with user interface in front of switch

High availability

Controller high availability enhances the fault tolerance capabilities of Load Balancer. Designed with packet-forwarding high availability in mind, controller high availability involves two controllers running simultaneously, one in the primary role, the other in the secondary role.

Each controller is configured with identical switch information, and only one controller is active at a time. This means that, as determined by the high availability logic, only the active controller calculates and updates the switch with new weights.

Controller high availability communicates with its partner using simple user datagram protocol (UDP) packets over an address and port that you configure. These packets are used to exchange information between controllers as it pertains to high availability (reach information), and to determine partner controller availability (heartbeats). If the standby controller determines that the active controller has failed for any reason, the standby controller takes over from the failed active controller. The standby controller then becomes the active controller, and begins calculating and updating the switch with new weights.

In addition to partner availability, reach targets can be configured for high availability. Controller high availability uses the reach information to determine which controller is active and which is standby. The active controller is the controller that can ping more targets and is reachable from its partner.

See “High availability” on page 197 for more information.

Calculating weights

If the consultant determines that a service is unavailable, it will suspend that service on the switch to prevent the switch from considering the server when it load balances requests. When the service is available again, the consultant activates the service on the switch so that it is considered for load balancing requests.

Problem determination

Cisco CSS Controller posts entries to the following logs:

- server.log
- consultant.log
- highavailability.log
- metriccollector.log
- binary.log

These logs are located in the following directories:

- For AIX, HP-UX, Linux, and Solaris systems: **/opt/ibm/edge/lb/servers/logs/cco/***consultantName*
- For Windows systems: **<***install_root***>ibm\edge\lb\servers\logs\cco***consultantName***

In each log, you can set the log size and logging level. See "Using Load Balancer logs" on page 217 for more information.

Chapter 16. Configuring Cisco CSS Controller

Before following the steps in this chapter, see Chapter 15, "Planning for Cisco CSS Controller," on page 107. This chapter explains how to create a basic configuration for the Cisco CSS Controller component of Load Balancer.

- See Chapter 22, "Advanced features for Cisco CSS Controller and Nortel Alteon Controller," on page 197 for more complex configurations.
- See Chapter 23, "Operating and managing Load Balancer," on page 213 for information on remote authenticated administration, logs, and usage of the Cisco CSS Controller component.

Overview of configuration tasks

Before you begin any of the configuration methods in this chapter:

1. Ensure that your Cisco CSS Switch and all server machines are properly configured.
2. Configure Cisco CSS Controller, ensuring that Cisco CSS Switch's address and SNMP community name match the corresponding attributes on the Cisco CSS Switch. See "ccocontrol consultant — configure and control a consultant" on page 368 for information on configuring the consultant.

Table 8. Configuration tasks for the Cisco CSS Controller component

Task	Description	Related information
Set up the Cisco CSS Controller machine	Finding out about the requirements	"Setting up the Controller for Cisco CSS Switches machine" on page 116
Test your configuration	Confirming that the configuration is working	"Testing your configuration" on page 117

Methods of configuration

To create a basic configuration for the Cisco CSS Controller component of Load Balancer, there are three methods:

- Command line
- XML file
- Graphical user interface (GUI)

Command line

This method is the most direct means of configuring Cisco CSS Controller. The procedures in this manual assume use of the command line. The command parameter values must be entered in English characters. The only exceptions are host names (used, for example, in the **consultant add** command) and file names.

To start Cisco CSS Controller from the command line:

1. Issue the **ccoserver** command from the command prompt. To stop the server, type: **ccoserver stop**

Notes:

a. For Windows systems, click **Start** > **Control Panel** > **Administrative Tools** > **Services**. Right-click **IBM Cisco CSS Controller** and select **Start**. To stop the service, follow the same steps and select **Stop**.

b. On Windows systems, you can start the **ccoserver** automatically during boot:

   1) Click **Start** > **Settings** > **Control Panel** > **Administrative Tools** > **Services**.

   2) Right-click **IBM Cisco CSS Controller**, then select **Properties**.

   3) Click the arrow for the **Startup** type field, then select **Automatic**.

   4) Click **OK**.

2. Next, issue the Cisco CSS Controller control commands you want to set up your configuration. The procedures in this manual assume use of the command line. The command is **ccocontrol**. For more information about commands, see Chapter 28, "Command reference for Cisco CSS Controller," on page 367.

You can enter an abbreviated version of the ccocontrol command parameters. You only need to enter the unique letters of the parameters. For example, to get help on the file save command, you can type **ccocontrol he f** instead of **ccocontrol help file**.

To start up the command line interface: issue **ccocontrol** to receive an ccocontrol command prompt.

To end the command line interface: issue **exit** or **quit**.

Note: On Windows platforms, the Dispatcher component's dsserver starts automatically. If you are using only Cisco CSS Controller and not the Dispatcher component, you can stop dsserver from starting automatically as follows:

1. Form Windows Services, right-click **IBM Dispatcher**.
2. Select **Properties**.
3. In the **Startup type** field, select **Manual**.
4. Click **OK**, and close the Services window.

XML

The currently-defined configuration can be saved to an XML file. This enables the configuration to be loaded at a later time when you want to quickly recreate the configuration.

To run the content of an XML file (for example, **myscript.xml**), use either of the following commands:

- To save the current configuration into an XML file, issue the following command:

  ccocontrol file save *XMLFilename*

- To load a saved configuration, issue the following command:

  ccocontrol file load *XMLFileName*

  Use the load command only if you have previously done a **file save**.

The XML files are saved in the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/configurations/cco**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\configurations\cco**

GUI

For general instructions and an example of the graphical user interface (GUI), see Figure 38 on page 403.

To start the GUI, follow these steps

1. If ccoserver is not already running, start it now by running the following as root:

   ccoserver.
2. Next, do one of the following:
   - For AIX, HP-UX, Linux, or Solaris systems: enter **lbadmin**
   - For Windows systems: click **Start** > **Programs** > **IBM WebSphere** > **Edge Components** > **IBM Load Balancer** > **Load Balancer**

To configure the Cisco CSS Controller component from the GUI:

1. Right-click Cisco CSS Controller in the tree structure.
2. Connect to a Host.
3. Create one or more switch consultants containing the desired ownercontents and their associated metrics.
4. Start the consultant.

You can use the GUI to do anything that you would do with the **ccocontrol** command. For example:

- To define a consultant using the command line, type **ccocontrol consultant add** *consultantID* **address** *IPAddress* **community** *name* .
- To define a consultant from the GUI, right-click the Host node, then click **Add a switch consultant.** Type the switch address and community name in the pop-up window, then click OK.
- Use **Load Configuration** presented in the Host pop-up menu to load pre-existing Cisco CSS Controller configuration files and to append to the current configuration.
- Select **Save Configuration File As** to periodically save your Cisco CSS Controller configuration to a file.
- Select **File** from the menu bar to save your current host connections to a file or to restore connections in existing files across all Load Balancer components.

To run a command from the GUI:

1. Right-click the **Host** node and select **Send command...**
2. In the Command entry field, type the command that you want to run; for example, **consultant report**.
3. Click Send.

The results and history of the commands that you run in the current session appear in the Result box.

To access **Help** click the question mark icon in the upper right corner of the Load Balancer window.

- **Help: Field level —** describes each field, default values
- **Help: How do I —** lists tasks that can be done from that screen
- **InfoCenter —** provides centralized access to product information

For more information about using the GUI, see Appendix A, "GUI: General instructions," on page 403.

Setting up the Controller for Cisco CSS Switches machine

Before setting up the Cisco CSS Controller machine, you must be the root user (on AIX, HP-UX, Linux , or Solaris systems) or the Administrator (on Windows systems).

Consultant must be able to connect to the Cisco CSS Switch as a Cisco CSS Switch administrator.

When configuring the consultant, you must configure the address and SNMP community name to match the corresponding attributes on the Cisco CSS Switch.

For help with commands used in this procedure, see Chapter 28, "Command reference for Cisco CSS Controller," on page 367.

Step 1. Start the server function

If the ccoserver is not already running, type **ccoserver** as root to start it now.

Note: For Windows systems, click **Start** > **Control Panel** > **Administrative Tools** > **Services**. Right-click IBM Cisco Controller and select Start.

Step 2. Start the command line interface

Type **ccocontrol** to start the command line interface.

Step 3. Configure the consultant

You must configure the switch address and SNMP community name. These values must match the corresponding attributes on the Cisco CSS Switch.

To add a consultant, type:

```
consultant add switchConsultantID address switchIPAddress
community communityName
```

Step 3. Configure an ownercontent

An ownercontent is a representation of a content rule for an owner, which is defined on the Cisco CSS Switch. The owner name and content rule name must match how it is defined on the switch.

To define an ownercontent, type:

```
ownercontent add switchConsultantID:ownercontentID ownername ownerName
contentrule contentRuleName
```

Step 4. Verify that services are correctly defined

When the ownercontent is defined, the consultant completes the configuration by retrieving the services configured on the switch. Compare the configuration on the switch with the configuration for the consultant to ensure that the services match.

Step 5. Configure metrics

Metrics are the measurements used to determine the service weights and associated proportions (importance of one metric compared to another), and can be any combination of connection data metrics, application advisor metrics, and metric server metrics. The proportions must always total 100.

When the ownercontent is configured, the default metrics are defined as **activeconn** and **connrate**. If you want additional metrics, or if you want metrics that are altogether different from the defaults, type:

ownercontent metrics *switchConsultantID*:*ownercontentID metric1 proportion1 metric2 proportion2...metricN proportionN*

Step 6. Start the consultant

To start the consultant, type:

consultant start *switchConsultantID*

This starts the metric collectors, and weight calculation begins.

Step 7. Start the Metric Server (optional)

If system metrics are defined in Step 5, the metric server must be started on the service machines. See "Metric Server" on page 157 for information on using the metric server.

Step 8. Configure high availability (optional)

To configure high availability, type:

highavailability add address *IPaddress* **partneraddress** *IPaddress* **port** *80* **role** *primary*

In a high availability environment, you can configure multiple switches. To ensure that weight information is always available when one switch takes over for another switch, the Cisco CSS Controller must be configured to provide weights for all switches and their backups.

See Chapter 22, "Advanced features for Cisco CSS Controller and Nortel Alteon Controller," on page 197 for detailed information on how to use and configure controller high availability.

Testing your configuration

Test to see if the configuration is working:

1. Set the consultant loglevel to 4.
2. Disconnect a server from the Cisco CSS Switch for one minute, or shut down the application server for one minute.
3. Reconnect the server, or restart the application server.
4. Set the consultant loglevel back to the desired level (1).
5. View the consultant.log file located in the following directories, and look for **setServerWeights setting service**:
   - For AIX, HP-UX, Linux, and Solaris systems: **/opt/ibm/edge/lb/servers/logs/cco/***consultantName*
   - For Windows systems: **<***install_root***>ibm\edge\lb\servers\logs\cco***consultantName*

Part 6. Nortel Alteon Controller component

This part provides information on a quick start configuration, planning considerations, and describes the methods of configuring Load Balancer's Nortel Alteon Controller component. It contains the following chapters:

Chapter 17. Quick start configuration

This quick start example shows how to create a configuration using the Nortel Alteon Controller component. Nortel Alteon Controller provides server weights to the Nortel Alteon Web Switch. These weights are used to select servers for services that the switch is load balancing.

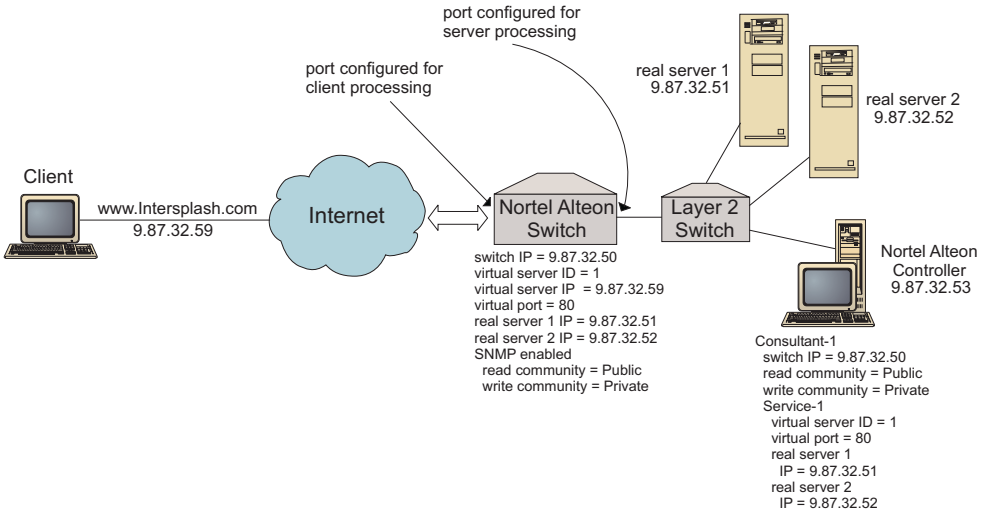


Figure 25. A simple Nortel Alteon Controller configuration

What do you need?

For this quick start configuration example, you will need the following:

- A Nortel Alteon Web Switch, running Web OS version 9.0 or version 10.0
- A server machine with the Nortel Alteon Controller component
- Two Web server machines
- A Layer 2 Switch connected to a port on the Nortel Alteon Web Switch

  Note: If a Layer 2 Switch is not used, the Nortel Alteon Controller machine and the Web server machines can be connected directly to ports on Nortel Alteon Web Switch.
- This configuration example requires five IP addresses:
  - An IP address that you provide to clients to access your Web site, www.Intersplashx.com (9.87.32.59)
  - An IP address for an interface configured to the Nortel Alteon Web Switch (9.87.32.50)
  - An IP address for real server 1 (9.87.32.51)
  - An IP address for real server 2 (9.87.32.52)
  - An IP address for the Nortel Alteon Controller (9.87.32.53)

How do you prepare?

Ensure the following steps are complete before beginning configuration for this example:

- Ensure that the Nortel Alteon Web Switch is properly configured. (For more complete configuration information, refer to your Nortel Alteon Web OS Application Guide):
  - Enable layer 4 server load balancing on the switch.
  - Configure an IP interface (9.87.32.50) on Nortel Alteon Web Switch
  - Enable SNMP on Nortel Alteon Web Switch
  - Enable server load-balancing client processing on the Nortel Alteon Web Switch port that receives client requests.
  - Enable server load-balancing server processing on the Nortel Alteon Web Switch port that the servers are connected.
  - Configure the default gateway to be the switch IP interface (9.87.32.50) on real server 1, real server 2, and Nortel Alteon Controller.
  - Configure Nortel Alteon Web Switch with real server 1 and real server 2.
  - Configure Nortel Alteon Web Switch with a Server Group comprised of real server 1and real server 2. Assign the group an ID of 1.
  - Configure Nortel Alteon Web Switch with a Virtual Server. The virtual server IP address is 9.87.32.59. Assign an ID of 1 to the virtual server.
  - Configure Nortel Alteon Web Switch with a service that uses virtual port 80 and is serviced by group 1.
- Ensure that the client machine can ping the Virtual Server IP address 9.87.32.59.
- Ensure that the Nortel Alteon Controller machine can ping the Nortel Alteon Web Switch IP interface (9.87.32.50), real server 1 (9.87.32.51), and real server 2 (9.87.32.52).

Configuring the Nortel Alteon Controller component

With Nortel Alteon Controller, you can create a configuration by using the command line or the graphical user interface (GUI). For this quick start example, configuration steps are demonstrated using the command line.

Note: The parameter values must be typed in English characters. The only exceptions are parameter values for host names and file names.

Configuring with the command line

From a command prompt, follow these steps:

1. Start the nalserver on Nortel Alteon Controller. As root user or administrator, issue the following from a command prompt: **nalserver**
2. Add a consultant to the Nortel Alteon Controller configuration, specifying the Nortel Alteon Web Switch IP interface address. (Only specify the read community and write community if it is different from the default (public, private):

   nalcontrol consultant add Consultant-1 address 9.87.32.50

   This will check connectivity to Nortel Alteon Web Switch and will verify that the SNMP community names are working properly.
3. Add a service (Service-1) to the consultant (Consultant-1), specifying the virtual server identifier (1) and virtual port number (80) for the service:

   nalcontrol service add Consultant-1:Service-1 vsid 1 vport 80

Nortel Alteon Controller will communicate with the switch over SNMP and will obtain the necessary configuration information from the switch. After this step, you should see information in the Nortel Alteon Controller about what servers are configured on Nortel Alteon Web Switch for the service.

4. Configure the metrics to be collected for the set of servers associated with the service:

   nalcontrol service metrics Consultant-1:Service-1 http 40 activeconn 30 connrate 30

   This command will configure which metric information you want to collect from the servers and the relative importance of those metrics during weight calculation.

5. Start the consultant function of Nortel Alteon Controller:

   nalcontrol consultant start Consultant-1

   With this command, all the metric collectors will start, and server weight calculations will begin. Nortel Alteon Controller communicates the results of its server weight calculations to Nortel Alteon Web Switch using SNMP.

Your basic Nortel Alteon Controller configuration is now complete.

Testing your configuration

Test to see if the configuration is working:

1. From the Client Web browser, go to location **http://www.Intersplashx.com**. If a page is displayed, the configuration is working.
2. Reload the page in the Web browser.
3. Look at the results of the following command: **nalcontrol service report Consultant-1:Service-1**. The total connections column of the two Web servers should add up to "2."

Configuring with the graphical user interface (GUI)

For information on using the Nortel Alteon Controller GUI, see "GUI" on page 134 and Appendix A, "GUI: General instructions," on page 403.

Chapter 18. Planning for Nortel Alteon Controller

Note: The Nortel Alteon Controller component is shipped with Load Balancer for IPv4, but this component might not support newer hardware. Consult the prerequisites page for supported hardware: *http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921*

This chapter describes what a network planner should consider before installing and configuring the Nortel Alteon Controller component.

- See Chapter 19, "Configuring Nortel Alteon Controller," on page 133 for information on configuring the load-balancing parameters of the Nortel Alteon Controller component.
- See Chapter 22, "Advanced features for Cisco CSS Controller and Nortel Alteon Controller," on page 197 for information on how to configure advisors and metric servers.
- See Chapter 23, "Operating and managing Load Balancer," on page 213 for information on remote authenticated administration, Load Balancer logs, and usage of the Load Balancer components.

This chapter includes:

- "System requirements"
- "Planning considerations"
  - "Placement of the consultant in the network" on page 126
  - "Server attributes on the switch (set by the controller)" on page 128
  - "Configuring backup servers" on page 128
  - "Configuring groups" on page 129
  - "High availability" on page 130
  - "Tuning" on page 131
  - "Problem determination" on page 132

System requirements

For hardware and software requirements, refer to the following Web page:
`http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921`

You will also need

- A system on which to run the Nortel Alteon Controller.
- An installed and configured Nortel Alteon Web Switch. The Web switch hardware platforms are AD3, AD4, 180e 184, and the layer 4/7 blade for the Passport 8600.

Planning considerations

The Nortel Alteon Controller manages a set of switch consultants. Each consultant determines weights for servers that are load balanced by a single switch. The switch for which the consultant provides weights is configured for server load balancing. The consultant uses the SNMP protocol to send the calculated weights to the switch. The switch uses the weights to select a server for the service it is load balancing. To determine weights, the consultant uses one or more of the following pieces of information:

- Availability and response times, determined through the use of **advisors** that communicate with applications running on the servers.
- System load information, determined by retrieving a metric value from **metric server agents** running on the servers.
- Connection information about the servers, obtained from the switch.
- Reachability information, obtained by pinging the servers.

See your Nortel Alteon Web OS Application Guide for a description of server load balancing and for detailed information on configuring the switch.

For a consultant to obtain the information it needs to determine server weights, you must have:

- IP connectivity between the consultant and the servers for which weights are calculated.
- IP connectivity between the consultant and the switch that is load balancing the servers for which weights are calculated.
- SNMP enabled on the switch. Both read and write capabilities must be enabled.

Placement of the consultant in the network

The consultant can be connected to the network in front of or behind the switch or switches for which it provides weights. Some parameters must be configured on the switch and some on the controller to enable connectivity between the controller, the switch, and the servers.

In Figure 26 on page 127:

- A consultant is connected to the network behind the switches for which it is providing weights.
- The network consists of two VLANs.
- For the consultant to communicate with servers in both VLANs, IP forwarding must be enabled on the interfaces through which the servers are connected, and on the interface through which the consultant is connected.
- The IP address of the switch must be configured as the default gateway on the consultant and the server systems.

Refer to your Nortel Alteon Web OS Application Guide or Command Reference for detailed information about configuring VLANs and IP routing on the switch.

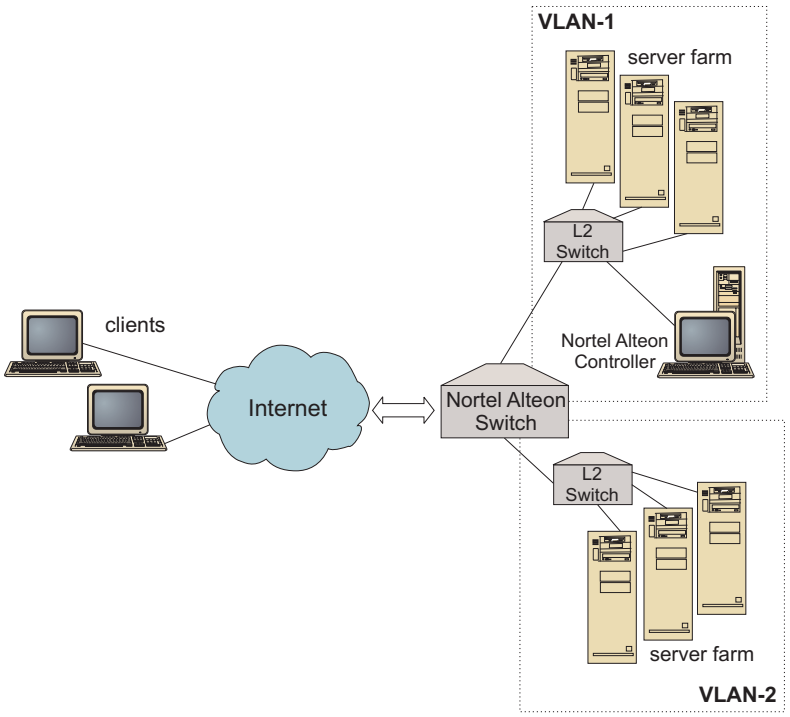


Figure 26. Example of a consultant connected behind the switch

In Figure 27:

- The consultant is connected to the switch through an intranet in front of the switch.
- Server load balancing direct access mode must be enabled on the switch to allow the consultant to communicate with the switch and with the servers.
- With server load balancing direct access mode enabled, any client can send traffic directly to any server. To limit direct server access to only the consultant, you can specify load balancing *mnet* and *mmask* to the switch. Refer to your Nortel Alteon Web OS Application Guide or Command Reference for detailed information on configuring server load balancing and on direct server interaction.

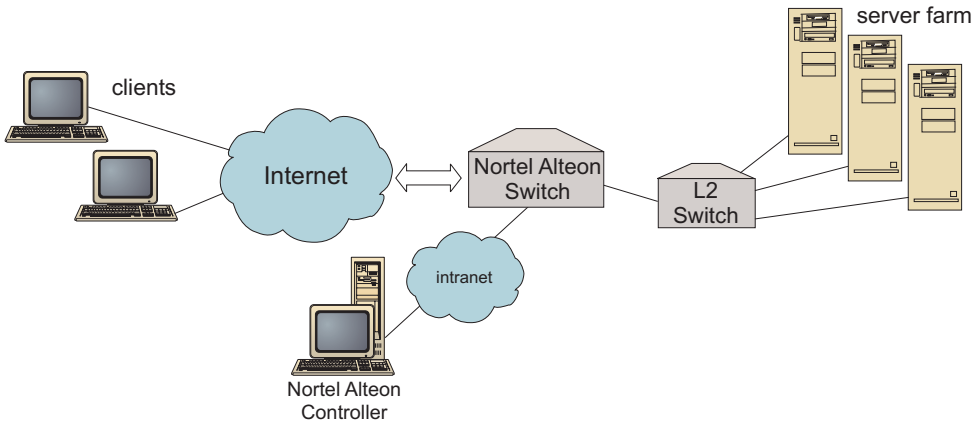


Figure 27. Example of consultant connected through an intranet in front of switch

You can manage the Nortel Alteon Controller using any of the following interfaces:

- A browser
- A GUI
- A remote command line

In Figure 28:

- The consultant is connected behind the switch for which it is providing weights.
- The user interface is running on a remote system in front of the switch.
- The network must be configured so that the user interface is able to communicate with the controller.

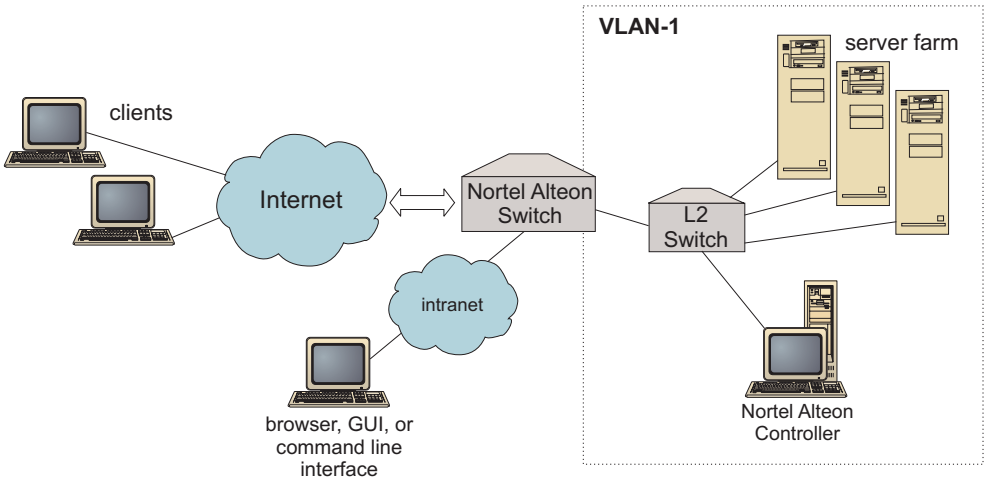


Figure 28. Example of consultant behind switch and user interface in front of switch

Server attributes on the switch (set by the controller)

When a consultant calculates weights for servers that provide a service that is load balanced by a switch, the consultant disables the normal server health checking at the switch to reduce unnecessary traffic to the servers. The consultant re-enables the health checking when it stops providing weights for the service. The server health check interval corresponds to MIB variable slbNewCgRealServerPingInterval.

If the consultant determines that a server is unavailable, the consultant sets the server's maximum number of connections to zero to prevent the switch from considering the server when it load balances requests. When the server is available again, the maximum number of connections is restored to its original value. The server maximum connections value corresponds to MIB variable slbNewCfgRealServerMaxCons.

When a weight is calculated for a real server, the weight is set for the server. The server weight value corresponds to MIB variable slbNewCfgRealServerWeight.

Configuring backup servers

The switch allows the configuration of some servers as backups to others. If the switch determines that a server that has a backup is unavailable, the switch might start sending requests to the backup. When the consultant calculates weights for a service with a backup, it calculates weights for both the backup and the primary servers, and subsequently has weights to use for server selection when the backup is required.

The weight for a backup server might be higher than the weight for a primary server. This is because no requests are forwarded to it, so it has low loads until the switch decides to use it.

To avoid idle server resources, it is common practice that servers assigned to one service be used as backups for servers assigned to a different service. When implementing a configuration like this, avoid assigning the same real servers to multiple concurrently-active services. If this occurs, the weight for the server is overwritten by the consultant for each service in which the server is a part.

Each real server is identified by an integer and has a weight and IP address attribute. Two real servers might have the same IP address. In this case, two real servers are associated with the same physical server machine. The real servers identified as backups should only be configured as backups for a single service. If the same physical server machines will backup servers assigned to multiple services, they must be configured once for each service and be given a server identification that is unique for each service. This allows the backups to have a unique weight assigned to them for each service they are backing up.

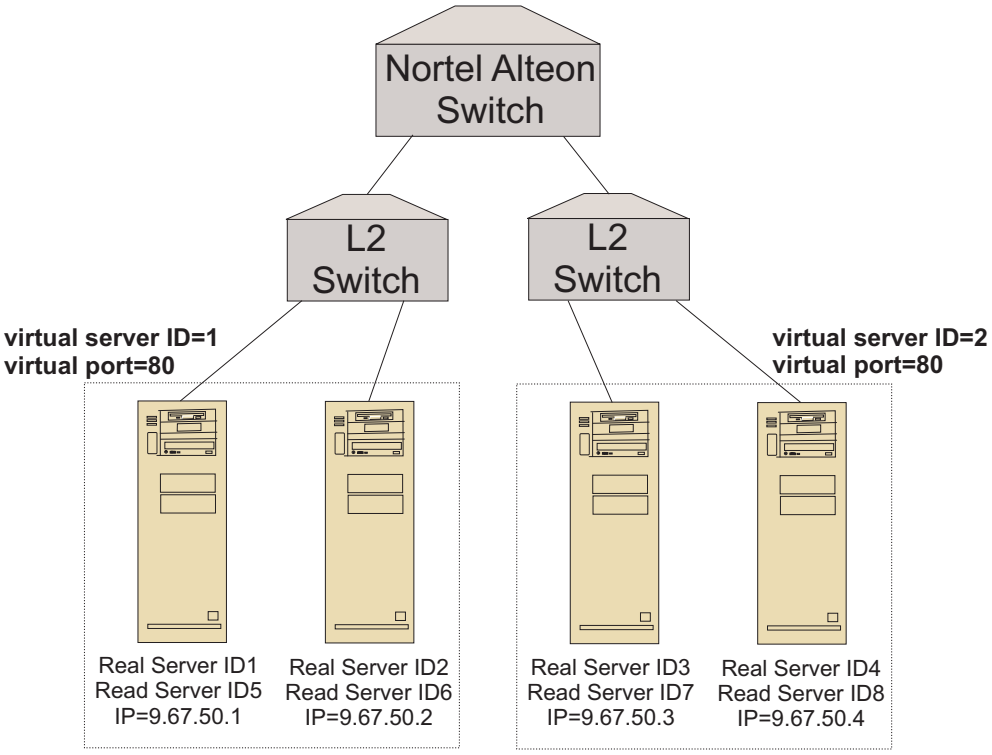


Figure 29. Example of consultant configured with backup servers

Configuring groups

Servers on a switch can be configured as part of multiple groups, and groups on the switch can be configured to service multiple services.

Because it is possible to configure the same server for multiple services, the weight is calculated for each service in which the server is a part. It is possible, therefore, for the weight to be incorrect because it is unknown at any time for which service the weight is intended.

In addition, if the consultant is determining weights for one service and not for another, it is possible that the service that the consultant is not calculating weights for has server health checking disabled. In this case, the switch might not properly load balance that service.

Because of these possibilities, you must ensure that a real server is not assigned to multiple services that are being load balanced. This does not mean that the same server machine cannot be servicing requests for multiple services. It means that a real server with a unique identifier must be configured on the switch for each service that the server machine will handle requests for.

High availability

Both the Nortel Alteon Controller and the Nortel Alteon Web Switch have high availability capabilities.

You can configure two controllers to run on different systems in a hot-standby configuration.

Two or more switches can back each other up when you configure them to act as a virtual IP interface router (VIR) or as a virtual IP server router (VSR).

One consultant (managed by the controller) provides weights for only one switch. Because a backup switch might take over for the master, you must configure the controller with one consultant for each switch that has the possibility of becoming master. In this way, when a switch becomes master, it is ensured of being provided with weights.

In addition, when the controllers are connected to a VIR, they are ensured of communication with the servers, the switches, and the backup controller, should it lose connectivity to one of the switches.

Refer to your Nortel Alteon Web OS Application Guide for information about high availability on the switch.

Controller high availability enhances the fault tolerance capabilities of Load Balancer. Designed with classic packet-forwarding high availability in mind, controller high availability involves two controllers running simultaneously, one in the primary role, the other in the secondary role.

Each controller is configured with identical switch information. Similar to classic high availability, only one controller is active at a time. This means that, as determined by the high availability logic, only the active controller calculates and updates the switch with new weights.

Controller high availability communicates with its partner using simple user datagram protocol (UDP) packets over an address and port that you configure. These packets are used to exchange information between controllers as it pertains to high availability (reach information), and to determine partner controller availability (heartbeats). If the standby controller determines that the active controller has failed for any reason, the standby controller takes over from the failed active controller. The standby controller then becomes the active controller, and begins calculating and updating the switch with new weights.

In addition to partner availability, reach targets can be configured for high availability. As with classic high availability, controller high availability uses the

reach information to determine which controller is active and which is standby. The active controller is the controller that can ping more targets and is reachable from its partner.

See “High availability” on page 197 for more information.

In Figure 30:

- Two Nortel Alteon Controllers are connected behind switches.
- One controller is primary and is actively providing the switches with server weights; the other controller is backup.
- The controllers must have TCP/IP communication for the backup to know when it should take primary responsibility.
- Two Nortel Alteon Web Switches are configured, as a VIR and a VSR.
- The VIR provides high availability for connections to the servers.
- The VSR provides high availability for access to the virtual servers configured on the switches.
- One of the switches is master and the other is backup.
- The primary controller is providing weights for both switches.
- The backup controller is sending heartbeats to the primary to determine when to take over.

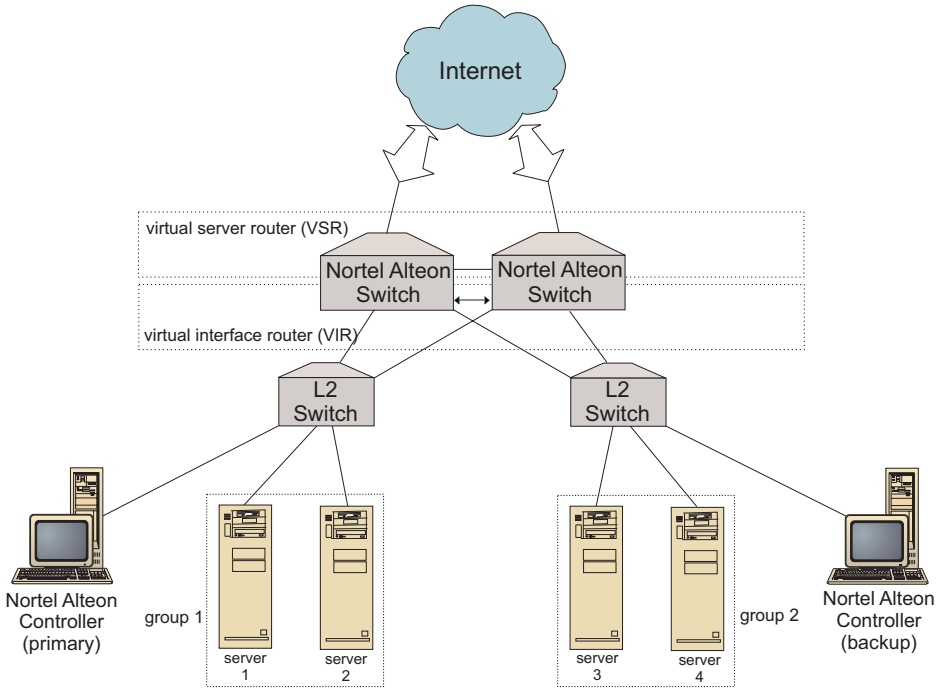


Figure 30. Example of Nortel Alteon Controller and Nortel Alteon Web Switch high availability

Tuning

To avoid changing weights too often, you can configure the consultant with a sensitivity threshold. The sensitivity threshold specifies the amount of change that must take place between the old and new weights before the weight can change. See “Sensitivity threshold” on page 201 for more information.

If the switch becomes too busy updating weights, you can increase the consultant sleeptime to reduce the traffic between the controller and the servers and the switch. Sleeptime sets the number of seconds to sleep between weight-setting cycles.

If the servers are handling too many monitoring requests from the consultant, you can modify the metric collectors' sleeptime. See "Weight calculation sleeptimes" on page 201 for a detailed description.

Problem determination

Cisco CSS Controller posts entries to the following logs:

- server.log
- consultant.log
- highavailability.log
- metriccollector.log
- binary.log

These logs are located in the following directories:

- For AIX, HP-UX, Linux , and Solaris systems: **/opt/ibm/edge/lb/servers/logs/nal/***consultantName*
- For Windows systems: **<***install_root***>ibm\edge\lb\servers\logs\nal***consultantName*

In each log, you can set the log size and logging level. See "Using Load Balancer logs" on page 217 for more information.

Chapter 19. Configuring Nortel Alteon Controller

Before following the steps in this chapter, see Chapter 18, "Planning for Nortel Alteon Controller," on page 125. This chapter explains how to create a basic configuration for the Nortel Alteon Controller component of Load Balancer.

- See Chapter 22, "Advanced features for Cisco CSS Controller and Nortel Alteon Controller," on page 197 form more complex configurations.
- See Chapter 23, "Operating and managing Load Balancer," on page 213 for information on remote authenticated administration, logs, and usage of the Nortel Alteon Controller component.

Overview of configuration tasks

Before you begin any of the configuration methods in this chapter, ensure that your Nortel Alteon Web Switch and all server machines are properly configured.

Table 9. Configuration tasks for the Nortel Alteon Controller component

Task	Description	Related information
Configure the Nortel Alteon Web Switch and the servers	Configuring the switch.	Configure the switch, on page 136
Set up the Nortel Alteon Controller machine	Configuring the controller.	"Step 1. Start the server function" on page 136
Test your configuration	Confirm that the configuration is working	"Testing your configuration" on page 137

Methods of configuration

To create a basic configuration for the Nortel Alteon Controller component of Load Balancer, there are three methods:

- Command line
- XML file
- Graphical user interface (GUI)

Command line

This is the most direct means of configuring Nortel Alteon Controller. The procedures in this manual assume use of the command line.

To start Nortel Alteon Controller from the command line:

1. Issue the **nalserver** command from the command prompt. To stop the service, type: **nalserver stop**

   Notes:

   a. For Windows systems, click **Start** > **Control Panel** > **Administrative Tools** > **Services**. Right-click IBM Nortel Alteon Controller and select Start. To stop the service, follow the same steps and select Stop.

   b. For Windows systems, you can start the nalserver automatically during boot:

      1) Click **Start** > **Control Panel** > **Administrative Tools** > **Services**.

      2) Right-click IBM Nortel Alteon Controller, then select Properties.

3) Click the arrow for the Startup type field, then select Automatic.
4) Click OK.

2. Next, issue the Nortel Alteon Controller control commands you want to set up your configuration. The procedures in this manual assume use of the command line. The command is **nalcontrol**. For more information about commands, see Chapter 29, "Command reference for Nortel Alteon Controller," on page 385.

You can use an abbreviated version of the nalcontrol command parameters by typing the unique letters of the parameters. For example, to get help on the file save command, you can type **nalcontrol he f** instead of **nalcontrol help file**.

To end the command line interface: type **exit** or **quit**.

Notes:

1. You must use English characters for all command parameter values. The only exceptions are host names (used in server commands) and file names (used in file commands).
2. On Windows systems, the Dispatcher component's dsserver starts automatically. If you are using only Nortel Alteon Controller and not the Dispatcher component, you can prevent ndserver from starting automatically as follows:
   a. In Windows Services, right-click IBM Dispatcher.
   b. Select Properties.
   c. In the **Startup type** field, select Manual.
   d. Click OK, and close the Services window.

XML

The currently-defined configuration can be saved to an XML file. This enables the configuration to be loaded at a later time when you want to quickly recreate the configuration.

To run the content of an XML file (for example, **myscript.xml**), use the following commands:

- To save the current configuration into an XML file, issue the following command:

  nalcontrol file save *XMLFilename*

  Use the load command only if you have previously done a **file save**.
- To load a saved configuration, issue the following command:

  nalcontrol file load *XMLFileName*

  Use the load command only if you have previously done a **file save**.

The XML files are saved in the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/configurations/nal**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\configurations\nal**

GUI

For an example of the graphical user interface (GUI), see Figure 38 on page 403.

To start the GUI:

1. If nalserver is not already running, start it now by typing **nalserver** as root.

2. Next, do one of the following:
   - For AIX, HP-UX, Linux , or Solaris systems: type **lbadmin**
   - For Windows systems: click Start > **Programs** > **IBM WebSphere** > **Edge Components** > **IBM Load Balancer** > **Load Balancer**

To configure the Nortel Alteon Controller component from the GUI:

1. Right-click Nortel Alteon Controller in the tree structure.
2. Connect to a Host.
3. Create one or more switch consultants containing the desired services and their associated metrics.
4. Start the consultant.

You can use the GUI to do anything that you would do with the **nalcontrol** command. For example:

- To define a reach target using the command line, type **nalcontrol highavailability usereach** *address*. To define a reach target from the GUI, right-click High Availability > Add Reach Target.... Type the reach address in the pop-up window, then click OK.
- Use **Load Configuration** presented in the Host pop-up menu to append the configuration stored in a file to the running configuration. If you want to load a *new* configuration, you must stop and restart the server before you load the file.
- Right-click the Host node, then select **Save Configuration File As** to periodically save your Nortel Alteon Controller configuration to a file.
- Select **File** from the menu bar to save your current host connections to a file or to restore connections in existing files across all Load Balancer components.

To run a command from the GUI:

1. Right-click the **Host** node and select **Send command....**
2. In the Command entry field, type the command that you want to run; for example, **consultant report**.
3. Click Send.

The results and history of the commands that you run in the current session appear in the Result box.

To access Help, click the question mark icon in the upper right corner of the Load Balancer window.

- **Help: Field level —** describes each field, default values
- **Help: How do I —** lists tasks that can be done from that screen
- **InfoCenter —** provides centralized access to product information

For more information about using the GUI, see Appendix A, “GUI: General instructions,” on page 403.

Setting up Nortel Alteon Controller

For help with commands used in this procedure, see Chapter 29, “Command reference for Nortel Alteon Controller,” on page 385.

Before setting up the Nortel Alteon Controller machine:

- You must be the root user (on AIX, HP-UX, Linux , and Solaris systems) or the Administrator (on Windows systems).

- Nortel Alteon Controller must have IP connectivity to a Nortel Alteon Web Switch and to all servers for which weights are calculated.
- The Nortel Alteon Web Switch must be configured as follows:
  1. Enable layer 4 server load balancing on the switch.
  2. Configure an IP interface.
  3. Enable SNMP.
  4. Enable server load-balancing client processing on the port that receives client requests.
  5. Enable server load-balancing server processing on the port that the real servers are connected through.
  6. Configure real servers for the Web server machines.
  7. Configure a real server group consisting of the real servers that are running the application server.
  8. Configure a virtual server.
  9. Configure a service on a virtual port and assign the real server group to service it.

Step 1. Start the server function

If the nalserver is not already running, type **nalserver** as root to start it now.

Note: For Windows systems, click **Start** > **Control Panel** > **Administrative Tools** > **Services**. Right-click IBM Nortel Alteon Controller and select Start.

Step 2. Start the command line interface

Type **nalcontrol** to start the command line interface.

Step 3. Define a Nortel Alteon Web Switch consultant

To add a switch consultant, type:

consultant add *switchconsultantID* **address** *switchIPAddress*

Step 4. Add a service to the switch consultant

To add a service, type:

service add *switchConsultantID*:*serviceID* **vsid** *virtualServerID* **vport** *virtualPortNumber*

A service is identified by a virtual server identifier (VSID) and a virtual port (VPORT) number, both of which are associated with a virtual server previously configured on the switch.

Step 5. Configure metrics

Metrics is the information used to determine the server weights. Each metric is assigned a proportion to indicate its importance relative to other metrics. Any combination of metrics can be configured: connection data metrics, application advisor metrics, and metric server metrics. The proportions must always total 100.

When a service is configured, the default metrics are defined as **activeconn** and **connrate**. If you want additional metrics, or if you want metrics that are altogether different from the defaults, type:

service metrics *switchConsultantID*:*serviceID metricName* **50** *metricName2* **50**

Step 6. Start the consultant

To start the consultant, type:

```
consultant start switchConsultantID
```

This starts the metric collectors, and weight calculation begins.

Step 7. Configure high availability (optional)

To configure high availability, type:

```
highavailability add address IPaddress partneraddress IPaddress port 80
role primary
```

SeeChapter 22, “Advanced features for Cisco CSS Controller and Nortel Alteon Controller,” on page 197 for detailed information on how to use and configure controller high availability.

Step 8. Start the Metric Server (optional)

If system metrics are defined in Step 5, the metric server must be started on the service machines. See “Metric Server” on page 206 for information on using the metric server.

Step 9. Refreshing the Nortel Alteon Controller configuration

If you modify the configuration on the Nortel Alteon Web Switch, you can refresh the controller configuration. Type:

```
service refresh
```

Before you do a refresh of the configuration, stop the consultant. After the refresh command updates the configuration, restart the consultant.

Testing your configuration

Test to see if the configuration is working:

1. Set the consultant loglevel to 4.
2. Disconnect a server from the Nortel Alteon Web Switch for one minute, or shut down the application server for one minute.
3. Reconnect the server, or restart the application server.
4. Set the consultant loglevel back to the desired level (1).
5. View the consultant.log file located in the following directories, and look for **setServerWeights setting service**. This means that an attempt has been made to send weights to the switch.
   - For AIX, HP-UX, Linux, and Solaris systems: **/opt/ibm/edge/lb/servers/logs/cco/***consultantName*
   - For Windows systems: **<***install_root***>ibm\edge\lb\servers\logs\cco***consultantName*
6. Display server weights on the switch and verify that these weights match the weights shown on the controller report.

Part 7. Functions and advanced features for Load Balancer

This part provides information on functions and advanced configuration features that are available for Load Balancer. It contains the following chapters:

Chapter 20. Manager, Advisors, and Metric Server functions for Dispatcher, CBR, and Site Selector

This chapter explains how to configure the load balancing parameters and how to set up the manager, advisors, and Metric Server functions of Load Balancer.

Note: When reading this chapter, if you are *not* using the Dispatcher component, then substitute "dscontrol" with the following:

- For CBR, use **cbrcontrol**
- For Site Selector, use **sscontrol** (see Chapter 27, "Command reference for Site Selector," on page 341)

Table 10. Advanced configuration tasks for Load Balancer

Task	Description	Related information
Optionally, change load-balancing settings	You can change the following load-balancing settings: • Proportion of importance given to status information The default ratio is 50-50-0-0. If you use the default, information from advisors, Metric Server, and WLM are not used. • Weights • Manager fixed weights • Manager intervals • Sensitivity threshold • Smoothing index	"Optimizing the load balancing provided by Load Balancer" on page 142
Use scripts to generate an alert or record server failure when manager marks server(s) down or up	Load Balancer provides user exits that trigger scripts that you can customize when the manager marks server(s) down or up	"Using scripts to generate an alert or record server failure" on page 145
Use advisors	Describes and lists the advisors, which report on specific statuses of your servers	"Advisors" on page 146
Use HTTP or HTTPS advisor request and response (URL) option	Define a unique client HTTP URL string, specific for a service that you want to query on the machine	"Configuring the HTTP or HTTPS advisor using the request and response (URL) option" on page 152
Use self advisor	Provides backend server load status in a Load Balancer two-tiered WAN configuration	"Using Self Advisor in a two-tiered WAN configuration" on page 153
Create custom advisors	Describes how to write your own custom advisors	"Create custom (customizable) advisors" on page 154
Use Metric Server agent	Metric Server provides system load information to Load Balancer	"Metric Server" on page 157
Use Workload Manager advisor (WLM)	WLM advisor provides system load information to Load Balancer	"Workload Manager advisor" on page 159

Optimizing the load balancing provided by Load Balancer

The manager function of Load Balancer performs load balancing based on the following settings:

- "Proportion of importance given to status information"
- "Weights" on page 143
- "Manager intervals" on page 144
- "Advisor intervals" on page 148
- "Advisor report timeout" on page 148
- "Sensitivity threshold" on page 145
- "Smoothing index" on page 145

You can change these settings to optimize load balancing for your network.

Proportion of importance given to status information

The manager can use some or all of the following external factors in its weighting decisions:

- *Active connections*: The number of active connections on each load balanced server machine (as tracked by the executor). This proportion does not apply to Site Selector.

  Or —

  CPU: The percentage of CPU in use on each load balanced server machine (input from Metric Server agent). For Site Selector only, this proportion appears in place of the active connection proportion column.
- *New connections*: The number of new connections on each load balanced server machine (as tracked by the executor). This proportion does not apply to Site Selector.

  Or —

  Memory: The percentage of memory in use (input from Metric Server agent) on each load balanced server. For Site Selector only, this proportion appears in place of the new connection proportion column.
- *Port-specific*: The input from advisors listening on the port.
- *System metric*: The input from the system monitoring tools, such as Metric Server or WLM.

Along with the current weight for each server and some other information required for its calculations, the manager gets the first two values (active and new connections) from the executor. These values are based on information that is generated and stored internally in the executor.

Note: For Site Selector, the manager obtains the first two values (CPU and memory) from Metric Server.

You can change the relative proportion of importance of the four values on a per cluster (or site name) basis. Think of the proportions as percentages; the sum of the relative proportions must equal 100%. The default ratio is 50/50/0/0, which ignores the advisor and system information. In your environment, you may need to try different proportions to find the combination that gives the best performance.

Note: When adding an advisor (other than WLM), if the **port proportion** is zero, then the manager increases this value to 1. Because the sum of the relative proportions must total 100, the highest value is then decreased by 1.

When adding the WLM advisor, if the **system metric proportion** is zero, then the manager increases this value to 1. Because the sum of the relative proportions must total 100, the highest value is then decreased by 1.

The number of active connections is dependent upon the number of clients as well as the length of time necessary to use the services that are being provided by the load balanced server machines. If the client connections are quick (such as small Web pages served using HTTP GET), then the number of active connections are fairly low. If the client connections are slower (such as a database query), then the number of active connections are higher.

You should avoid setting active and new connections proportions values too low. You will disable load balancing and smoothing unless you have these first two values set to at least 20 each.

To set the proportion of importance values use the **dscontrol cluster set** *cluster* **proportions** command. See "dscontrol cluster — configure clusters" on page 295 for more information.

Weights

Weights are set by the manager function based upon internal counters in the executor, feedback from the advisors, and feedback from a system-monitoring program, such as Metric Server. If you want to set weights manually while running the manager, specify the fixedweight option on the dscontrol server command. For a description of the fixedweight option, see "Manager fixed weights" on page 144.

Weights are applied to all servers on a port. For any particular port, the requests are distributed between servers based on their weights relative to each other. For example, if one server is set to a weight of 10, and the other to 5, the server set to 10 should get twice as many requests as the server set to 5.

To specify the maximum weight boundary that any server can have, use the **dscontrol port set** *port* **weightbound** *weight* command. This command affects how much difference there can be between the number of requests each server will get. If you set the maximum weightbound to 1, then all the servers can have a weight of 1, 0 if quiesced, or -1 if marked down. As you increase this number, the difference in how servers can be weighted is increased. At a maximum weightbound of 2, one server could get twice as many requests as another. At a maximum weightbound of 10, one server could get 10 times as many requests as another. The default maximum weightbound is 20.

If an advisor finds that a server has gone down, it tells the manager, which sets the weight for the server to zero. As a result, the executor will not send any additional connections to that server as long as that weight remains zero. If there were any active connections to that server before the weight changed, they will be left to complete normally.

If all the servers are down, the manager sets the weights to half the weightbound.

Manager fixed weights

Without the manager, advisors cannot be run and cannot detect if a server is down. If you choose to run the advisors, but do *not* want the manager to update the weight you have set for a particular server, use the **fixedweight** option on the dscontrol server command. For example:

```
dscontrol server set cluster:port:server fixedweight yes
```

After fixedweight is set to yes, use the **dscontrol server set weight** command to set the weight to the value you desire. The server weight value remains fixed while the manager is running until you issue another dscontrol server command with fixedweight set to no. For more information, see "dscontrol server — configure servers" on page 330.

Sending TCP reset to a down server (Dispatcher component only)

If TCP **reset** is activated, Dispatcher will send a TCP reset to the client when the client has a connection to a server whose weight is 0. A server's weight may be 0 if it is configured 0 or if an advisor marks it down. A TCP reset will cause the connection to be immediately closed. This feature is useful for long-lived connections where it hastens the client's ability to renegotiate a failed connection. To activate TCP reset, use the **dscontrol port add|set** *port* **reset yes** command. The default value for reset is no.

Note: TCP reset applies to all of the Dispatcher's forwarding methods. However, to use the TCP reset feature, the **clientgateway** on the **dscontrol executor** command must be set to a router address.

A useful feature to configure, in conjunction with TCP reset, is **advisor retry**. With this feature, an advisor has the ability to retry a connection before marking a server down. This would help prevent the advisor from marking the server down prematurely which could lead to connection-reset problems. That is, just because the advisor failed on the first attempt does not necessarily mean that the existing connections are also failing. See "Advisor retry" on page 149 for more information.

Manager intervals

To optimize overall performance, the manager is restricted in how often it can interact with the executor. You can make changes to this interval by entering the **dscontrol manager interval** and **dscontrol manager refresh** commands.

The manager interval specifies how often the manager will update the server weights that the executor uses in routing connections. If the manager interval is too low, it can mean poor performance as a result of the manager constantly interrupting the executor. If the manager interval is too high, it can mean that the executor's request routing will not be based on accurate, up-to-date information.

For example, to set the manager interval to 1 second, enter the following command:

```
dscontrol manager interval 1
```

The manager refresh cycle specifies how often the manager will ask the executor for status information. The refresh cycle is based on the interval time.

For example, to set the manager refresh cycle to 3, enter the following command:

```
dscontrol manager refresh 3
```

This will cause the manager to wait for 3 intervals before asking the executor for status.

Sensitivity threshold

Other methods are available for you to optimize load balancing for your servers. To work at top speed, updates to the weights for the servers are only made if the weights have changed significantly. Constantly updating the weights when there is little or no change in the server status would create an unnecessary overhead. When the percentage weight change for the total weight for all servers on a port is greater than the sensitivity threshold, the manager updates the weights used by the executor to distribute connections. Consider, for example, that the total weight changes from 100 to 105. The change is 5%. With the default sensitivity threshold of 5, the manager will not update the weights used by the executor, because the percentage change is not **above** the threshold. If, however, the total weight changes from 100 to 106, the manager will update the weights. To set the manager's sensitivity threshold to a value other than the default (for example, 6), enter the following command:

```
dscontrol manager sensitivity 6
```

In most cases, you will not need to change this value.

Smoothing index

The manager calculates the server weights dynamically. As a result, an updated weight can be very different from the previous one. Under most circumstances, this will not be a problem. Occasionally, however, it may cause an oscillating effect in the way the requests are load balanced. For example, one server can end up receiving most of the requests due to a high weight. The manager will see that the server has a high number of active connections and that the server is responding slowly. It will then shift the weight over to the free servers and the same effect will occur there too, creating an inefficient use of resources.

To alleviate this problem, the manager uses a smoothing index. The smoothing index limits the amount that a server's weight can change, effectively smoothing the change in the distribution of requests. A higher smoothing index will cause the server weights to change less drastically. A lower index will cause the server weights to change more drastically. The default value for the smoothing index is 1.5. At 1.5, the server weights can be rather dynamic. An index of 4 or 5 will cause the weights to be more stable. For example, to set the smoothing index to 4, enter the following command:

```
dscontrol manager smoothing 4
```

In most cases, you will not need to change this value.

Using scripts to generate an alert or record server failure

Load Balancer provides user exits that trigger scripts that you can customize. You can create the scripts to perform automated actions, such as alerting an Administrator when servers are marked down by the manager or simply record the event of the failure. Sample scripts, which you can customize, are in the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/samples**
- Windows operating systems: **<***install_root***>ibm\edge\lb\samples**

In order to run the files, you must move them to the following directory and remove the "sample" file extension:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/bin**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\bin**

The following sample scripts are provided:

- **serverDown** — a server is marked down by the manager.
- **serverUp** — a server is marked back up by the manager.
- **managerAlert** — all servers are marked down for a particular port.
- **managerClear** — at least one server is now up, after all were marked down for a particular port.

If all servers on a cluster are marked down (either by the user or by the advisors), the managerAlert (if configured) starts, and Load Balancer attempts to route traffic to the servers using a round-robin technique. The serverDown script does not start when the last server in the cluster is detected as offline.

By design, Load Balancer attempts to continue to route the traffic in case a server comes back online and responds to the request. If Load Balancer instead dropped all traffic, the client would receive no response.

When Load Balancer detects that the first server of a cluster is back online, the managerClear script (if configured) starts, but the serverUp script (if configured) is not run until an additional server is brought back online.

Considerations when using **serverUp** and **serverDown** scripts:

- If you define the manager cycle to be less than 25% of the advisor time, false reports of servers up or down can result. By default, the manager runs every 2 seconds, but the advisor runs every 7 seconds. Therefore, the manager expects new advisor information within 4 cycles. However, removing this restriction (that is, defining the manager cycle to be greater than 25% of the advisor time) significantly decreases performance because multiple advisors can advise on a single server.
- When a server goes down, the serverDown script starts. However, if you issue a serverUp command, it is assumed that the server is up until the manager obtains new information from the advisor cycle. If the server is still down, the serverDown script runs again.

Advisors

Advisors are agents within Load Balancer. Their purpose is to assess the health and loading of server machines. They do this with a proactive client-like exchange with the servers. Advisors can be considered as lightweight clients of the application servers.

The product provides several protocol-specific advisors for the most popular protocols. However, it does not make sense to use all of the provided advisors with every component of Load Balancer. (For instance, you do not use the Telnet advisor with the CBR component.) Load Balancer also supports the concept of a “custom advisor” that allows users to write their own advisors.

Limitation on using bind-specific server applications: In order to use advisors on bind specific servers, start two instances of the server: One instance to bind on the cluster:port and the other instance to bind on the server:port. To determine if the

server is bind specific, issue the `netstat -an` command and look for the server:port. If the server is not bind specific, the result from this command will be 0.0.0.0:80. If the server is bind specific, you will see an address such as 192.168.15.103:80.

For HP-UX and Solaris systems, limitation on using bind-specific server applications: If using arp publish instead of ifconfig alias command, Load Balancer *will* support the use of advisors when load-balancing servers with bind-specific server applications (including other Load Balancer components such as CBR or Site Selector) when they are binding to the cluster IP address. However, when using advisors against bind-specific server application, do not collocate Load Balancer on the same machine with the server application.

Note: When Load Balancer is running on a computer with multiple network adapter cards, and if you want the advisor traffic to flow over a particular adapter, you can force the source IP address of the packets to be a particular address. To force the advisor packet source address to a particular address, add the following to the `java...SRV_XXXConfigServer...` line of the appropriate Load Balancer start script file (dsserver, cbrserver, or ssserver):
`-DLB_ADV_SRC_ADDR=`*`IP_address`*

How advisors work

Advisors periodically open a TCP connection with each server and send a request message to the server. The content of the message is specific to the protocol running on the server. For example, the HTTP advisor sends an HTTP "HEAD" request to the server.

Advisors then listen for a response from the server. After getting the response, the advisor makes an assessment of the server. To calculate this "load" value, most advisors measure the time for the server to respond, and then use this value (in milliseconds) as the load.

Advisors then report the load value to the manager function, where it appears in the manager report in the "Port" column. The manager then calculates aggregate weight values from all its sources, per its proportions, and sets these weight values into the executor function. The Executor will then use these weights for load balancing new incoming client connections.

If the advisor determines that a server is alive and well, it will report a positive, non-zero load number to the Manager. If the advisor determines that a server is not active, it will return a special load value of negative one (-1). The Manager and the Executor will not forward any further connections to that server until that server has come back up.

Note: Before sending the initial request message, the advisor will ping the server. This is intended to provide quick status to determine if the machine is online. After the server responds to the ping, no more pings are sent. To disable the pings, add -DLB_ADV_NO_PING to the Load Balancer start script file.

Starting and stopping an advisor

You can start an advisor for a particular port across all clusters (group advisor). Or, you can choose to run different advisors on the same port, but on different clusters

(cluster/site specific advisor). For example, if you have Load Balancer defined with three clusters (*clusterA, clusterB, clusterC*), each having port 80 you can do the following:

- Cluster/site specific advisor: To start an advisor on port 80 for *clusterA*, specify both the cluster and port:

  ```
  dscontrol advisor start http clusterA:80
  ```

  This command will start the HTTP advisor on port 80 for *clusterA*. The HTTP advisor will advise on all servers attached to port 80 for clusterA.
- Group advisor: To start a custom advisor on port 80 for all other clusters, simply specify the port:

  ```
  dscontrol advisor start ADV_custom 80
  ```

  This command will start the *ADV_custom* advisor on port 80 for *clusterB* and *clusterC*. Your custom advisor will advise on all servers attached to port 80 for *clusterB* and *clusterC*. (For more information on custom advisors, see "Create custom (customizable) advisors" on page 154.)

  Note: The group advisor will advise on all clusters/sites that do not currently have a cluster/site specific advisor.

Using the previous configuration example for the group advisor, you can choose to stop the custom advisor *ADV_custom* for port 80 on just one of the clusters or for both clusters (*clusterB* and *clusterC*).

- To stop the custom advisor for port 80 on just *clusterB*, specify cluster and port:

  ```
  dscontrol advisor stop ADV_custom clusterB:80
  ```
- To stop the custom advisor for port 80 on *clusterB* and *clusterC*, specify just the port:

  ```
  dscontrol advisor stop ADV_custom 80
  ```

Advisor intervals

Note: The advisor defaults should work efficiently for the great majority of possible scenarios. Be careful when entering values other than the defaults.

The advisor interval sets how often an advisor asks for status from the servers on the port it is monitoring and then reports the results to the manager. If the advisor interval is too low, it can mean poor performance as a result of the advisor constantly interrupting the servers. If the advisor interval is too high, it can mean that the manager's decisions about weighting will not be based on accurate, up-to-date information.

For example, to set the interval to 3 seconds for the HTTP advisor for port 80, enter the following command:

```
dscontrol advisor interval http 80 3
```

It does not make sense to specify an advisor interval that is smaller than the manager interval. The default advisor interval is seven seconds.

Advisor report timeout

To make sure that out-of-date information is not used by the manager in its load-balancing decisions, the manager will not use information from the advisor whose time stamp is older than the time set in the advisor report timeout. The

advisor report timeout should be larger than the advisor polling interval. If the timeout is smaller, the manager will ignore reports that logically should be used. By default, advisor reports do not timeout — the default value is unlimited.

For example, to set the advisor report timeout to 30 seconds for the HTTP advisor for port 80, enter the following command:

```
dscontrol advisor timeout http 80 30
```

For more information on setting the advisor report timeout, see “dscontrol advisor — control the advisor” on page 289.

Advisor connect timeout and receive timeout for servers

For Load Balancer, you can set the advisor's timeout values at which it detects a particular port on the server (a service) is failed. The failed-server timeout values (connecttimeout and receivetimeout) determine how long an advisor waits before reporting that either a connect or receive has failed.

To obtain the fastest failed-server detection, set the advisor connect and receive timeouts to the smallest value (one second), and set the advisor and manager interval time to the smallest value (one second).

Note: If your environment experiences a moderate to high volume of traffic such that server response time increases, be careful not to set the connecttimeout and receivetimeout values too small, or the advisor may prematurely mark a busy server as failed.

For example, to set the connecttimeout and receivetimeout to 9 seconds for the HTTP advisor on port 80, type the following command:

```
dscontrol advisor connecttimeout http 80 9
dscontrol advisor receivetimeout http 80 9
```

The default for connect and receive timeout is 3 times the value specified for the advisor interval time.

Advisor retry

Advisors have the ability to retry a connection before marking a server down. The advisor will not mark a server down until the server query has failed the number of retries plus 1. The **retry** value should be no larger than 3. The following command sets a retry value of 2 for the LDAP advisor on port 389:

```
dscontrol advisor retry ldap 389 2
```

List of advisors

- The **HTTP** advisor opens a connection, sends a HEAD request by default, waits for a response connection, and returns the elapsed time as a load. See “Configuring the HTTP or HTTPS advisor using the request and response (URL) option” on page 152 for more information on how to change the type of request sent by the HTTP advisor.
- The **HTTPS** advisor is a "heavyweight" advisor for SSL connections. It performs a full SSL socket connection with the server. The HTTPS advisor opens an SSL connection, sends an HTTPS request, waits for a response, closes the connnection, and returns the elapsed time as a load. (See also the SSL advisor, which is a "lightweight" advisor for SSL connections.)

Note: The HTTPS advisor has no dependency upon server key or certificate content, but they must not be expired.

- The **SIP** advisor opens a connection, sends an OPTIONS request, waits for a response, closes the connection, and returns the elapsed time as a load. The SIP advisor that is supported runs on TCP only and requires an application to be installed on a server that responds to an OPTIONS request.
- The **FTP** advisor opens a connection, sends a SYST request, waits for a response, closes the connection, and returns the elapsed time as a load.
- The **LDAP** advisor opens a connection, sends an anonymous BIND request, waits for a response, closes the connection, and returns the elapsed time as a load.

  Note: Use the LDAPS advisor instead of this advisor if it is possible for your environment.
- The **LDAPS** is a secure version of the LDAP advisor. It performs a full SSL socket connection with the server. The LDAPS advisor opens an SSL connection, sends an LDAPS request, waits for a response, closes the connnection, and returns the elapsed time as a load.
- The **Telnet** advisor opens a connection, waits for an initial message from the server, closes the connection, and returns the elapsed time as a load.
- The **NNTP** advisor opens a connection, waits for an initial message from the server, sends a quit command, closes the connection, and returns the elapsed time as a load.
- The **IMAP** advisor opens a connection, waits for an initial message from the server, sends a quit command, closes the connection, and returns the elapsed time as a load.
- The **POP3** advisor opens a connection, waits for an initial message from the server, sends a quit command, closes the connection, and returns the elapsed time as a load.
- The **SMTP** advisor opens a connection, waits for an initial message from the server, sends a quit, closes the connection, and returns the elapsed time as a load.
- The **SSL** advisor is a "lightweight" advisor for SSL connections. It does not establish a full SSL socket connection with the server. The SSL advisor opens a connection, sends an SSL CLIENT_HELLO request, waits for a response, closes the connection, and returns the elapsed time as a load. (See also the HTTPS advisor, which is a "heavyweight" advisor for SSL connections.)

  Note: Be aware of the following:
  - The SSL advisor has no dependency on key management or certificates. Server key and certificate content must not be expired.
  - Use the LDAPS or TLS advisor if it is possible for your environment; these advisors are preferred over the SSL advisor.
  - You might want to use the TLS advisor if you find that the SSL advisor is marking servers down, you know that the server is still up, and you see messages stating "No ciphers found."
  - The SSL advisor has no dependency upon key management or certificates.
- The **ssl2http** advisor starts and advises on the servers listed under port 443, but the advisor will open a socket to the “mapport" for HTTP requests. Only use the ssl2http advisor for CBR if the client-to-proxy protocol is SSL and the

proxy-to-server protocol is HTTP. See “Load balancing client-to-proxy in SSL and proxy-to-server in HTTP” on page 73 for more information

- The **TLS** advisor is an alternative to the SSL advisor, and it is preferred over the SSL advisor. Transport Layer Security (TLS) is the successor to the SSL protocol. TLS 1.0 was the follow up to SSLv3.

  Use the TLS advisor if you find that the SSL advisor is marking servers down, you know that the server is still up, and you see messages stating "No ciphers found."
- The Caching Proxy (cachingproxy) advisor opens a connection, sends a Caching Proxy specific HTTP GET request, and interprets the response as a Caching Proxy load.

  Note: When using the Caching Proxy advisor, Caching Proxy needs to be running on all servers being load balanced. The machine on which the Load Balancer resides does not need to have Caching Proxy installed unless it is collocated on the same machine it is load balancing.
- The **connect** advisor does not exchange any protocol-specific data with the server. It simply measures the time it takes to open and close a TCP connection with the server. This advisor is useful for server applications which use TCP, but with a higher-level protocol for which an IBM-supplied or custom advisor is not available.
- The **DB2** advisor works in conjunction with the DB2 servers. Dispatcher has the built in capability of checking the health of DB2 servers without the need for customers to write their own custom advisors. The DB2 advisor communicates with the DB2 connection port only, not the Java connection port.
- The **DNS** advisor opens a connection, sends a pointer query for DNS, waits for a response, closes the connection and returns the elapsed time as a load.
- The **ping** advisor does not open a TCP connection with the servers, but instead reports whether the server responds to a ping. While the ping advisor may be used on any port, it is also designed for configurations using the wildcard port, over which multiple protocol traffic may be flowing. It is also useful for configurations using non-TCP protocols with their servers, such as UDP.

  Note: If you disable pings in your Load Balancer start script, the ping advisor will not work.
- The **reach** advisor pings its target machines. This advisor is also designed for the Dispatcher's high availability components to determine reachability of its reach targets. Its results flow to high availability component and do *not* appear in the manager report. Unlike the other advisors, the reach advisor starts automatically by the manager function of the Dispatcher component.
- The **self** advisor collects load status information on backend servers. You can use the self advisor when using Dispatcher in a two–tiered configuration, where the Dispatcher furnishes information from the self advisor to the top-tiered Load Balancer. The self advisor specifically measures the connections per second rate on backend servers of the Dispatcher at the executor level. See “Using Self Advisor in a two-tiered WAN configuration” on page 153 for more information.
- The **WLM** (Workload Manager) advisor is designed to work in conjunction with servers on OS/390 mainframes running the MVS™ Workload Manager (WLM) component. For more information, see “Workload Manager advisor” on page 159.
- Dispatcher provides the ability for a customer to write a *custom* (customizable) advisor. This enables support for proprietary protocols (on top of TCP) for

which IBM has not developed a specific advisor. For more information, see "Create custom (customizable) advisors" on page 154.

- The **WAS** (WebSphere Application Server) advisor works in conjunction with the WebSphere Application servers. Customizable sample files for this advisor are provided in the installation directory. For more information, see "WAS advisor" on page 155.

Configuring the HTTP or HTTPS advisor using the request and response (URL) option

The URL option for the HTTP or HTTPS advisor is available for the Dispatcher and CBR components.

After you have started an HTTP or HTTPS advisor, you can define a unique client HTTP URL string, specific for the service that you want to query on the server. This allows the advisor to assess the health of the individual services within a server. You can do this by defining logical servers with unique server names that have the same physical IP address. See "Server Partitioning: logical servers configured to one physical server (IP address)" on page 43 for more information.

For each defined logical server under the HTTP port you can specify a unique client HTTP URL string, specific for the service that you want to query on the server. The HTTP or HTTPS advisor uses the **advisorrequest** string to query the health of the servers. The default value is `HEAD / HTTP/1.0`. The **advisorresponse** string is the response that the advisor scans for in the HTTP response. The advisor uses the **advisorresponse** string to compare to the real response that is received from the server. The default value is null.

Important: If a blank is contained within the HTTP URL string:

- When issuing the command from the **dscontrol>>** shell prompt, you must place quotes around the string if a blank is contained within the string. For example:

```
server set cluster:port:server advisorrequest "head / http/1.0"
server set cluster:port:server advisorresponse "HTTP 200 OK"
```

- When issuing the **dscontrol** command from the operating system prompt, you must precede the text with **"\"** and follow the text with **\""**. For example:

```
dscontrol server set cluster:port:server
advisorrequest "\"head / http/1.0\""

dscontrol server set cluster:port:server advisorresponse "\"HTTP 200 OK\""
```

When you create the request that the HTTP or HTTPS advisor sends to backend servers to see if they are functioning, you type the start of the HTTP request and Load Balancer completes the end of the request with the following:

```
\r\nAccept:
*/*\r\nUser-Agent:IBM_Load_Balanacer_HTTP_Advisor\r\n\r\n
```

If you want to add other HTTP header fields before Load Balancer appends this string to the end of the request, you can do so by including your own `\r\n` string in the request. The following is an example of what you might type to add the HTTP host header field to your request:

```
GET /pub/WWW/TheProject.html HTTP/1.0 \r\nHost: www.w3.org
```

Note: After starting an HTTP or HTTPS advisor for a specified HTTP port number, the advisor request and response value is enabled for servers under that HTTP port.

See “dscontrol server — configure servers” on page 330 for more information.

Using Self Advisor in a two-tiered WAN configuration

The self advisor is available on the Dispatcher component.

For Load Balancer in a two-tiered WAN (wide area network) configuration, Dispatcher provides a *self* advisor that collects load status information on backend servers.

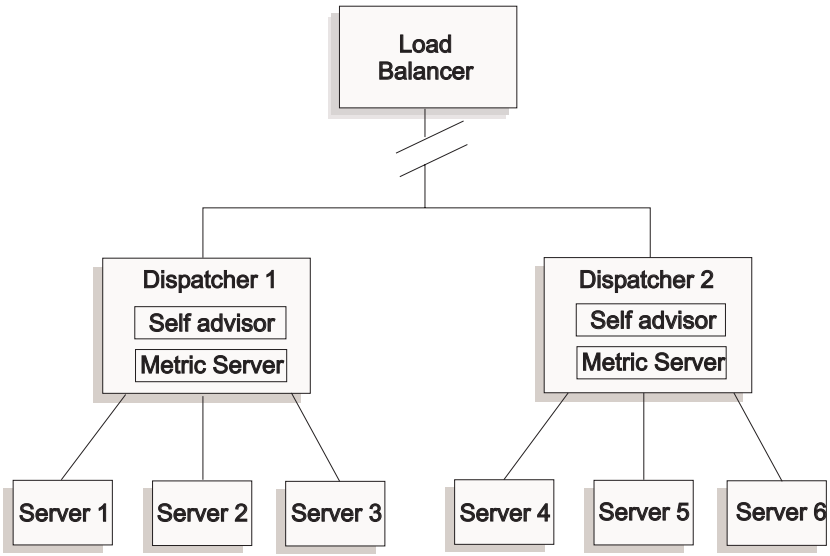


Figure 31. Example of a two-tiered WAN configuration using the self advisor

In this example, the self advisor along with Metric Server reside on the two Dispatcher machines that are being load balanced by the top tier Load Balancer. The self advisor specifically measures the connections per second rate on backend servers of the Dispatcher at the executor level.

The self advisor writes the results to the dsloadstat file. Load Balancer also provides an external metric called dsload. The Metric Server agent on each Dispatcher machine runs its configuration that calls the external metric dsload. The dsload script extracts a string from the dsloadstat file and returns it to the Metric Server agent. Subsequently, each of the Metric Server agents (from each of the Dispatchers) returns the load status value to the top-tiered Load Balancer for use in determining which Dispatcher to forward client requests.

The dsload executable resides in the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/ms/script**
- Windows operating systems: **<***install_root***>ibm\edge\lb\ms\script**

See “Configure wide area Dispatcher support” on page 184 for more information on using Dispatcher in WAN configurations. See “Metric Server” on page 157 for more information on Metric Server.

Create custom (customizable) advisors

The custom (customizable) advisor is a small piece of Java code, which you provide as a class file, that gets called by the base code. The base code provides all administrative services, such as starting and stopping an instance of the custom advisor, providing status and reports, and recording history information in a log file. It also reports results to the manager component. Periodically the base code will perform an advisor cycle, where it individually evaluates all servers in its configuration. It starts by opening a connection with a server machine. If the socket opens, the base code will call the "getLoad" method (function) in the custom advisor. The custom advisor then performs whatever steps are necessary to evaluate the health of the server. Typically, it will send a user-defined message to the server and then wait for a response. (Access to the open socket is provided to the custom advisor.) The base code then closes the socket with the server and reports the load information to the Manager.

The base code and custom advisor can operate in either normal or replace mode. Choice of the mode of operation is specified in the custom advisor file as a parameter in the constructor method.

In normal mode, the custom advisor exchanges data with the server, and the base advisor code times the exchange and calculates the load value. The base code then reports this load value to the manager. The custom advisor needs only return a zero (on success) or negative one (on error). To specify normal mode, the replace flag in the constructor is set to false.

In replace mode, the base code does not perform any timing measurements. The custom advisor code performs whatever operations are desired for its unique requirements, and then returns an actual load number. The base code will accept the number and report it to the manager. For best results, normalize your load number between 10 and 1000, with 10 representing a fast server, and 1000 representing a slow server. To specify replace mode, the replace flag in the constructor is set to true.

With this feature, you can write your own advisors that will provide the precise information about servers that you need. A sample custom advisor, **ADV_sample.java**, is provided with the Load Balancer product.

After installing Load Balancer, you may find the sample code in:

- For AIX, HP-UX, Linux, Solaris systems: /opt/ibm/edge/lb/servers/samples/CustomAdvisors
- For Windows systems: <*install_root*>ibm\edge\lb\servers\samples\CustomAdvisors

Note: If you add a custom advisor to Dispatcher, or any other applicable Load Balancer component, you must stop and then restart **dsserver** (or the service for Windows systems) to enable the Java process to read the new custom advisor class files. The custom advisor class files are loaded only at startup. It is not necessary to stop the executor. The executor continues to run even when dsserver, or the service, has been stopped.

If the custom advisor references additional Java classes, the classpath in the Load Balancer start script file (dsserver, cbrserver, ssserver) should be updated to include the location.

WAS advisor

Sample custom advisor files specifically for the WebSphere Application Server (WAS) advisor are provided in the Load Balancer installation directory.

- ADV_was.java is the file to be compiled and run on the Load Balancer machine
- LBAdvisor.java.servlet (to be renamed LBAdvisor.java) is the file to be compiled and run on the WebSphere Application Server machine.

The WebSphere Application Server advisor sample files reside in the same samples directory as the ADV_sample.java file.

Naming Convention

Your custom advisor file name must be in the form "ADV_*myadvisor*.java." It must start with the prefix " ADV_" in uppercase. All subsequent characters must be in lowercase letters.

As per Java conventions, the name of the class defined within the file must match the name of the file. If you copy the sample code, be sure to change all instances of "ADV_sample" inside the file to your new class name.

Compilation

Custom advisors are written in Java language. Use the Java compiler that is installed with Load Balancer. These files are referenced during compilation:

- the custom advisor file
- the base classes file, ibmlb.jar, found in the following directory:
  - AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/lib**
  - Windows operating systems: **<***install_root***>ibm\edge\lb\servers\lib**

Your class path must point to both the custom advisor file and the base classes file during the compile.

For Windows systems, a sample compile command is:

```
install_dir/java/bin/javac -classpath
    install_dir\lb\servers\lib\ibmlb.jar ADV_fred.java
```

where:

- Your advisor file is named ADV_fred.java
- Your advisor file is stored in the current directory

The output for the compilation is a class file, for example

```
ADV_fred.class
```

Before starting the advisor, copy the class file to the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/lib/CustomAdvisors**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\lib\CustomAdvisors**

Note: If you wish, custom advisors may be compiled on one operating system and run on another. For example, you may compile your advisor on Windows systems, copy the class file (in binary) to an AIX machine, and run the custom advisor there.

For AIX, HP-UX, Linux, and Solaris systems, the syntax is similar.

Run

To run the custom advisor, you must first copy the class file to the proper installation directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/lib/CustomAdvisors/ADV_fred.class**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\lib\CustomAdvisors\ADV_fred.class**

Configure the component, start its manager function, and issue the command to start your custom advisor:

```
dscontrol advisor start fred 123
```

where:

- fred is the name of your advisor, as in ADV_fred.java
- 123 is the port on which your advisor will operate

If the custom advisor references additional Java classes, the classpath in the Load Balancer start script file (dsserver, cbrserver, ssserver) should be updated to include the location.

Required routines

Like all advisors, a custom advisor extends the function of the advisor base, called ADV_Base. It is the advisor base that actually performs most of the advisor's functions, such as reporting loads back to the manager for use in the manager's weight algorithm. The advisor base also performs socket connect and close operations and provides send and receive methods for use by the advisor. The advisor itself is used only for sending and receiving data to and from the port on the server being advised. The TCP methods within the advisor base are timed to calculate the load. A flag within the constructor in the ADV_base overwrites the existing load with the new load returned from the advisor if desired.

Note: Based on a value set in the constructor, the advisor base supplies the load to the weight algorithm at specified intervals. If the actual advisor has not completed so that it can return a valid load, the advisor base uses the previous load.

These are base class methods:

- A **constructor** routine. The constructor calls the base class constructor (see the sample advisor file)
- An **ADV_AdvisorInitialize** method. This method provides a hook in case additional steps need to be taken after the base class completes its initialization.
- A **getload** routine. The base advisor class performs the open socket; therefore getload needs only to issue the appropriate send and receive requests to complete the advise cycle.

Search order

Load Balancer first looks at the list of native advisors that it provides. If it does not find a given advisor there, Load Balancer then looks at the customer's list of customized advisors.

Naming and path

- The custom advisor class must be located within the subdirectory of the CustomAdvisors directory. The defaults for this directory vary by operating system:
  - AIX, HP-UX, Linux, and Solaris systems

    `/opt/ibm/edge/lb/servers/lib/CustomAdvisors/`
  - Windows systems

    `<install_root>ibm\edge\lb\servers\lib\CustomAdvisors`
- Only lowercase, alphabetic characters are permitted. This eliminates case sensitivity when an operator types in commands on the command line. The advisor file name must be prefixed with **ADV_**.

Sample advisor

The program listing for a sample advisor is included in "Sample advisor" on page 419. After installation, this sample advisor can be found in the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/lib/CustomAdvisors**
- Windows operating systems: **<*install_root*>ibm\edge\lb\servers\lib\CustomAdvisors**

Metric Server

This feature is available for all the Load Balancer components.

Metric Server provides server load information to the Load Balancer in the form of system-specific metrics, reporting on the health of the servers. The Load Balancer manager queries the Metric Server agent residing on each of the servers, assigning weights to the load balancing process using the metrics gathered from the agents. The results are also placed into the manager report.

Note: When two or more metrics are gathered and normalized for each server into a single system load value, rounding errors may occur.

For information on operating Metric Server (starting and stopping) and using Metric Server logs see "Using the Metric Server component" on page 230.

For a configuration example see Figure 5 on page 12.

WLM Restriction

Like the WLM advisor, the Metric Server reports on server systems as a whole, rather than on individual protocol-specific server daemons. Both WLM and Metric Server place their results into the system column of the manager report. As a consequence, running both the WLM advisor and Metric Server at the same time is not supported.

Prerequisites

The Metric Server agent must be installed and running on all servers that are being load balanced.

How to Use Metric Server

Below are the steps to configure Metric Server for Dispatcher. Similar steps can be used for configuring Metric Server for the other components of Load Balancer.

- Load Balancer manager (Load Balancer side)
  1. Start **dsserver**.
  2. Issue command: **dscontrol manager start** *manager.log port*

     port is the RMI port chosen for all the Metric Server agents to run on. The default RMI port that is set in the metricserver.cmd file is 10004.
  3. Issue command: **dscontrol metric add** *cluster:systemMetric*

     systemMetric is the name of the script (residing on the backend server) which should run on each of the servers in the configuration under the specified cluster (or site name). Two scripts are provided for the customer - **cpuload** and **memload**. Or, you can create custom system metric scripts. The script contains a command which should return a numeric value in the range of 0-100 or a value of -1 if the server is down. This numeric value should represent a load measurement, not an availability value.

     Note: For Site Selector, cpuload and memload run automatically.

     Limitation: For Windows platform, if the name of your System Metric script has an extension other than ".exe", you must specify the full name of the file (for example, "mysystemscript.bat"). This is due to a Java limitation.
  4. Add to the configuration only servers that contain a Metric Server agent running on the port specified in the metricserver.cmd file. The port should match the port value specified in the **manager start** command.

  Note: Ensure Security —
     - On the Load Balancer machine, create a key file (using **lbkeys create** command). See “Remote Method Invocation (RMI)” on page 213 for more information on lbkeys.
     - On the backend server machine, copy the resulting key file, for the component you are using, to the following directory:
       - AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/admin/keys**
       - Windows operating systems: **<***install_root***>ibm\edge\lb\admin\keys**

       Verify that the key file's permissions enable the file to be readable by the root.
- Metric Server agent (Server machine side)
  1. Install the Metric Server package from the Load Balancer install.
  2. Check the **metricserver** script in the **/usr/bin** directory to verify that the desired RMI port is being used. (For Windows 2003, the directory is C:\WINDOWS\system32.) The default RMI port is 10004.

     Note: The RMI port value specified must be the same value as the RMI port value for the Metric Server on the Load Balancer machine.
  3. The following two scripts are already provided for the customer: **cpuload** (returns the percentage of cpu in use ranging from 0-100) and **memload** (returns the percentage of memory in use ranging from 0-100). These scripts reside in the following directory:
     - AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/ms/script**
     - Windows operating systems: **<***install_root***>ibm\edge\lb\ms\script**

Optionally, customers can write their own customized metric script files which define the command that the Metric Server will issue on the server machines. Ensure that any custom scripts are executable and located in the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/ms/script**
- Windows operating systems: **<***install_root***>ibm\edge\lb\ms\script**

Custom scripts **must** return a numeric load value in the range of 0-100.

Note: A custom metric script must be a valid program or script with a ".bat" or ".cmd" extension. Specifically, for AIX, HP-UX, Linux, and Solaris operating systems, scripts must begin with the shell declaration, otherwise they may not properly run.

4. Start the agent by issuing the **metricserver** command.
5. To stop the Metric Server agent, issue the **metricserver stop** command.

To have Metric Server run on an address other than the local host, you need to edit the metricserver file on the load balanced server machine. After the occurence of "java" in the metricserver file, insert the following:

```
-Djava.rmi.server.hostname=OTHER_ADDRESS
```

In addition, before the "if" statements in the metricserver file, add the following line: `hostname OTHER_ADDRESS`.

For Windows platform: You will also need to alias the *OTHER_ADDRESS* on the Microsoft stack of the Metric Server machine. For example:

```
call netsh interface ip add address "Local Area Connection"
   addr=9.37.51.28 mask=255.255.240.0
```

When gathering metrics across different domains, you must explicitly set the java.rmi.server.hostname in the server script (dsserver, cbrserver, etc) to the fully qualified domain name (FQDN) of the machine that is requesting the metrics. This is necessary because, depending on your setup and operating system, InetAddress.getLocalHost.getHostName() might not return the FQDN.

Workload Manager advisor

WLM is code that runs on MVS mainframes. It can be queried to ask about the load on the MVS machine.

When MVS Workload Management has been configured on your OS/390 system, Dispatcher can accept capacity information from WLM and use it in the load balancing process. Using the WLM advisor, Dispatcher will periodically open connections through the WLM port on each server in the Dispatcher host table and accept the capacity integers returned. Because these integers represent the amount of capacity that is still available and Dispatcher expects values representing the loads on each machine, the capacity integers are inverted by the advisor and normalized into load values (that is, a large capacity integer but a small load value both represent a healthier server). The resulting loads are placed into the System column of the manager report.

There are several important differences between the WLM advisor and other Dispatcher advisors:

1. Other advisors open connections to the servers using the same port on which flows normal client traffic. The WLM advisor opens connections to the servers using a port different from normal traffic. The WLM agent on each server machine must be configured to listen on the same port on which the Dispatcher WLM Advisor is started. The default WLM port is 10007.
2. Other advisors only assess those servers defined in the Dispatcher cluster:port:server configuration for which the server's port matches the advisor's port. The WLM advisor advises upon *every* server in the Dispatcher configuration (regardless of the cluster:port). Therefore you must not define any non-WLM servers when using the WLM advisor.
3. Other advisors place their load information into the manager report under its "Port" column. The WLM advisor places its load information into the manager report under its system column.
4. It is possible to use both protocol-specific advisors along with the WLM advisor. The protocol-specific advisors will poll the servers on their normal traffic ports, and the WLM advisor will poll the system load using the WLM port.

Metric Server Restriction

Like the Metric Server agent, the WLM agent reports on server systems as a whole, rather than on individual protocol-specific server daemons. Metric Server and WLM place their results into the system column of the manager report. As a consequence, running both the WLM advisor and Metric Server at the same time is not supported.

Chapter 21. Advanced features for Dispatcher, CBR, and Site Selector

This chapter explains how to configure the load balancing parameters and how to set up Load Balancer for advanced functions.

Note: When reading this chapter, if you are *not* using the Dispatcher component, then substitute "dscontrol" with the following:

- For CBR, use **cbrcontrol**
- For Site Selector, use **sscontrol** (see Chapter 27, "Command reference for Site Selector," on page 341)

Table 11. Advanced configuration tasks for the Load Balancer

Task	Description	Related information
Collocate Load Balancer on a machine that it is load balancing	Set up a collocated Load Balancer machine.	"Using collocated servers" on page 162
Configure high availability or mutual high availability	Set up a second Dispatcher machine to provide a backup.	"High availability" on page 164
Configure rules-based load balancing	Define conditions under which a subset of your servers are used.	"Configure rules-based load balancing" on page 170
Use port affinity override to provide a mechanism for a server to override the port sticky feature	Allows a server to override the stickytime setting on its port.	"port affinity override" on page 176
Use sticky (affinity) feature to configure a cluster's port to be sticky	Allows client requests to be directed to the same server.	"How affinity feature for Load Balancer works" on page 178
Use cross port affinity to expand the sticky (affinity) feature across ports	Allows client requests received from different ports to be directed to the same server.	"Cross port affinity" on page 179
Use affinity address mask to designate a common IP subnet address	Allows clients requests received from the same subnet to be directed to the same server.	"Affinity address mask (stickymask)" on page 179
Use active cookie affinity to load balance servers for CBR	A rule option that allows a session to maintain affinity for a particular server.	"Active cookie affinity" on page 181
Use passive cookie affinity to load balance servers for Dispatcher's content-based routing and the CBR component	A rule option that allows a session to maintain affinity for a particular server based on the cookie name/cookie value.	"Passive cookie affinity" on page 183
Use URI affinity to load-balance across Caching Proxy servers with unique content to be cached on each individual server	A rule option that allows a session to maintain affinity for a particular server based on the URI.	"URI affinity" on page 184
Configure wide area Dispatcher support	Set up a remote Dispatcher to load balance across a wide area network. Or, load balance across a wide area network (without a remote Dispatcher) using a server platform that supports GRE.	"Configure wide area Dispatcher support" on page 184
Use explicit linking	Avoid bypassing the Dispatcher in your links.	"Using explicit linking" on page 190

Table 11. Advanced configuration tasks for the Load Balancer (continued)

Task	Description	Related information
Use a private network	Configure the Dispatcher to load balance servers on a private network.	"Using a private network configuration" on page 190
Use wildcard cluster to combine common server configurations	Addresses that are not explicitly configured will use the wildcard cluster as a way to load balance traffic.	"Use wildcard cluster to combine server configurations" on page 191
Use wildcard cluster to load balance firewalls	All traffic will be load balanced to firewalls.	"Use wildcard cluster to load balance firewalls" on page 191
Use wildcard cluster with Caching Proxy for transparent proxy	Allows Dispatcher to be used to enable a transparent proxy.	"Use wildcard cluster with Caching Proxy for transparent proxy" on page 192
Use wildcard port to direct unconfigured port traffic	Handles traffic that is not configured for any specific port.	"Use wildcard port to direct unconfigured port traffic" on page 192
Use "Denial of Service Attack" detection to notify administrators (via an alert) of potential attacks	Dispatcher analyzes incoming requests for a conspicuous amount of half-open TCP connections on servers.	"Denial of service attack detection" on page 193
Use binary logging to analyze server statistics	Allows server information to be stored in and retrieved from binary files.	"Using binary logging to analyze server statistics" on page 194
Use a collocated client configuration	Allow Load Balancer to reside on the same machine as a client	"Using a collocated client" on page 196

Using collocated servers

Load Balancer can reside on the same machine as a server for which it is load balancing requests. This is commonly referred to as *collocating* a server. Collocation applies to the Dispatcher and Site Selector components. Collocation is also supported for CBR, but only when using bind-specific Web servers and bind-specific Caching Proxy.

Note: A collocated server competes for resources with Load Balancer during times of high traffic. However, in the absence of overloaded machines, using a collocated server offers a reduction in the total number of machines necessary to set up a load-balanced site.

For the Dispatcher component

Linux: In order to configure both collocation and high availability at the same time, when running the Dispatcher component using the mac forwarding method, see "Linux loopback aliasing alternatives when using Load Balancer's mac forwarding" on page 60.

Solaris: There is a limitation that you cannot configure WAN advisors when the entry-point Dispatcher is collocated. See "Using remote advisors with Dispatcher's wide area support" on page 186.

Windows: Collocation is no longer available.

In earlier releases, it was necessary to specify the collocated server address to be the same as the nonforwarding address (NFA) in the configuration. That restriction has been lifted.

To configure a server to be collocated, the **dscontrol server** command provides an option called **collocated** which can be set to *yes* or *no*. The default is no. The address of the server must be a valid IP address of a network interface card on the machine. The collocated parameter should not be set for servers which are collocated using Dispatcher's nat or cbr forwarding method.

You can configure a collocated server in one of the following ways:

- If you are using the NFA as the collocated server address: Set the NFA using the **dscontrol executor set nfa** *IP_address* command. And, add the server using the NFA address with the **dscontrol server add** *cluster:port:server* command.
- If you are using an address other than the NFA: Add the server with the desired IP address with the collocated parameter set to yes as follows: **dscontrol server add** *cluster:port:server* **collocated** *yes*.

For Dispatcher's nat or cbr forwarding, you must configure (alias) an unused adapter address on the NFA. The server should be configured to listen on this address. Configure the server using the following command syntax:

```
dscontrol server add cluster:port:new_alias address new_alias router router_ip
returnaddress return_address
```

Failure to configure for this can lead to system errors, no response from the server, or both.

Configuring server collocation with Dispatcher's nat forwarding

When configuring a collocated server using Dispatcher's nat forwarding method, the router specified in the `dscontrol server add` command must be a real router address and not the server IP address.

Support for collocation when configuring Dispatcher's nat forwarding method can now be done on the following operating systems if the following steps are performed on the Dispatcher machine:

- **AIX, Linux, and Windows:** the collocated server is configured the same as any server. No changes are necessary to the configuration.
- **Solaris and HP-UX:** the cluster is aliased using ifconfig as normal; however, the return address must be arp published instead of aliased. To do this, run the following command:

  ```
  arp -s hostname ether_addr pub
  ```

  using the local MAC address for ether_addr. This enables the local application to send traffic to the return address in the kernel.

For the CBR component

CBR supports collocation on AIX, HP-UX, Linux, and Solaris platforms with no additional configurations required. However, the Web servers and Caching Proxy that you use must be bind-specific.

For the Site Selector component

Site Selector supports collocation on AIX, HP-UX, Linux, and Solaris platforms with no additional configurations required.

High availability

The high availability function (configurable using **dscontrol highavailability** command) is available for the Dispatcher component (but not for the CBR or Site Selector component).

To improve Dispatcher availability, the Dispatcher high availability function uses the following mechanisms:

- Two Dispatchers with connectivity to the same clients, and the same cluster of servers, as well as connectivity between the Dispatchers. Both Dispatchers must run on the same type of operating system and platform.
- A "heartbeat" mechanism between the two Dispatchers to detect Dispatcher failure. At least one heartbeat pair must have the NFAs of the pair as the source and destination address.

  If possible, at least one of the heartbeat pairs should be across a separate subnet than the regular cluster traffic. Keeping the heartbeat traffic distinct will help prevent false takeovers during very heavy network loads and also improve complete recovery times after a failover.
- A list of reach targets, addresses that both Dispatcher machines must be able to contact in order to load balance traffic normally. For more information, see "Failure detection capability using heartbeat and reach target" on page 166.
- Synchronization of the Dispatcher information (that is, the connection tables, reachability tables, and other information).
- Logic to elect the active Dispatcher which is in charge of a given cluster of servers, and the standby Dispatcher which continuously gets synchronized for that cluster of servers.
- A mechanism to perform IP takeover, when the logic or an operator decides to switch active and standby.

Note: For an illustration and description of a *mutual high availability* configuration, where two Dispatcher machines sharing two cluster sets provide backup for each other, see "Mutual high availability" on page 46. Mutual high availability is similar to high availability but is based specifically on cluster address rather than on a Dispatcher machine as a whole. Both machines must configure their shared cluster sets the same.

Configure high availability

Complete syntax for **dscontrol highavailability** is in "dscontrol highavailability — control high availability" on page 307.

For a more complete discussion of many of the tasks below, see "Setting up the Dispatcher machine" on page 50.

1. Create alias script files on each of the 2 Dispatcher machines. See "Using scripts" on page 167.
2. Start the server on both Dispatcher server machines.
3. Start the executor on both machines.
4. Ensure that the nonforwarding address (NFA) of each Dispatcher machine is configured, and is a valid IP address for the subnet of the Dispatcher machines.
5. Add the heartbeat information on both machines:

   ```
   dscontrol highavailability heartbeat add sourceaddress destinationaddress
   ```

Note: *Sourceaddress* and *destinationaddress* are the IP addresses (either DNSnames or IP addresses) of the Dispatcher machines. The values will be reversed on each machine. For example:

```
Primary - highavailability heartbeat add 9.67.111.3 9.67.186.8
Backup  - highavailability heartbeat add 9.67.186.8 9.67.111.3
```

At least one heartbeat pair must have the NFAs of the pair as the source and destination address.

If possible, at least one of the heartbeat pairs should be across a separate subnet than the regular cluster traffic. Keeping the heartbeat traffic distinct will help prevent false takeovers during very heavy network loads and also improve complete recovery times after a failover.

Set the number of seconds that the executor uses to timeout high availability heartbeats. For example:

```
dscontrol executor set hatimeout 3
```

The default is 2 seconds.

6. On both machines, configure the list of IP addresses that the Dispatcher must be able to reach in order to ensure full service, using the **reach add** command. For example:

   ```
   dscontrol highavailability reach add 9.67.125.18
   ```

   Reach targets are recommended but not required. See "Failure detection capability using heartbeat and reach target" on page 166 for more information.

7. Add the backup information to each machine:
   - For the **primary** machine:

     ```
     dscontrol highavailability backup add primary [auto | manual] port
     ```

   - For the **backup** machine:

     ```
     dscontrol highavailability backup add backup [auto | manual] port
     ```

   - For mutual high availability each Dispatcher machine has **both** primary and backup roles:

     ```
     dscontrol highavailability backup add both [auto | manual] port
     ```

   Note: Select an unused port on your machines as the *port*. The port number entered will be used as a key to ensure the correct host is receiving the packet.

8. Check the high availability status on each machine:

   ```
   dscontrol highavailability status
   ```

   The machines should each have the correct role (backup, primary, or both), states, and substates. The primary should be active and synchronized; the backup should be in standby mode and should be synchronized within a short time. The strategies must be the same.

9. Set up the cluster, port, and server information on both machines.

   Note: For mutual high availability configuration (Figure 14 on page 46), for example, configure the cluster sets shared between the 2 Dispatchers as follows:

   - For Dispatcher 1 issue:

     ```
     dscontrol cluster set clusterA primaryhost NFAdispatcher1
     dscontrol cluster set clusterB primaryhost NFAdispatcher2
     ```

- For Dispatcher 2 issue:

```
dscontrol cluster set clusterB primaryhost NFAdispatcher2
dscontrol cluster set clusterA primaryhost NFAdispatcher1
```

10. Start the manager and advisors on both machines.

Notes:

1. To configure a single Dispatcher machine to route packets without a backup, do not issue any of the high availability commands at startup.
2. To convert two Dispatcher machines configured for high availability to one machine running alone, stop the executor on one of the machines, then delete the high availability features (the heartbeats, reach, and backup) on the other.
3. In both of the two cases above, you must alias the network interface card with cluster addresses, as required.
4. When two Dispatcher machines are run in high availability configuration and are synchronized, enter all dscontrol commands (to update the configuration) on the standby machine first, and then on the active machine.
5. When running two Dispatcher machines in a high availability configuration, unexpected results may occur if you set any of the parameters for the executor, cluster, port, or server (for example, `port stickytime`) to different values on the two machines.
6. For mutual high availability, consider the case where one of the Dispatchers must actively route packets for its primary cluster as well as take over routing packets for the backup cluster. Ensure this will not exceed your capacity for throughput on this machine.
7. For Linux systems, when configuring high availability and collocation together when using the Dispatcher component's MAC port forwarding method, see “Linux loopback aliasing alternatives when using Load Balancer's mac forwarding” on page 60.
8. For tips to help alleviate problems that might arise from high availability configuration problems such as:
   - Connections dropped after takeover
   - Partner machines unable to synchronize
   - Requests erroneously directed to the backup partner machine

   See “Problem: Tips on configuring high availability” on page 263.

Failure detection capability using heartbeat and reach target

Besides the basic criteria of failure detection (the loss of connectivity between active and standby Dispatchers, detected through the heartbeat messages), there is another failure detection mechanism named *reachability criteria*. When you configure the Dispatcher you can provide a list of hosts that each of the Dispatchers should be able to reach in order to work correctly. The two high availability partners continually communicate with each other through heartbeats, and they update one another on how many reach targets either one of them can ping. If the standby pings more reach targets than the active, a failover occurs.

Heartbeats are sent by the active Dispatcher and are expected to be received by the standby Dispatcher every half second. If the standby Dispatcher fails to receive a heartbeat within 2 seconds, a failover begins. All heartbeats must break for a takeover from the standby Dispatcher to occur. In other words, when two heartbeat pairs are configured, both heartbeats must break. To stabilize a high availability environment and to avoid failover, add more than one heartbeat pair.

For reach targets, you should choose at least one host for each subnet your Dispatcher machine uses. The hosts could be routers, IP servers or other types of hosts. Host reachability is obtained by the reach advisor, which pings the host. Failover takes place either if the heartbeat messages cannot go through, or if the reachability criteria are met better by the standby Dispatcher than by the primary Dispatcher. To make the decision based on all available information, the active Dispatcher regularly sends the standby Dispatcher its reachability capabilities. The standby Dispatcher then compares those capabilities with its own and decides whether to switch.

Note: When you configure the reach target, the *reach advisor* must also be started. The reach advisor starts automatically when you start the manager function. For more information on the reach advisor, see page 151.

Recovery Strategy

Two Dispatcher machines are configured: the primary machine, and a second machine called the *backup*. At startup, the primary machine sends all the connection data to the backup machine until that machine is synchronized. The primary machine becomes *active*, that is, it begins load balancing. The backup machine, meanwhile, monitors the status of the primary machine, and is said to be in *standby* state.

If the backup machine at any point detects that the primary machine has failed, it performs a *takeover* of the primary machine's load balancing functions and becomes the active machine. After the primary machine has once again become operational, the machines respond according to how the recovery *strategy* has been configured by the user. There are two kinds of strategy:

Automatic
The primary machine resumes routing packets as soon as it becomes operational again.

Manual
The backup machine continues routing packets even after the primary becomes operational. Manual intervention is required to return the primary machine to active state and reset the backup machine to standby.

The strategy parameter must be set the same for both machines.

The manual recovery strategy allows you to force the routing of packets to a particular machine, using the takeover command. Manual recovery is useful when maintenance is being performed on the other machine. The automatic recovery strategy is designed for normal unattended operation.

For a mutual high availability configuration, there is no per cluster failure. If any problem occurs with one machine, even if it affects just one cluster, then the other machine will take over for both clusters.

Note: During takeover situations, some connection updates may be lost. This may cause existing long-running connections (such as telnet) that are being accessed at the time of the takeover to end.

Using scripts

For Dispatcher to route packets, each cluster address must be aliased to a network interface device.

- In a stand-alone Dispatcher configuration, each cluster address must be aliased to a network interface card (for example, en0, tr0).
- In a high availability configuration:
  - On the active machine, each cluster address must be aliased to a network interface card (for example, en0, tr0).
  - On the standby machine, each cluster address must be aliased to a loopback device (for example, lo0) only if you are using a MAC forwarding method with collocated servers.
- In any machine in which the executor has been stopped, all aliases should be removed to prevent conflicts with another machine that may be started.

For information on aliasing the network interface card, see "Step 5. Alias the network interface card" on page 53.

Because the Dispatcher machines will change states when a failure is detected, the commands above must be issued automatically. Dispatcher will run user-created scripts to do that. Sample scripts can be found in the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/samples**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\samples**

These scripts *must* be moved to the following directory in order to run:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/bin**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\bin**

The scripts will run automatically only if dsserver is running.

Notes:

1. For a mutual high availability configuration, each "go" script is called by the Dispatcher with a parameter identifying the primary Dispatcher address. The script must query this parameter and perform the **executor configure** commands for those cluster addresses associated with that primary Dispatcher.
2. In order to configure high availability for Dispatcher's nat forwarding method, you must add the return addresses to the script files.

The following sample scripts may be used:

goActive

The goActive script runs when a Dispatcher goes into active state and begins routing packets.

- If you run Dispatcher in a high availability configuration, you must create this script. This script deletes loopback aliases and adds device aliases.
- If you run Dispatcher in a stand-alone configuration, you do not need this script.

goStandby

The goStandby script runs when a Dispatcher goes into standby state monitoring the health of the active machine, but not routing any packets.

- If you run Dispatcher in a high availability configuration, you must create this script. This script should delete device aliases and add loopback aliases.
- If you run Dispatcher in a stand-alone configuration, you do not need this script.

goInOp

The goInOp script runs when a Dispatcher executor is stopped.

- If you normally run Dispatcher in a high availability configuration, you may create this script. This script deletes all devices and loopback aliases.
- If you normally run Dispatcher in a stand-alone configuration, this script is optional. You may create it and have it delete device aliases, or you may choose to delete them manually.

goIdle The goIdle script runs when a Dispatcher goes into idle state and begins routing packets. This occurs when the high availability features have not been added, as in a stand-alone configuration. It also occurs in a high availability configuration before the high availability features have been added or after they have been removed.

- If you normally run Dispatcher in a high availability configuration, you should *not* create this script.
- If you normally run Dispatcher in a stand-alone configuration, this script is optional. You may create it and have it add device aliases, or you may choose to add them manually. If you do not create this script for your stand-alone configuration, you will have to use the **dscontrol executor configure** command or manually configure the aliases each time the executor is started.

highavailChange

The highavailChange script runs whenever the high availability state changes within the Dispatcher, such that one of the "go" scripts is called. The single parameter passed to this script is the name of the "go" script just run by Dispatcher. You can create this script to use state change information, for instance, to alert an Administrator or simply record the event.

On Windows systems: In your configuration setup, if you have Site Selector load balancing two Dispatcher machines that are operating in a high availability environment, you will need to add an alias on the Microsoft stack for the metric servers. This alias should be added to the goActive script. For example:

```
call netsh interface ip add address "Local Area Connection"
  addr=9.37.51.28 mask=255.255.240.0
```

In the goStandby and goInOp, the alias will need to be removed. For example:

```
call netsh interface ip delete address "Local Area Connection"
  addr=9.37.51.28
```

If there are multiple NIC's on the machine, then first check which interface you should use by issuing the following command on the command prompt: `netsh interface ip show address`. This command will return a list of currently configured interfaces and will number the "Local Area Connection" (for example, "Local Area Connection 2") so you can determine which one you should use.

On Linux for S/390®: Dispatcher issues a gratuitous ARP to move IP addresses from one Dispatcher to another. This mechanism is therefore tied to the underlying network type. When running Linux for S/390, Dispatcher can natively do high availability takeovers (complete with IP address moves) only on those interfaces which can issue a gratuitous ARP and configure the address on the local interface. This mechanism will not work properly on point-to-point interfaces such as IUCV and CTC and will not work properly in certain configurations of qeth/QDIO.

For those interfaces and configurations where Dispatcher's native IP takeover function will not work properly, the customer may place appropriate commands in the go scripts to manually move the addresses. This will ensure that those network topologies can also benefit from high availability.

Configure rules-based load balancing

You can use rules-based load balancing to fine tune when and why packets are sent to which servers. Load Balancer reviews any rules you add from first priority to last priority, stopping on the first rule that it finds to be true, then load balancing the content between any servers associated with the rule. It already balances the load based on destination and port, but using rules expands your ability to distribute connections.

In most cases when configuring rules, you should configure a default **always true** rule in order to catch any request that is passed by other higher priority rules. This default can be a "Sorry, the site is currently down, try again later" response when all other servers fail for the client request.

You should use rules-based load balancing with Dispatcher and Site Selector when you want to use a subset of your servers for some reason. You *must* always use rules for the CBR component.

You can choose from the following types of rules:

- For Dispatcher:
  - Client IP address
  - Client port
  - Time of day
  - Type of service (TOS)
  - Connections per second
  - Active connections total
  - Reserved bandwidth
  - Shared bandwidth
  - Always true
  - Content of a request
- For CBR:
  - Client IP address
  - Time of day
  - Connections per second
  - Active connections total
  - Always true
  - Content of a request
- For Site Selector:
  - Client IP address
  - Time of day
  - Metric all
  - Metric average
  - Always true

Make a plan of the logic that you want the rules to follow before you start adding rules to your configuration.

How are rules evaluated?

All rules have a name, type, priority, and may have a begin range and end range, along with a set of servers. In addition, the content type rule for the CBR component has a matching regular expression pattern associated with it. (For examples and scenarios on how to use the content rule and valid pattern syntax for the content rule, see Appendix B, “Content rule (pattern) syntax,” on page 409.)

Rules are evaluated in priority order. In other words, a rule with a priority of 1 (lower number) is evaluated before a rule with a priority of 2 (higher number). The first rule that is satisfied will be used. When a rule has been satisfied, no further rules are evaluated.

For a rule to be satisfied, it must meet two conditions:

1. The predicate of the rule must be true. That is, the value it is evaluating must be between the begin and end ranges, or the content must match the regular expression specified in the content rule's pattern. For rules of type “true," the predicate is always satisfied, regardless of the begin and end ranges.
2. If there are servers associated with the rule, at least one server must have a weight greater than `0` to forward packets to.

If a rule has no servers associated with it, the rule only needs to meet condition one to be satisfied. In this case, Dispatcher will drop the connection request, Site Selector will return the name server request with an error, and CBR will cause Caching Proxy to return an error page.

If no rules are satisfied, Dispatcher will select a server from the full set of servers available on the port, Site Selector will select a server from the full set of servers available on the site name, and CBR will cause Caching Proxy to return an error page.

Using rules based on the client IP address

This rule type is available in the Dispatcher, CBR, or Site Selector component.

You may want to use rules based on the client IP address if you want to screen the customers and allocate resources based on where they are coming from.

For example, you notice that your network is getting a lot of unpaid and therefore unwanted traffic from clients coming from a specific set of IP addresses. You create a rule using the **dscontrol rule** command, for example:

```
dscontrol rule add 9.67.131.153:80:ni type ip
  beginrange 9.0.0.0 endrange 9.255.255.255
```

This "ni" rule screens out any connection from unwanted clients. You would then add to the rule the servers that you want accessible, or if you do not add any servers to the rule, requests coming from 9.x.x.x addresses are not served by any of your servers.

Using rules based on the client port

This rule type is only available in the Dispatcher component.

You may want to use rules based on the client port if your clients are using some kind of software that asks for a specific port from TCP/IP when making requests.

For example, you could create a rule that says that any request with a client port of 10002 will get to use a set of special fast servers because you know that any client request with that port is coming from an elite group of customers.

Using rules based on the time of day

This rule type is available in the Dispatcher, CBR, or Site Selector component.

You may want to use rules based on the time of day for capacity planning reasons. For example, if your Web site gets hit most during the same group of hours every day, you might want to dedicate five additional servers during the peak time period.

Another reason you might use a rule based on the time of day is when you want to take some of the servers down for maintenance every night at midnight, so you can set up a rule that excludes those servers during the necessary maintenance period.

Using rules based on type of service (TOS)

This rule type is only available in the Dispatcher component.

You may want to use rules based on the content of the "type of service" (TOS) field in the IP header. For example, if a client request comes in with one TOS value that indicates normal service, it can be routed to one set of servers. If a different client request comes in with a different TOS value that indicates a higher priority of service, it can be routed to a different set of servers.

The TOS rule allows you to fully configure each bit in the TOS byte using the **dscontrol rule** command. For significant bits that you want matched in the TOS byte, use 0 or 1. Otherwise, the value x is used. The following is an example for adding a TOS rule:

```
dscontrol rule add 9.67.131.153:80:tsr type service tos 0xx1010x
```

Using rules based on the connections per second

This rule type is available in the Dispatcher and CBR components.

Note: The manager must be running for the following to work.

You may want to use rules based on connections per second if you need to share some of your servers with other applications. For example, you can set two rules:

1. If connections per second on port 80 is between 0 and 2000, then use these 2 servers
2. If connections per second on port 80 is greater than 2000, then use these 10 servers

Or you might be using Telnet and want to reserve two of your five servers for Telnet, except when the connections per second increases above a certain level. This way, Dispatcher would balance the load across all five servers at peak times.

Setting rule evaluate option "upserversonrule" in conjunction with the "connection" type rule: When using the connections type rule and setting the **upserversonrule** option, if some of the servers in the server set are down, then you

can ensure that the remaining servers will not be overloaded. See “Server evaluation option for rules” on page 177 for more information.

Using rules based on the total active connections

This rule type is available in the Dispatcher or CBR component.

Note: The manager must be running for the following to work.

You may want to use rules based on active connections total on a port if your servers get overloaded and start throwing packets away. Certain Web servers will continue to accept connections even though they do not have enough threads to respond to the request. As a result, the client requests time out and the customer coming to your Web site is not served. You can use rules based on active connections to balance capacity within a pool of servers.

For example, you know from experience that your servers will stop serving after they have accepted 250 connections. You can create a rule using the **dscontrol rule** command or the **cbrcontrol rule** command, for example:

```
dscontrol rule add 130.40.52.153:80:pool2 type active
  beginrange 250 endrange 500

or

cbrcontrol rule add 130.40.52.153:80:pool2 type active
  beginrange 250 endrange 500
```

You would then add to the rule your current servers plus some additional servers, which will otherwise be used for other processing.

Using rules based on reserved bandwidth and shared bandwidth

Reserved bandwidth and shared bandwidth rules are only available in the Dispatcher component.

For bandwidth rules, Dispatcher calculates bandwidth as the rate at which data is delivered to clients by a specific set of servers. Dispatcher tracks capacity at the server, rule, port, cluster, and executor levels. For each of these levels, there is a byte counter field: kilobytes transferred per second. Dispatcher calculates these rates over a 60 second interval. You can view these rate values from the GUI or from the output of a command line report.

Reserved bandwidth rule

The reserved bandwidth rule allows you to control the number of kilobytes per second being delivered by a set of servers. By setting a threshold (allocating a specified bandwidth range) for each set of servers throughout the configuration, you can control and guarantee the amount of bandwidth being used by each cluster-port combination.

The following is an example for adding a reservedbandwidth rule:

```
dscontrol rule add 9.67.131.153:80:rbw type reservedbandwidth
  beginrange 0 endrange 300
```

The begin range and end range are specified in kilobytes per second.

Shared bandwidth rule

Prior to configuring the shared bandwidth rule, you must specify the maximum amount of bandwidth (kilobytes per second) that can be shared at the executor or cluster level using **dscontrol executor** or **dscontrol cluster** command with the sharedbandwidth option. The sharebandwidth value should not exceed the total bandwidth (total network capacity) available. Using the **dscontrol** command to set shared bandwidth only provides an upper limit for the rule.

The following are examples of the command syntax:

```
dscontrol executor set sharedbandwidth size
dscontrol cluster [add | set] 9.12.32.9 sharedbandwidth size
```

The *size* for sharedbandwidth is an integer value (kilobytes per second). The default is zero. If the value is zero, then bandwidth cannot be shared.

Sharing bandwidth at the cluster level allows a maximum specified bandwidth to be used by the cluster. As long as the bandwidth used by the cluster is below the specified amount, then this rule will evaluate as true. If the total bandwidth used is greater than the specified amount, then this rule will evaluate as false.

Sharing bandwidth at the executor level allows the entire Dispatcher configuration to share a maximum amount of bandwidth. As long as the bandwidth used at the executor level is below the specified amount, then this rule will evaluate as true. If the total bandwidth used is greater than that defined, then this rule will evaluate as false.

The following are examples of adding or setting a sharedbandwidth rule:

```
dscontrol rule add 9.20.30.4:80:shbw type sharedbandwidth sharelevel value
dscontrol rule set 9.20.34.11:80:shrule sharelevel value
```

The *value* for sharelevel is either executor or cluster. Sharelevel is a required parameter on the sharebandwidth rule.

Using reserved and shared bandwidth rules

Dispatcher allows you to allocate a specified bandwidth to sets of servers within you configuration using the *reserved bandwidth* rule. By specifying a begin and end range, you can control the range of kilobytes delivered by a set of servers to the clients. When the rule no longer evaluates as true (the end range is exeeded), the next lower priority rule is evaluated. If the next lower priority rule is an "always true" rule, a server could be selected to respond to the client with a "site busy" response.

For example: Consider a group of three servers on port 2222. If the reserved bandwidth is set to 300, then the maximum kbytes per second is 300, over a period of 60 seconds. When this rate is exceeded, then the rule no longer evaluates as true. If this were the only rule, then one of the three servers would be selected by Dispatcher to handle the request. If there were a lower priority "always true" rule, then the request could be redirected to another server and answered with "site busy".

The shared bandwidth rule can provide additional server access to clients. Specifically, when used as a lower priority rule following a reserved bandwidth rule, a client can still access a server even though the reserved bandwidth has been exceeded.

For example: By using a shared bandwidth rule following a reserved bandwidth rule you can allow clients to gain access to the three servers in a controlled manner. As long as there is shared bandwidth available to be used, the rule will evaluate as true and access is granted. If there is no shared bandwidth available, then the rule is not true and the next rule is evaluated. If an "always true" rule follows, the request can be redirected as needed.

By using both reserved and shared bandwidth as described in the preceding example, greater flexibility and control can be exercised in granting (or denying) access to the servers. Servers on a specific port can be limited in bandwidth usage, while others can use additional bandwidth as long as it is available.

Note: Dispatcher tracks bandwidth by measuring client traffic, such as data "acks", that flow to a server. If for any reason this traffic is not "seen" by Dispatcher, results are unpredictable when using the bandwidth rules.

Metric all rule

This rule type is only available in the Site Selector component.

For the metric all rule, you choose a system metric (cpuload, memload, or your own customized system metric script), and Site Selector compares the system metric value (returned by the Metric Server agent residing in each load-balanced server) with the begin and end range that you specify in the rule. The current system metric value for all the servers in the server set must be within the range for the rule to run.

Note: The system metric script you choose must reside on each of the load-balanced servers.

The following is an example of adding a metric all rule to your configuration:

```
sscontrol rule add dnsload.com:allrule1 type metricall
  metricname cpuload beginrange 0 endrange 100
```

Metric average rule

This rule type is only available in the Site Selector component.

For the metric average rule, you choose a system metric (cpuload, memload, or your own customized system metric script), and Site Selector compares the system metric value (returned by the Metric Server agent residing in each load-balanced server) with the begin and end range that you specify in the rule. The *average* of the current system metric values for all the servers in the server set must be within the range for the rule to run.

Note: The system metric script you choose must reside on each of the load-balanced servers.

The following is an example of adding a metric average rule to your configuration:

```
sscontrol rule add dnsload.com:avgrule1 type metricavg
  metricname cpuload beginrange 0 endrange 100
```

Using rules that are always true

This rule type is available in the Dispatcher, CBR, or Site Selector component.

A rule may be created that is “always true.” Such a rule will always be selected, unless all the servers associated with it are down. For this reason, it should ordinarily be at a lower priority than other rules.

You can even have multiple “always true” rules, each with a set of servers associated with it. The first true rule with an available server is chosen. For example, assume you have six servers. You want two of them to handle your traffic under all circumstances, unless they are both down. If the first two servers are down, you want a second set of servers to handle the traffic. If all four of these servers are down, then you will use the final two servers to handle the traffic. You could set up three “always true” rules. Then the first set of servers will always be chosen as long as at least one is up. If they are both down, one from the second set is chosen, and so forth.

As another example, you may want an “always true” rule to ensure that if incoming clients do not match any of the rules you have set, they will not be served. You would create a rule using the **dscontrol rule** command like:

```
dscontrol rule add 130.40.52.153:80:jamais type true priority 100
```

Then you would not add any servers to the rule, causing the clients packets to be dropped with no response.

Note: You do not need to set a beginrange or endrange when creating an always true rule.

You can define more than one “always true” rule, and thereafter adjust which one gets run by changing their priority levels.

Using rules based on the request content

This rule type is available in the CBR component or Dispatcher component (when using Dispatcher's cbr forwarding method).

You will want to use content type rules to send requests to sets of servers specifically set up to handle some subset of your site's traffic. For example, you may want to use one set of servers to handle all *cgi-bin* requests, another set to handle all streaming audio requests, and a third set to handle all other requests. You would add one rule with a pattern that matches the path to your cgi-bin directory, another that matches the file type of your streaming audio files, and a third always true rule to handle the rest of the traffic. You would then add the appropriate servers to each of the rules.

Important: For examples and scenarios on how to use the content rule and valid pattern syntax for the content rule, see Appendix B, “Content rule (pattern) syntax,” on page 409.

port affinity override

With port affinity override, you can override the stickiness of a port for a specific server. For example, you are using a rule to limit the amount of connections to each application server, and you have an overflow server with an always true rule that says “please try again later" for that application. The port has a stickytime value of 25 minutes, so you do not want the client to be sticky to that server. With port affinity override, you can change the overflow server to override the affinity normally associated with that port. The next time the client requests the cluster, it is load balanced to the best available application server, not the overflow server.

See “dscontrol server — configure servers” on page 330, for detailed information on command syntax for the port affinity override, using the server **sticky** option.

Adding rules to your configuration

You can add rules using the **dscontrol rule add** command, by editing the sample configuration file, or with the graphical user interface (GUI). You can add one or more rules to every port you have defined.

It is a two–step process: add the rule, then define which servers to serve to if the rule is true. For example, our system administrator wanted to track how much use the proxy servers were getting from each division on site. IP addresses are given to each division. Create the first set of rules based on client IP address to separate each division's load:

```
dscontrol rule add 130.40.52.153:80:div1 type ip b 9.1.0.0 e 9.1.255.255
dscontrol rule add 130.40.52.153:80:div2 type ip b 9.2.0.0 e 9.2.255.255
dscontrol rule add 130.40.52.153:80:div3 type ip b 9.3.0.0 e 9.3.255.255
```

Next, add a different server to each rule, then measure the load on each of the servers in order to bill the division properly to the services they are using. For example:

```
dscontrol rule useserver 130.40.52.153:80:div1 207.72.33.45
dscontrol rule useserver 130.40.52.153:80:div2 207.72.33.63
dscontrol rule useserver 130.40.52.153:80:div3 207.72.33.47
```

Server evaluation option for rules

The server evaluation option is only available in the Dispatcher component.

On the **dscontrol rule** command there is a server evaluation option for rules. Use the *evaluate* option to choose to evaluate the rule's condition across all the servers on the port or to evaluate the rule's condition across just the servers within the rule. (In earlier versions of Load Balancer, you could only measure each rule's condition across all servers on the port.)

Notes:

1. The server evaluation option is only valid for rules that make their decisions based upon the characteristics of the servers: total connections (per second) rule, active connections rule, and reserved bandwidth rule.
2. The "connection" type rule has an additional evaluate option to choose — **upserversonrule**. See “Using rules based on the connections per second” on page 172 for more information.

The following are examples of adding or setting the evaluate option on a reserved bandwidth rule:

```
dscontrol rule add 9.22.21.3:80:rbweval type reservedbandwidth evaluate level
dscontrol rule set 9.22.21.3:80:rbweval evaluate level
```

The evaluate *level* can be set to either port, rule, or upserversonrule. The default is port.

Evaluate servers within the rule

The option to measure the rule's condition across the servers within the rule allows you to configure two rules with the following characteristics:

- The first rule that gets evaluated contains all the servers maintaining the Web site content, and the evaluate option is set to *rule* (evaluate the rule's condition across the servers within the rule).

- The second rule is an always true rule that contains a single server that responds with a “site busy" type response.

The result is that when traffic exceeds the threshold of the servers within the first rule, traffic is sent to the “site busy" server within the second rule. When traffic falls below the threshold of the servers within the first rule, new traffic continues once again to the servers in the first rule.

Evaluate servers on the port

Using the two rules described in the previous example, if you set the evaluate option to *port* for the first rule (evaluate rule's condition across all the servers on the port), when traffic exceeds the threshold of that rule, traffic is sent to the “site busy" server associated to the second rule.

The first rule measures all server traffic (including the “site busy" server) on the port to determine whether the traffic exceeds the threshold. As congestion decreases for the servers associated to the first rule, an unintentional result may occur where traffic continues to the “site busy" server because traffic on the port still exceeds the threshold of the first rule.

How affinity feature for Load Balancer works

For the Dispatcher and CBR components: You enable the affinity feature when you configure a cluster's port to be sticky. Configuring a cluster's port to be sticky allows subsequent client requests to be directed to the same server. This is done by setting **stickytime** at the executor, cluster, or port level to some number of seconds. The feature is disabled by setting stickytime to zero.

If you are enabling cross port affinity, stickytime values of the shared ports must be the same (nonzero) value. See “Cross port affinity” on page 179 for more information.

For the Site Selector component: You enable the affinity feature when you configure a sitename to be sticky. Configuring a sitename to be sticky allows the client to use the same server for multiple name service requests. This is done by setting **stickytime** on the sitename to some number of seconds. The feature is disabled by setting stickytime to zero.

A sticky time value for a server is the interval between the closing of one connection and the opening of a new connection during which time a client is sent back to the same server used during the first connection. After the sticky time expires, the client can be sent to a server different from the first. The sticky time value for a server is configured using the dscontrol executor, port, or cluster commands.

Behavior when affinity is disabled

With the affinity feature disabled, whenever a new TCP connection is received from a client, Load Balancer picks the right server at that moment in time and forwards the packets to it. If a subsequent connection comes in from the same client, Load Balancer treats it as an unrelated new connection, and again picks the right server at that moment in time.

Behavior when affinity is enabled

With the affinity feature enabled, if a subsequent request is received from the same client, the request is directed to the same server.

Over time, the client will finish sending transactions, and the affinity record will go away. Hence the meaning of the sticky "time." Each affinity record lives for the "stickytime" in seconds. When subsequent connections are received within the stickytime, the affinity record is still valid and the request will go to the same server. If a subsequent connection is not received within stickytime, the record is purged; a connection that is received after that time will have a new server selected for it.

The server down command (`dscontrol server down`) is used to bring a server offline The server is not taken down until after the stickytime value expires.

Cross port affinity

Cross port affinity only applies to the Dispatcher component's MAC and NAT/NATP forwarding methods.

Cross port affinity is the sticky feature that has been expanded to cover multiple ports. For example, if a client request is first received on one port and the next request is received on another port, cross port affinity allows the dispatcher to send the client request to the same server. In order to use this feature, the ports must:

- share the same cluster address
- share the same servers
- have the same (nonzero) **stickytime** value
- have the same **stickymask** value

More than one port can link to the same **crossport**. When subsequent connections come in from the same client on the same port or a shared port, the same server will be accessed. The following is an example of configuring multiple ports with a cross port affinity to port 10:

```
dscontrol port set cluster:20 crossport 10
dscontrol port set cluster:30 crossport 10
dscontrol port set cluster:40 crossport 10
```

After cross port affinity has been established, you have the flexibility to modify the stickytime value for the port. However, it is recommended that you change the stickytime values for all shared ports to the same value, otherwise unexpected results may occur.

To remove the cross port affinity, set the crossport value back to its own port number. See “dscontrol port — configure ports” on page 319, for detailed information on command syntax for the **crossport** option.

Affinity address mask (stickymask)

Affinity address mask only applies to the Dispatcher component.

Affinity address mask is a sticky feature enhancement to group clients based upon common subnet addresses. Specifying **stickymask** on the **dscontrol port** command allows you to mask the common high-order bits of the 32-bit IP address. If this feature is configured, when a client request first makes a connection to the port, all subsequent requests from clients with the same subnet address (represented by that part of the address which is being masked) will be directed to the same server.

Note: In order to enable stickymask, port **stickytime** must be a nonzero value.

For example, if you want all incoming client requests with the same network Class A address to be directed to the same server, you set the stickymask value to 8 (bits) for the port. To group client requests with the same network Class B address, set the stickymask value to 16 (bits). To group client requests with the same network Class C address, set the stickymask value to 24 (bits).

For best results, set the stickymask value when first starting the Load Balancer. If you change the stickymask value dynamically, results will be unpredictable.

Interaction with cross port affinity: If you are enabling cross port affinity, stickymask values of the shared ports must be the same. See "Cross port affinity" on page 179 for more information.

To enable affinity address mask, issue an **dscontrol port** command similar to the following:

```
dscontrol port set cluster:port stickytime 10 stickymask 8
```

Possible stickymask values are 8, 16, 24 and 32. A value of 8 specifies the first 8 high-order bits of the IP address (network Class A address) will be masked. A value of 16 specifies the first 16 high-order bits of the IP address (network Class B address) will be masked. A value of 24 specifies the first 24 high-order bits of the IP address (network Class C address) will be masked. If you specify a value of 32, you are masking the entire IP address which effectively disables the affinity address mask feature. The default value of stickymask is 32.

See "dscontrol port — configure ports" on page 319, for detailed information on command syntax for stickymask (affinity address mask feature).

Quiesce server connection handling

Quiesce handling applies to the Dispatcher and CBR components.

To remove a server from the Load Balancer configuration for any reason (updates, upgrades, service, and so forth), you can use the **dscontrol manager quiesce** command. The quiesce subcommand allows existing connections to complete (without being severed) and forwards only subsequent new connections from the client to the quiesced server if the connection is designated as sticky and stickytime has not expired. The quiesce subcommand disallows any other new connections to the server.

Quiesce handling for sticky connections

Use the quiesce "now" option if you have stickytime set, and you want new connections sent to another server (instead of the quiesced server) before stickytime expires. The following is an example of using the now option to quiesce server 9.40.25.67:

```
dscontrol manager quiesce 9.40.25.67 now
```

The now option determines how sticky connections will be handled as follows:

- If you do *not* specify "now," you allow existing connections to complete and forward subsequent new connections to the quiesced server from those clients with existing connections that are designated as sticky, as long as the quiesced server receives the new request before stickytime expires. (However, if you have not enabled the sticky (affinity) feature, the quiesced server cannot receive any new connections.)

This is the more graceful, less abrupt, way to quiesce servers. For instance, you can gracefully quiesce a server and then wait for the time where there is the least amount of traffic (perhaps early morning) to completely remove the server from the configuration.

- By specifying “now," you quiesce the server so it allows existing connections to complete but disallows all new connections including subsequent new connections from those clients with existing connections that are designated as sticky. This is the more abrupt way to quiesce servers, which was the only way it was handled in earlier versions of the Load Balancer.

Affinity option on the rule based on the content of the client request

You can specify the following types of affinity on the **dscontrol rule** command:

- Active cookie — enables load-balancing Web traffic with affinity to the same server based upon cookies generated by Load Balancer.

  Active cookie affinity only applies to the CBR component.
- Passive cookie — enables load-balancing Web traffic with affinity to the same server based upon self-identifying cookies generated by the servers. In conjunction with passive cookie affinity, you must also specify the cookiename parameter on the rule command.

  Passive cookie applies to the CBR component and to Dispatcher component's cbr forwarding method.
- URI — enables load-balancing Web traffic to caching-proxy servers in a manner that effectively increases the capacity of the cache.

  URI affinity applies to the CBR component and to Dispatcher component's cbr forwarding method.

The default for the affinity option is "none." The **stickytime** option on the port command must be zero (not enabled) in order to set the **affinity** option on the rule command to active cookie, passive cookie, or URI. When affinity is set on the rule, you cannot enable stickytime on the port.

Active cookie affinity

The active cookie affinity feature applies only to the CBR component.

It provides a way to make clients “sticky” to a particular server. This function is enabled by setting the **stickytime** of a rule to a positive number, and setting the affinity to “activecookie.” This can be done when the rule is added, or using the rule set command. See “dscontrol rule — configure rules” on page 324, for detailed information on command syntax.

When a rule has been enabled for active cookie affinity, new client requests are load-balanced using standard CBR algorithms, while succeeding requests from the same client are sent to the initially chosen server. The chosen server is stored as a cookie in the response to the client. As long as the client's future requests contains the cookie, and each request arrives within the stickytime interval, the client will maintain affinity with the initial server.

Active cookie affinity is used to ensure that a client continues to be load balanced to the same server for some period of time. This is accomplished by sending a cookie to be stored by the clients browser. The cookie contains the cluster:port:rule that was used to make the decision, the server that was load balanced to, and a timeout timestamp for when the affinity is no longer valid. The cookie is in the

following format: **IBMCBR=***cluster***:***port***:***rule***+***server***-***time***!** The *cluster:port:rule* and *server* information are encoded so the CBR configuration is not revealed.

How active cookie affinity works

Whenever a rule fires that has active cookie affinity turned on, the cookie sent by the client is examined.

- If a cookie is found that contains the identifier for the cluster:port:rule that fired, then the server load balanced to, and the expires timestamp are extracted from the cookie.
- If the server is still in the set used by the rule, and its weight is positive or it is a quiesced server, and the expires timestamp is greater than now, then the server in the cookie is chosen to load balance to.
- If any of the conditions in the previous bullet are not met, a server is chosen using the normal algorithm.
- When a server has been chosen (using either of the two methods) a new cookie is constructed containing IBMCBR, cluster:port:rule, server_chosen information, and a timestamp. The timestamp is the time that affinity expires. The "cluster:port:rule and server_chosen" are encoded so that no information about the CBR configuration is revealed.
- An "expires" parameter is also inserted in the cookie. This parameter is in a format the browser can understand, and causes the cookie to become invalid seven days after the expires timestamp. This is so the client's cookie database does not become cluttered.

This new cookie is then inserted in the headers that go back to the client, and if the client's browser is configured to accept cookies, it will send back subsequent requests.

Each affinity instance in the cookie is 65 bytes in length and end at the exclamation mark. As a result, a 4096 byte cookie can hold approximately 60 individual active cookie rules per domain. If the cookie fills up completely, then all expired affinity instances are purged. If all instances are still valid, then the oldest one is dropped, and the new instances for the current rule is added.

Note: CBR will replace any occurrences of old format IBMCBR cookies as they appear in the proxy.

The active cookie affinity option, for the rule command, can only be set to activecookie if port stickytime is zero (not enabled). When active cookie affinity is active on a rule then you cannot enable stickytime on the port.

How to enable active cookie affinity

To enable active cookie affinity for a particular rule, use the rule set command:

```
rule set cluster:port:rule stickytime 60
rule set cluster:port:rule affinity activecookie
```

Why use active cookie affinity

Making a rule sticky would normally be used for CGI or servlets that store client state on the server. The state is identified by a cookie ID (these are server cookies). Client state is only on the selected server, so the client needs the cookie from that server to maintain that state between requests.

Active cookie affinity expiration time override

Active cookie affinity has a default expiration of the current server time, plus the stickytime interval, plus twenty-four hours. If your clients (those sending requests

to your CBR machine) have inaccurate times on their system (for example, they are more than one day ahead of the server time), then those clients' systems will ignore the cookies from CBR because the system will assume that the cookies have already expired. To set a longer expiration time, modify the cbrserver script. In the script file, edit the javaw line, adding the following parameter after LB_SERVER_KEYS: `-DCOOKIEEXPIREINTERVAL=X` where *X* is the number of days to add to the expiration time.

On AIX, Solaris and Linux systems, the cbrserver file is located in `/usr/bin` directory.

On Windows systems, the cbrserver file is located in `\winnt\system32` directory.

Passive cookie affinity

Passive cookie affinity applies to the Dispatcher component's content-based routing (cbr) forwarding method and to the CBR component. See “Dispatcher's content-based routing (cbr forwarding method)” on page 41 for information on how to configure Dispatcher's cbr forwarding method.

Passive cookie affinity provides a way to make clients sticky to a particular server. When you enable the affinity of a rule to "passivecookie", passive cookie affinity allows you to load-balance Web traffic with affinity to the same server, based on self-identifying cookies generated by the servers. You configure passive cookie affinity at the rule level.

When the rule fires, if passive cookie affinity is enabled, Load Balancer will choose the server based on the cookie name in the HTTP header of the client request. Load Balancer begins to compare the cookie name from the client's HTTP header to the configured cookie value for each server.

The first time Load Balancer finds a server whose cookie value *contains* the client's cookie name, Load Balancer chooses that server for the request.

Note: Load Balancer provides this flexibility in order to handle cases where the server might generate a cookie value that has a static part appended with a variable part. For example, the server's cookie value might be the server name (a static value) appended with a timestamp (a variable value).

If the cookie name in the client request is not found or does not match any of the content within the servers' cookie values, the server is chosen using existing server selection or the weighted round-robin technique.

To configure **passive cookie affinity**:

- For Dispatcher, first configure Dispatcher's cbr forwarding method. (See “Dispatcher's content-based routing (cbr forwarding method)” on page 41.) This step is omitted for the CBR component.
- Set the **affinity** parameter to "passivecookie" on the **dscontrol rule [add|set]** command. Also, the **cookiename** parameter must be set to the name of the cookie that Load Balancer should look for in the client HTTP header request.
- Set the **cookievalue** parameter, for each server in the rule's server set, on the **dscontrol server [add|set]** command.

The passive cookie affinity option, for the rule command, can only be set to passivecookie if port stickytime is zero (not enabled). When passive cookie affinity is active on a rule then you cannot enable stickytime on the port.

URI affinity

URI affinity applies to Dispatcher's cbr forwarding method and the CBR component. See “Dispatcher's content-based routing (cbr forwarding method)” on page 41 for information on how to configure the cbr forwarding method.

URI affinity allows you to load-balance Web traffic to Caching Proxy servers which allow unique content to be cached on each individual server. As a result, you will effectively increase the capacity of your site's cache by eliminating redundant caching of content on multiple machines. Configure URI affinity at the rule level. After the rule fires, if URI affinity is enabled and the same set of servers are up and responding, then Load Balancer will forward new incoming client requests with the same URI to the same server.

Typically, Load Balancer can distribute requests to multiple servers that serve identical content. When using Load Balancer with a group of caching servers, frequently accessed content eventually becomes cached on all the servers. This supports a very high client load by replicating identical cached content on multiple machines. This is particularly useful for high volume Web sites.

However, if your Web site supports a moderate volume of client traffic to very diverse content, and you prefer to have a larger cache spread across multiple servers, your site would perform better if each caching server contained unique content and Load Balancer distributed the request only to the caching server with that content.

With URI affinity, Load Balancer allows you to distribute the cached content to individual servers, eliminating redundant caching of content on multiple machines. Performance for diverse-content server sites using Caching Proxy servers is improved with this enhancement. It will send identical requests to the same server, thereby caching content on single servers only. And, the effective size of the cache will grow larger with each new server machine added to the pool.

To configure **URI affinity**:

- For Dispatcher, first configure Dispatcher's cbr forwarding method. (See “Dispatcher's content-based routing (cbr forwarding method)” on page 41.) This step is omitted for the CBR component.
- Set the **affinity** parameter to "uri" on the **dscontrol rule [add|set]** or **cbrcontrol rule [add|set]** command.

  The URI affinity option, for the rule command, can only be set to URI if port stickytime is zero (not enabled). When URI affinity is active on a rule then you cannot enable stickytime on the port.

Configure wide area Dispatcher support

This feature is only available for the Dispatcher component.

If you are not using the Dispatcher's wide area support and not using Dispatcher's nat forwarding method, a Dispatcher configuration requires that the Dispatcher machine and its servers all be attached to the same LAN segment (see Figure 32 on page 185). A client's request comes into the Dispatcher machine and is sent to the server. From the server, the response is sent directly back to the client.

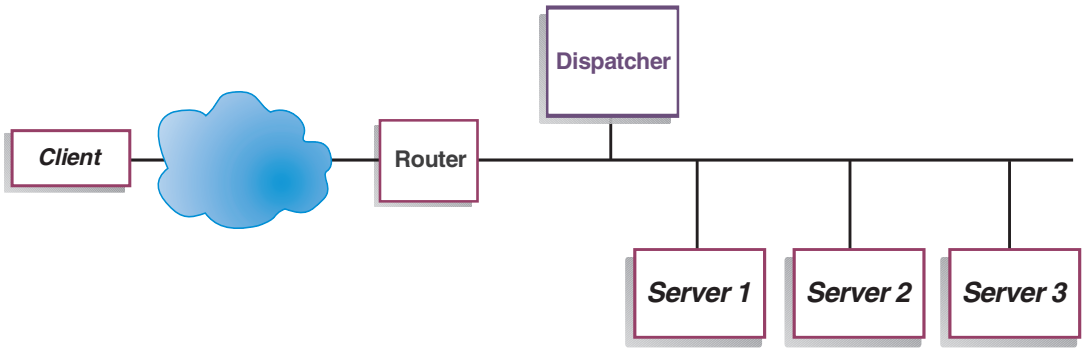


Figure 32. Example of a configuration consisting of a single LAN segment

The wide area Dispatcher feature adds support for offsite servers, known as *remote servers* (see Figure 33). If GRE is not supported at the remote site and if Dispatcher's nat forwarding method is not being used, then the remote site must consist of a remote Dispatcher machine (Dispatcher 2) and its locally attached servers (ServerG, ServerH, and ServerI). A client's packet will go from the Internet to the initial Dispatcher machine. From the initial Dispatcher machine, the packet will then go to a geographically remote Dispatcher machine and one of its locally attached servers.

All the Dispatcher machines (local and remote) must be on the same type of operating system and platform in order to run wide area configurations.

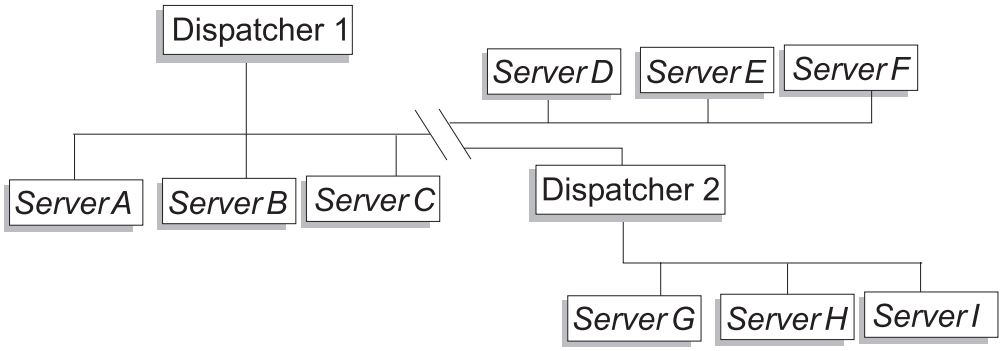


Figure 33. Example of configuration using local and remote servers

This allows one cluster address to support all worldwide client requests while distributing the load to servers around the world.

The Dispatcher machine initially receiving the packet can still have local servers attached to it, and it can distribute the load between its local servers and the remote servers.

Command Syntax

To configure wide area support :

1. Add the servers. When you add a server to a Dispatcher, you must define whether the server is local or remote (see above). To add a server and define it as local, issue the **dscontrol server add** command without specifying a router. This is the default. To define the server as remote, you must specify the router through which Dispatcher must send the packet in order to reach the remote server. The server must be another Dispatcher and the server's address must be

the nonforwarding address of the Dispatcher. For example, in Figure 34 on page 187, if you are adding *LB 2* as a remote server under *LB 1*, you must define *router 1* as the router address. General syntax:

```
dscontrol server add cluster:port:server router address
```

For more information on the router keyword, see "dscontrol server — configure servers" on page 330.

2. Configure aliases. On the first Dispatcher machine (where the client request arrives from the Internet), the cluster address must be aliased using the **executor configure** command. (For Linux or UNIX systems, you can use the **executor configure** or **ifconfig** command.) On the remote Dispatcher machines, however, the cluster address is *not* aliased to a network interface card.

Using remote advisors with Dispatcher's wide area support

On entry-point Dispatchers:

An entry-point dispatcher will treat the second-level Dispatcher as a server, and it will monitor the health of it as a server and tie results to the real IP of the dispatcher.

On remote Dispatchers: Perform the following configuration steps for each remote cluster address. For a high-availability configuration at the remote Dispatcher location, you must perform these steps on both machines.

AIX systems

- Dispatcher must have each cluster configured on the interface with a netmask 255.255.255.255 in order for the advisors to work properly. Use one of the following syntax formats for configuring a cluster:
  - `ifconfig interface_name alias cluster_address netmask 255.255.255.255`. For example,

    ```
    ifconfig en0 alias 10.10.10.99 netmask 255.255.255.255
    ```

  - `dscontrol executor configure interface_address interface_name netmask`. For example,

    ```
    dscontrol executor configure 204.67.172.72 en0 255.255.255.255
    ```

Note: Advisors running on both the local and remote Dispatcher machines are necessary.

HP-UX systems, Linux, Solaris, and Windows systems

- No additional configuration steps are required.

Configuration example

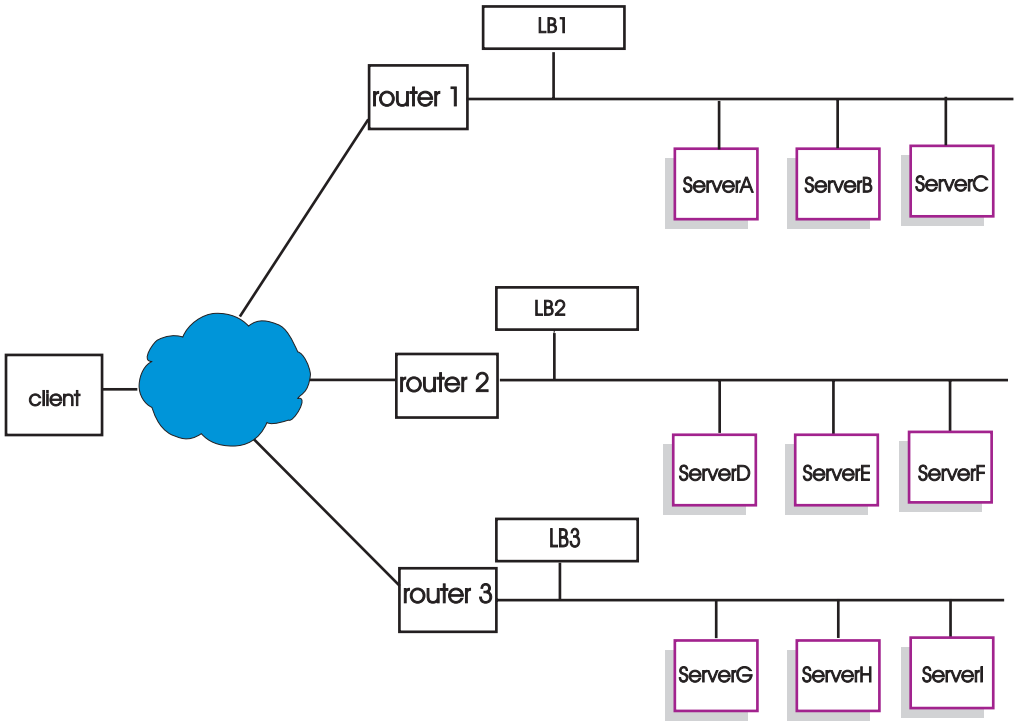


Figure 34. Wide area example configuration with remote Load Balancers

This example applies to the configuration illustrated in Figure 34.

Here is how to configure the Dispatcher machines to support cluster address xebec on port 80. LB1 is defined as the "entry-point" Load Balancer. An Ethernet connection is assumed. Note that LB1 has five servers defined: three local (ServerA, ServerB, ServerC) and two remote (LB2 and LB3). Remotes LB2 and LB3 each have three local servers defined.

At the console of the first Dispatcher (LB1), do the following:

1. Start the executor.
   dscontrol executor start
2. Set the nonforwarding address of the Dispatcher machine.
   dscontrol executor set nfa LB1
3. Define the cluster.
   dscontrol cluster add xebec
4. Define the port.
   dscontrol port add xebec:80
5. Define the servers.
   a. **dscontrol server add xebec:80:ServerA**
   b. **dscontrol server add xebec:80:ServerB**
   c. **dscontrol server add xebec:80:ServerC**
   d. **dscontrol server add xebec:80:LB2 router Router1**
   e. **dscontrol server add xebec:80:LB3 router Router1**
6. Configure the cluster address.

dscontrol executor configure xebec

At the console of the second Dispatcher (LB2):

1. Start the executor.
   dscontrol executor start
2. Set the nonforwarding address of the Dispatcher machine.
   dscontrol executor set nfa LB2
3. Define the cluster.
   dscontrol cluster add xebec
4. Define the port.
   dscontrol port add xebec:80
5. Define the servers.
   a. **dscontrol server add xebec:80:ServerD**
   b. **dscontrol server add xebec:80:ServerE**
   c. **dscontrol server add xebec:80:ServerF**

At the console of the third Dispatcher (LB3):

1. Start the executor.
   dscontrol executor start
2. Set the nonforwarding address of the Dispatcher machine.
   dscontrol executor set nfa LB3
3. Define the cluster.
   dscontrol cluster add xebec
4. Define the port.
   dscontrol port add xebec:80
5. Define the servers.
   a. **dscontrol server add xebec:80:ServerG**
   b. **dscontrol server add xebec:80:ServerH**
   c. **dscontrol server add xebec:80:ServerI**

Notes

1. On all servers (A-I), alias the cluster address to the loopback.
2. Clusters and ports are added with `dscontrol` on all participating Dispatcher machines: the entry-point Dispatcher and all remotes.
3. See "Using remote advisors with Dispatcher's wide area support" on page 186 for help with using remote advisors with wide area support.
4. Wide area support prohibits infinite routing loops. (If a Dispatcher machine receives a packet from another Dispatcher, it will not forward it to a third Dispatcher.) Wide area supports only one level of remotes.
5. Wide area supports UDP and TCP.
6. Wide area works along with high availability: Each Dispatcher may be backed up by an adjacent standby machine (on the same LAN segment).
7. The Manager and Advisors work with wide area, and, if used, should be started on all participating Dispatcher machines.
8. Load Balancer supports WAN only on like operating systems.

GRE (Generic Routing Encapsulation) support

Generic Routing Encapsulation (GRE) is an Internet Protocol specified in RFC 1701 and RFC 1702. Using GRE, the Load Balancer can encapsulate client IP packets inside IP/GRE packets and forward them to server platforms such as OS/390 that support GRE. GRE support allows the Dispatcher component to load balance packets to multiple server addresses associated with one MAC address.

Load Balancer implements GRE as part of its WAN feature. This allows Load Balancer to provide wide area load balancing directly to any server systems that can unwrap the GRE packets. Load Balancer does not need to be installed at the remote site if the remote servers support the encapsulated GRE packets. Load Balancer encapsulates WAN packets with the GRE key field set to decimal value 3735928559.

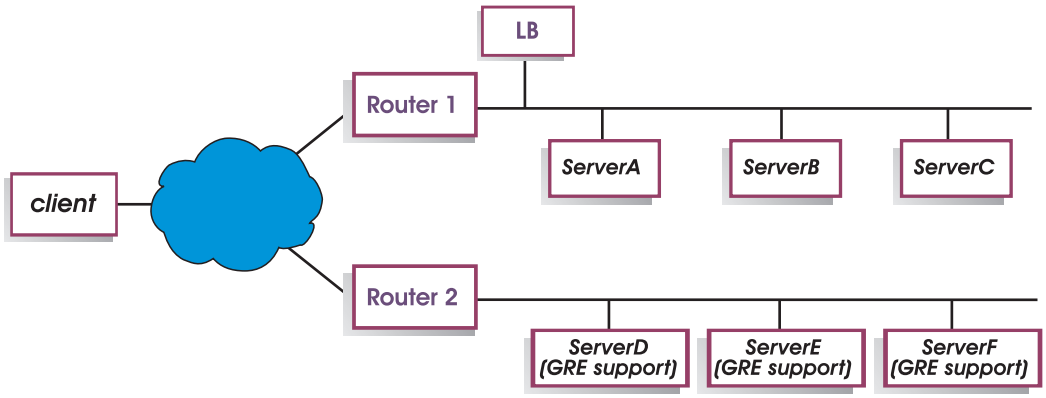


Figure 35. Wide area example configuration with server platform that supports GRE

For this example (Figure 35), to add remote ServerD, which supports GRE, define it within your Load Balancer configuration as if you are defining a WAN server in the cluster:port:server hierarchy:

dscontrol server add *cluster:port:***ServerD router Router1**

For Linux systems, configuring GRE excapsulation for WAN

Linux systems have the native ability to excapsulate GRE which allows Load Balancer to load balance to Linux for S/390 server images, where many server images share a MAC address. This permits the entry-point Load Balancer to load balance directly to Linux WAN servers, without passing through a Load Balancer at the remote site. This also allows the entry-point Load Balancer's advisors to operate directly with each remote server.

On the entry point Load Balancer, configure as described for WAN.

To configure each Linux backend server, issue the following commands as root. (These commands may be added to the system's startup facility so that changes are preserved across reboots.)

```
# modprobe ip_gre
# ip tunnel add gre-nd mode gre ikey 3735928559
# ip link set gre-nd up
# ip addr add cluster address dev gre-nd
```

Note: The Linux server configured using these instructions *must not* be on the same physical segment as the entry-point Load Balancer. This is because the

Linux server will respond to "ARP who-has" requests for the cluster address, causing a race condition leading to a possible "short-circuit" in which all traffic to the cluster address is directed only to the winner of the ARP-race.

Using explicit linking

In general, the load-balancing functions of the Dispatcher work independently of the content of the sites on which the product is used. There is one area, however, where site content can be important, and where decisions made regarding content can have a significant impact upon the Dispatcher's efficiency. This is in the area of link addressing.

If your pages specify links that point to individual servers for your site, you are in effect forcing a client to go to a specific machine, thus bypassing any load balancing function that might otherwise be in effect. For this reason, always use the address of Dispatcher in any links contained in your pages. Note that the kind of addressing used may not always be apparent, if your site uses automated programming that dynamically creates HTML. To maximize your load-balancing, you should be aware of any explicit addressing and avoid it where possible.

Using a private network configuration

You can set up Dispatcher and the TCP server machines using a private network. This configuration can reduce the contention on the public or external network that can affect performance.

For AIX systems, this configuration can also take advantage of the fast speeds of the SP High Performance Switch if you are running Dispatcher and the TCP server machines on nodes in an SP Frame.

To create a private network, each machine must have at least two LAN cards, with one of the cards connected to the private network. You must also configure the second LAN card on a different subnet. The Dispatcher machine will then send the client requests to the TCP server machines through the private network.

Windows systems: Configure the nonforwarding address using the executor configure command.

The servers added using the **dscontrol server add** command must be added using the private network addresses; for example, referring to the Apple server example in Figure 36 on page 191, the command should be coded as:

dscontrol server add *cluster_address*:**80:10.0.0.1**

not

dscontrol server add *cluster_address*:**80:9.67.131.18**

If you are using Site Selector to provide load information to Dispatcher, you must configure Site Selector to report loads on the private addresses.

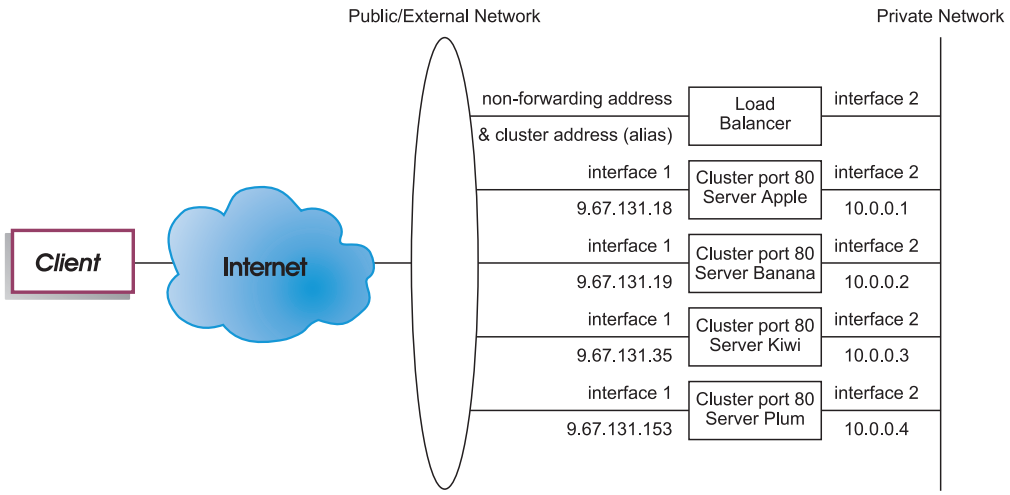


Figure 36. Example of a private network using Dispatcher

Using a private network configuration only applies to the Dispatcher component.

Use wildcard cluster to combine server configurations

Using wildcard cluster to combine server configurations only applies to the Dispatcher component.

The “wildcard” refers to the cluster's ability to match multiple IP addresses (that is, acts as a wildcard). Cluster address 0.0.0.0 is used to specify a wildcard cluster.

If you have many cluster addresses to load-balance, and the port/server configurations are identical for all your clusters, you can combine all the clusters into one wildcard cluster configuration.

You must still explicitly configure each cluster address on one of the network adapters of your Dispatcher workstation. You should not add any of the cluster addresses to the Dispatcher configuration using the dscontrol cluster add command however.

Add only the wildcard cluster (address 0.0.0.0), and configure the ports and servers as required for load balancing. Any traffic to any of the adapter configured addresses is load balanced using the wildcard cluster configuration.

An advantage of this approach is that traffic to all the cluster addresses is taken into account when determining the best server to go to. If one cluster is getting a lot of traffic, and it has created many active connections on one of the servers, traffic to other cluster addresses is load balanced using this information.

You can combine the wildcard cluster with actual clusters if you have some cluster addresses with unique port/server configurations, and some with common configurations. The unique configurations must each be assigned to an actual cluster address. All common configurations can be assigned to the wildcard cluster.

Use wildcard cluster to load balance firewalls

Using wildcard cluster to load balance firewalls only applies to the Dispatcher component. Cluster address 0.0.0.0 is used to specify a wildcard cluster.

The wildcard cluster can be used to load balance traffic to addresses that are not explicitly configured on any network adapter of the Dispatcher workstation. In order for this to work, the Dispatcher must at least be able to see all the traffic it is to load balance. The dispatcher workstation will not see traffic to addresses that have not been explicitly configured on one of its network adapters unless it is set up as the default route for some set of traffic.

After Dispatcher has been configured as a default route, any TCP or UDP traffic through the Dispatcher machine is load balanced using the wildcard cluster configuration.

One application of this is to load balance firewalls. Because firewalls can process packets for any destination address and any destination port, you need to be able to load balance traffic independent of the destination address and port.

Firewalls are used to handle traffic from non-secure clients to secure servers, and the responses from the secure servers, as well as traffic from clients on the secure side to servers on the non-secure side, and the responses.

You must set up two Dispatcher machines, one to load balance non-secure traffic to the non-secure firewall addresses and one to load balance secure traffic to the secure firewall addresses. Because both of these Dispatchers must use the wildcard cluster and wildcard port with different sets of server addresses, the two Dispatchers must be on two separate workstations.

Use wildcard cluster with Caching Proxy for transparent proxy

Using wildcard cluster with Caching Proxy for transparent proxy only applies to the Dispatcher component. Cluster address 0.0.0.0 is used to specify a wildcard cluster.

The wildcard cluster function also allows Dispatcher to be used to enable a transparent proxy function for a Caching Proxy server residing on the same machine as Dispatcher. This is an AIX feature only, as there must be communication from the dispatcher component to the TCP component of the operating system.

To enable this feature, you must start Caching Proxy listening for client requests on port 80. You then configure a wildcard cluster (0.0.0.0). In the wildcard cluster, you configure port 80. In port 80, you configure the NFA of the Dispatcher machine as the only server. Now any client traffic to any address on port 80 is delivered to the Caching Proxy server running on the Dispatcher workstation. The client request will then be proxied as usual, and the response is sent back from Caching Proxy to the client. In this mode, the Dispatcher component is not performing any load balancing.

Use wildcard port to direct unconfigured port traffic

The wildcard port can be used to handle traffic that is not for any explicitly configured port. One use of this is for load balancing firewalls. A second use is to ensure that traffic to an unconfigured port is handled appropriately. By defining a wildcard port with no servers, you will guarantee that any request to a port that has not been configured is discarded rather than delivered back to the operating system. Port number 0 (zero) is used to specify a wildcard port, for example:

```
dscontrol port add cluster:0
```

Wildcard port to handle FTP traffic

When configuring a cluster to handle passive FTP and the wildcard port, passive FTP by default utilizes the entire non-privileged TCP port range for data connections. This means a client, with an existing connection through a load-balancing cluster to an FTP control port, will have subsequent control connections and high port connections (port >1023) to the same cluster automatically routed by Load Balancer to the same server as the FTP control connection.

If the wildcard port and the FTP port on the same cluster do not have the same server set, then high port applications (port >1023) may fail when a client has an existing FTP control connection. Therefore, configuring different server sets for the FTP and wildcard ports on the same cluster is not recommended. If this scenario is desired, the FTP daemon passive port range must be configured in the Load Balancer configuration.

Denial of service attack detection

This feature is only available for the Dispatcher component.

Dispatcher provides the ability to detect potential "denial of service" attacks and notify administrators by an alert. Dispatcher does this by analyzing incoming requests for a conspicuous amount of half-open TCP connections on servers, a common trait of simple denial of service attacks. In a denial of service attack, a site receives a large quantity of fabricated SYN packets from a large number of source IP addresses and source port numbers, but the site receives no subsequent packets for those TCP connections. This results in a large number of half-opened TCP connections on the servers, and over time the servers can become very slow, accepting no new incoming connections.

Note: There must be incoming traffic through the cluster and port that are under attack for Dispatcher to determine the end of a denial of service attack. Dispatcher is unable to detect that the attack stops until traffic begins to flow again.

Load Balancer provides user exits that trigger scripts which you can customize that alert the Administrator to a possible denial of service attack. Dispatcher provides sample script files in the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/samples**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\samples**

The following scripts are available:

- halfOpenAlert — a probable denial of service (DoS) attack has been detected
- halfOpenAlertDone — the DoS attack has finished

In order to run the files, you must move them to the following directory and remove the ".sample" file extension:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/bin**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\bin**

To implement the DoS attack detection, set the **maxhalfopen** parameter on the **dscontrol port** command as follows:

```
dscontrol port set 127.40.56.1:80 maxhalfopen 1000
```

In the above example, Dispatcher will compare the current total number of half-open connections (for all servers residing on cluster 127.40.56.1 on port 80) with the threshold value of 1000 (specified by the maxhalfopen parameter). If the current half-open connections exceeds the threshold, then a call to an alert script (halfOpenAlert) is made. When the number of half-open connections drops below the threshold, a call to another alert script (halfOpenAlertDone) is made to indicate that the attack is over.

To determine how to set the maxhalfopen value: Periodically (perhaps every 10 minutes) run a half-open connection report (**dscontrol port halfopenaddressreport** *cluster:port*) when your site is experiencing normal to heavy traffic. The half-open connection report will return the current "total half-open connections received." You should set maxhalfopen to a value that is anywhere from 50 to 200% greater than the largest number of half-open connnections that your site experiences.

In addition to statistical data reported, the halfopenaddressreport will also generate entries in the log (..ibm/edge/lb/servers/logs/dispatcher/halfOpen.log) for all the client addresses (up to approximately 8000 address pairs) that have accessed servers that resulted in half open connnections.

Note: There is an SNMP trap corresponding to the halfOpenAlert and halfOpenAlertDone scripts. If the SNMP subagent is configured and running, the corresponding traps are sent under the same conditions which trigger the scripts. For more information on the SNMP subagent, see "Using Simple Network Management Protocol with the Dispatcher component" on page 221.

To provide additional protection from denial of service attacks for backend servers, you can configure wildcard clusters and ports. Specifically, under each configured cluster add a wildcard port with no servers. Also add a wildcard cluster with a wildcard port and no servers. This will have the effect of discarding all packets which are not addressed to a non-wildcard cluster and port. For information on wildcard clusters and wildcard ports, see "Use wildcard cluster to combine server configurations" on page 191 and "Use wildcard port to direct unconfigured port traffic" on page 192.

Using binary logging to analyze server statistics

Note: The binary logging feature applies to the Dispatcher and CBR component.

The binary logging feature allows server information to be stored in binary files. These files can then be processed to analyze the server information that has been gathered over time.

The following information is stored in the binary log for each server defined in the configuration.

- cluster address
- port number
- serverID
- server address
- server weight
- server total connections
- server active connections

- server port load
- server system load

Some of this information is retrieved from the executor as part of the manager cycle. Therefore the manager must be running in order for the information to be logged to the binary logs.

Use **dscontrol binlog** command set to configure binary logging.

- binlog start
- binlog stop
- binlog set interval <second>
- binlog set retention <hours>
- binlog status

The start option starts logging server information to binary logs in the logs directory. One log is created at the start of every hour with the date and time as the name of the file.

The stop option stops logging server information to the binary logs. The log service is stopped by default.

The set interval option controls how often information is written to the logs. The manager will send server information to the log server every manager interval. The information is written to the logs only if the specified log interval seconds have elapsed since the last record was written to the log. By default, the log interval is set to 60 seconds. There is some interaction between the settings of the manger interval and the log interval. Since the log server is provided with information no faster than manager interval seconds setting the log interval less than the manager interval effectively sets it to the same as the manager interval. This logging technique allows you to capture server information at any granularity. You can capture all changes to server information that are seen by the manager for calculating server weights. However, this amount of information is probably not required to analyze server usage and trends. Logging server information every 60 seconds gives you snapshots of server information over time. Setting the log interval very low can generate huge amounts of data.

The set retention option controls how long log files are kept. Log files older than the retention hours specified are deleted by the log server. This will only occur if the log server is being called by the manager, so stopping the manager will cause old log files not to be deleted.

The status option returns the current settings of the log service. These settings are whether the service is started, what the interval is, and what the retention hours are.

A sample Java program and command file have been provided in the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/samples/BinaryLog**
- Windows operating systems: **<***install_root***>ibm\edge\lb\samples\BinaryLog**

This sample shows how to retrieve all the information from the log files and print it to the screen. It can be customized to do any type of analysis you want with the data. An example using the supplied script and program for the dispatcher would be:

```
dslogreport 2001/05/01 8:00 2001/05/01 17:00
```

to get a report of the Dispatcher component's server information from 8:00 AM to 5:00 PM on May 1, 2001. (For CBR, use **cbrlogreport**.)

Using a collocated client

Only Linux systems support configurations where the client is located on the same machine as Load Balancer.

Collocated client configurations might not function correctly on other platforms because Load Balancer uses different techniques to examine the incoming packets on the various operating systems that it supports. In most cases, on systems other than Linux, Load Balancer does not receive packets from the local machine. It receives packets coming from the network only. Because of this, requests made to the cluster address from the local machine are not received by Load Balancer and cannot be serviced.

Chapter 22. Advanced features for Cisco CSS Controller and Nortel Alteon Controller

This chapter includes the following sections:

Note: In this chapter **xxxcontrol** denotes **ccocontrol** for Cisco CSS Controller and **nalcontrol** for Nortel Alteon Controller.

Collocation

Cisco CSS Controller or Nortel Alteon Controller can reside on the same machine as a server for which you are load balancing requests. This is commonly referred to as *collocating* a server. No additional configuration steps are required.

Note: A collocated server competes for resources with Load Balancer during times of high traffic. However, in the absence of overloaded machines, using a collocated server offers a reduction in the total number of machines necessary to set up a load-balanced site.

High availability

The high availability feature is now available for Cisco CSS Controller and Nortel Alteon Controller.

To improve controller fault tolerance, the high availability function contains these features:

- Heartbeat mechanism to determine availability of partner controllers. Heartbeats are exchanged between addresses configured on the **xxxcontrol highavailability add** command. You can configure the interval during which beats are exchanged and the interval during which a controller takes over from its partner.
- A list of reach targets that each controller must be able to reach to calculate weights and update the switch. See "Failure detection" on page 199 for more information.
- Logic to elect the active controller based on availability and reach information.
- Configurable takeover strategy used in determining how a controller takes over from its partner.
- Manual takeover mechanism for maintenance on active controllers.
- Reports that display current controller role, state, synchronization, and so forth.

Configuration

See “ccocontrol highavailability — control high availability” on page 375 and “nalcontrol highavailability — control high availability” on page 393 for the complete syntax for **xxxcontrol highavailability**.

To configure controller high availability:

1. Start the controller server on both controller machines.
2. Configure each controller with identical configurations.
3. Configure the local high availability role, address, and partner address as follows:

   ```
   xxxcontrol highavailability add address 10.10.10.10
   partneraddress 10.10.10.20 port 143 role primary
   ```

4. Configure the partner high availability role, address, and partner address as follows:

   ```
   xxxcontrol highavailability add address 10.10.10.20
   partneraddress 10.10.10.10 port 143 role secondary
   ```

   The address and partneraddress parameters are reversed on the primary and secondary machines.
5. Optionally, configure high availability parameters on the local and partner controllers; for example:

   ```
   xxxcontrol highavailability set beatinterval 1000
   ```

6. Optionally, configure reach targets on local and partner controllers as follows:

   ```
   xxxcontrol highavailability usereach 10.20.20.20
   ```

   The same number of reach targets must be configured on the local and partner controllers.
7. Start the high availability component and define recovery strategy on local and partner controllers as follows:

   ```
   xxxcontrol highavailability start auto
   ```

8. Optionally, display high availability information on local and partner controllers as follows:

   ```
   xxxcontrol highavailability report
   ```

9. Optionally, specify takeover on standby controller to take over from active controller as follows:

   ```
   xxxcontrol highavailability takeover
   ```

   This is necessary only for maintenance.

Notes:

1. To configure a single controller without high availability, do not issue any high availability commands.
2. To convert two controllers in a high availability configuration to a single controller, stop high availability on the standby controller first; optionally, then stop high availability on the active controller.
3. When you run two controllers in a high availability configuration, unexpected results can occur if any of the controller properties differ between the switches; for example, switchconsultantid, switch address, and so forth. You can also get unexpected results if the controller high availability properties do not match; for example, port, role, reach targets, beatinterval, takeoverinterval, and recovery strategy.

Failure detection

In addition to the loss of connectivity between active and standby controllers, which is detected through the heartbeat messages, *reachability* is another failure detection mechanism.

When you configure controller high availability, you can provide a list of hosts that each of the controllers must reach to work correctly. There must be at least one host for each subnet that your controller machine uses. These hosts can be routers, IP servers, or other host types.

Host reachability is obtained by the reach advisor, which pings the host. Switchover takes place if the heartbeat messages cannot go through, or if the reachability criteria are better met by the standby controller than by the active controller. To make this decision based on all available information, the active controller regularly sends the standby controller its reachability capabilities and vice versa. The controllers then compare their reachability information with their partner's information and decide who should be active.

Recovery strategy

The roles of the two controller machines are configured as primary and secondary. At startup the controllers exchange information until each machine is synchronized. At this point, the primary controller moves to the active state and begins calculating weights and updating the switch, while the secondary machine moves to standby state and monitors the availability of the primary machine.

At any point if the standby machine detects that the active machine has failed, the standby machine performs a takeover of the active (failed) machine's load-balancing functions and becomes the active machine. When the primary machine is again operational, the two machines determine which controller will be active according to how recovery strategy is configured.

There are two kinds of recovery strategy:

Automatic recovery

The primary controller moves to the active state, calculating and updating weights, as soon as it becomes operational again. The secondary machine moves to standby after the primary is active.

Manual recovery

The active secondary controller remains in active state, even after the primary controller is operational.

The primary controller moves to standby state and requires manual intervention to move to the active state.

The strategy parameter must be set the same for both machines.

Examples

For Cisco CSS Controller high availability configuration examples, see “Examples” on page 377.

For Nortel Alteon Controller high availability configuration examples, see “Examples” on page 395.

Optimizing the load balancing provided by Load Balancer

The controller function of Load Balancer performs load balancing based on the following settings:

- "Importance given to metric information"
- "Weights"
- "Weight calculation sleeptimes" on page 201
- "Advisor sleeptimes" on page 202
- "Sensitivity threshold" on page 201

You can change these settings to optimize load balancing for your network.

Importance given to metric information

The controller can use some or all of the following metric collectors in its weighting decisions:

- *Active connections*: The number of active connections on each load balanced server machine, retrieved from the switch.
- *Connection rate*: The number of new connections since the last query on each load balanced server machine retrieved from the switch.
- *CPU*: The percentage of CPU in use on each load balanced server machine (input from Metric Server agent).
- *Memory*: The percentage of memory in use (input from Metric Server agent) on each load balanced server.
- *System metric*: The input from the system monitoring tools, such as Metric Server or WLM.
- *Application-specific*: The input from advisors listening on the port.

The default metrics are activeconn and connrate.

You can change the relative proportion of importance of the metric values. Think of the proportions as percentages; the sum of the relative proportions must equal 100%. By default, the active connections and new connections metrics are used and their proportions are set to 50/50. In your environment, you might need to try different metric proportion combinations to find the combination that gives the best performance.

To set the proportion values:

For Cisco CSS Controller

ccocontrol ownercontent metrics *metricName1 proportion1 metricName2 proportion2*

For Nortel Alteon Controller

nalcontrol service metrics *metricName1 proportion1 metricName2 proportion2*

Weights

Weights are set based upon application response time and availability, feedback from the advisors, and feedback from a system-monitoring program, such as Metric Server. If you want to set weights manually, specify the fixedweight option for the server. For a description of the fixedweight option, see "Controller fixed weights" on page 201.

Weights are applied to all servers providing a service. For any particular service, the requests are distributed between servers based on their weights relative to each

other. For example, if one server is set to a weight of 10, and the other to 5, the server set to 10 should get twice as many requests as the server set to 5.

If an advisor finds that a server has gone down, the weight for the server is set to -1. For Cisco CSS Controller and Nortel Alteon Controller the switch is informed that the server is not available and the switch stops assigning connections to the server.

Controller fixed weights

Without the controller, advisors cannot run and cannot detect if a server is down. If you choose to run the advisors, but do *not* want the controller to update the weight you have set for a particular server, use the **fixedweight** option on the **ccocontrol service** command for Cisco CSS Controller or the **nalcontrol server** command for Nortel Alteon Controller.

Use the **fixedweight** command to set the weight to the value you desire. The server weight value remains fixed while the controller is running until you issue another command with fixedweight set to no.

Weight calculation sleeptimes

To optimize overall performance, you can restrict how often metrics are collected.

The consultant sleeptime specifies how often the consultant updates the server weights. If the consultant sleeptime is too low, it can mean poor performance as a result of the consultant constantly interrupting the switch. If the consultant sleeptime is too high, it can mean that the switch's load balancing is not based on accurate, up-to-date information.

For example, to set the consultant sleeptime to 1 second:

```
xxxcontrol consultant set consultantID sleeptime interval
```

Sensitivity threshold

Other methods are available for you to optimize load balancing for your servers. To work at top speed, updates to the weights for the servers are only made if the weights have changed significantly. Constantly updating the weights when there is little or no change in the server status would create an unnecessary overhead. When the percentage weight change for the total weight for all servers providing a service is greater than the sensitivity threshold, the weights used by the load balancer to distribute connections are updated. Consider, for example, that the total weight changes from 100 to 105. The change is 5%. With the default sensitivity threshold of 5, the weights used by the load balancer are not updated, because the percentage change is not **above** the threshold. If, however, the total weight changes from 100 to 106, the weights are updated. To set the consultant's sensitivity threshold to a value other than the default, enter the following command:

```
xxxcontrol consultant set consultantID sensitivity percentageChange
```

In most cases, you will not need to change this value.

Advisors

Advisors are agents within Load Balancer. Their purpose is to assess the health and load of server machines. They do this with a proactive client-like exchange with the servers. Consider advisors as lightweight clients of the application servers.

Note: For a detailed list of advisors, see “List of advisors” on page 149.

How advisors work

Advisors periodically open a TCP connection with each server and send a request message to the server. The content of the message is specific to the protocol running on the server. For example, the HTTP advisor sends an HTTP “HEAD” request to the server.

Advisors then listen for a response from the server. After getting the response, the advisor makes an assessment of the server. To calculate this *load* value, most advisors measure the time for the server to respond, then use this value (in milliseconds) as the load.

Advisors then report the load value to the consultant function, where it appears in the consultant report. The consultant then calculates aggregate weight values from all its sources, per its proportions, and sends these weight values to the switch. The switch uses these weights for load balancing new incoming client connections.

If the advisor determines that a server is alive and well, it reports a positive, non-zero load number to the consultant. If the advisor determines that a server is not active, it returns a special load value of negative one (-1) to inform the switch that the server is down. Subsequently, the switch does not forward any further connections to that server until the server has come back up.

Advisor sleeptimes

Note: The advisor defaults work efficiently for the great majority of possible scenarios. Use caution when entering values other than the defaults.

The advisor sleeptime sets how often an advisor asks for status from the servers on the port it is monitoring and then reports the results to the consultant. If the advisor sleeptime is too low, it can result in poor performance because the advisor constantly interrupts the servers. If the advisor sleeptime is too high, it can mean that the consultant's weighting decisions are not based on accurate, up-to-date information.

For example, to set the interval to 3 seconds for the HTTP advisor, type the following command:

xxxcontrol metriccollector set *consultantID*:**HTTP sleeptime 3**

Advisor connect timeout and receive timeout for servers

You can set the amount of time an advisor takes to detect that a particular port on the server or service has failed. The failed-server timeout values, connecttimeout and receivetimeout, determine how long an advisor waits before reporting that either a connect or receive has failed.

To obtain the fastest failed-server detection, set the advisor connect and receive timeouts to the smallest value (one second), and set the advisor and consultant sleeptime to the smallest value (one second).

Note: If your environment experiences a moderate-to-high volume of traffic and server response time increases, do not set the timeoutconnect and timeoutreceive values too small. If these values are too small, the advisor might prematurely mark a busy server as failed.

To set the timeoutconnect to 9 seconds for the HTTP advisor, type the following command:

xxxcontrol metriccollector set *consultantID*:**HTTP timeoutconnect 9**

The default for connect and receive timeout is 3 times the value specified for the advisor sleeptime.

Advisor retry

Advisors have the ability to retry a connection before marking a server down. The advisor will not mark a server down until the server query has failed the number of retries plus 1. If not set the retry value defaults to zero.

For the Cisco CSS Controller, set the **retry** value using **ccocontrol ownercontent set** command. For more information, see "ccocontrol ownercontent — control the owner name and content rule" on page 380.

For the Nortel Alteon Controller, set the **retry** value using **nalcontrol service set** command. For more information, see "nalcontrol service — configure a service" on page 400.

Create custom (customizable) advisors

Note: In this section **server** is used as a generic term to refer to a service for Cisco CSS Controller or to a server for Nortel Alteon Controller.

The custom (customizable) advisor is a small piece of Java code that you provide as a class file, and is called by the base code. The base code provides all administrative services, such as:

- Starting and stopping an instance of the custom advisor
- Providing status and reports
- Recording history information in a log file

It also reports results to the consultant. Periodically the base code performs an advisor cycle, where it individually evaluates all servers in its configuration. It starts by opening a connection with a server machine. If the socket opens, the base code calls the getLoad method (function) in the custom advisor. The custom advisor then performs the necessary steps to evaluate the health of the server. Typically, it sends a user-defined message to the server and then waits for a response. (Access to the open socket is provided to the custom advisor.) The base code then closes the socket with the server and reports the load information to the consultant.

The base code and custom advisor can operate in either normal or replace mode. Choice of the mode of operation is specified in the custom advisor file as a parameter in the constructor method.

In normal mode, the custom advisor exchanges data with the server, and the base advisor code times the exchange and calculates the load value. The base code then reports this load value to the consultant. The custom advisor needs only return a zero (on success) or negative one (on error). To specify normal mode, the replace flag in the constructor is set to false.

In replace mode, the base code does not perform any timing measurements. The custom advisor code performs whatever operations are desired for its unique

requirements, and then returns an actual load number. The base code will accept the number and report it to the consultant. For best results, normalize your load number between 10 and 1000, with 10 representing a fast server, and 1000 representing a slow server. To specify replace mode, the replace flag in the constructor is set to true.

With this feature, you can write your own advisors to provide the precise information about servers that you need. A sample custom advisor, **ADV_ctlrsample.java**, is provided for the controllers. After installing Load Balancer, you can find the sample code in:

- AIX, HP-UX, Linux, and Solaris operating systems: /opt/ibm/edge/lb/servers/samples/CustomAdvisors
- Windows systems: <*install_root*>ibm\edge\lb\servers\samples\CustomAdvisors

Note: If you add a custom advisor to Cisco CSS Controller or Nortel Alteon Controller, you must stop and then restart **ccoserver** or **nalserver** (for Windows systems, use Services) to enable the Java process to read the new custom advisor class files. The custom advisor class files are loaded only at startup.

Naming Convention

Your custom advisor file name must be in the form ADV_*myadvisor*.java. It must start with the prefix ADV_ in uppercase. All subsequent characters must be lowercase letters.

As per Java conventions, the name of the class defined within the file must match the name of the file. If you copy the sample code, be sure to change all instances of ADV_ctrlsample inside the file to your new class name.

Compilation

Custom advisors are written in Java language. Use the Java compiler that is installed with Load Balancer. The following files are referenced during compilation:

- The custom advisor file
- The base classes file, ibmlb.jar, found in the following directory:
  - AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/lib**
  - Windows operating systems: **<*install_root*>ibm\edge\lb\servers\lib**

Your classpath must point to both the custom advisor file and the base classes file during the compile.

For Windows platform, a compile command might look like this:

```
install_dir/java/bin/javac -classpath
    <install_root>ibm\edge\lb\servers\lib\ibmlb.jar ADV_pam.java
```

where:

- Your advisor file is named ADV_pam.java
- Your advisor file is stored in the current directory

The output for the compilation is a class file; for example:

```
ADV_pam.class
```

Before starting the advisor, copy the class file to the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/lib/CustomAdvisors**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\lib\CustomAdvisors**

Note: If you want, custom advisors can be compiled on one operating system and run on another. For example, you can compile your advisor on Windows systems, copy the class file (in binary) to an AIX machine, and run the custom advisor there.

For AIX, HP-UX, Linux , and Solaris systems, the syntax is similar.

Run

To run the custom advisor, you must first copy the class file to the proper installation directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/lib/CustomAdvisors/ADV_pam.class**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\lib\CustomAdvisors\ADV_pam.class**

Start the consultant, then issue this command to start your custom advisor:

For Cisco CSS Controller

ccocontrol ownercontent metrics *consultantID***:***ownerContentID pam* **100**

For Nortel Alteon Controller

nalcontrol service metrics *consultantID***:***serviceID pam* **100**

where:

- pam is the name of your advisor, as in ADV_pam.java
- 100 is the proportion of weight given to this advisor

Required routines

Like all advisors, a custom advisor extends the function of the advisor base, called ADV_Base. It is the advisor base that actually performs most of the advisor's functions, such as reporting loads back to the consultant for use in the consultant's weight algorithm. The advisor base also performs socket connect and close operations and provides send and receive methods for use by the advisor. The advisor itself is used only for sending and receiving data to and from the port on the server being advised. The TCP methods within the advisor base are timed to calculate the load. A flag within the constructor in the ADV_base overwrites the existing load with the new load returned from the advisor if desired.

Note: Based on a value set in the constructor, the advisor base supplies the load to the weight algorithm at specified intervals. If the actual advisor has not completed so that it can return a valid load, the advisor base uses the previous load.

These are base class methods:

- A **constructor** routine. The constructor calls the base class constructor (see the sample advisor file)
- An **ADV_AdvisorInitialize** method. This method provides a hook in case additional steps need to be taken after the base class completes its initialization.

- A **getLoad** routine. The base advisor class performs the open socket; therefore getLoad needs only to issue the appropriate send and receive requests to complete the advise cycle.

Search order

The controllers first look at the provided list of native advisors; if they do not find a given advisor there, they look at the list of custom advisors.

Naming and path

- The custom advisor class must be located within the subdirectory of the CustomAdvisors directory in the Load Balancer base directory. The defaults for this directory vary by operating system:
  - AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/lib/CustomAdvisors**
  - Windows operating systems: **<***install_root***>ibm\edge\lb\servers\lib\CustomAdvisors**
- Only lowercase, alphabetic characters are permitted. This eliminates case sensitivity when an operator types in commands on the command line. The advisor file name must be prefixed with **ADV_**.

Sample advisor

The program listing for a controller sample advisor is included in "Sample advisor" on page 419. After installation, this sample advisor can be found in the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/lib/CustomAdvisors**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\lib\CustomAdvisors**

Metric Server

Metric Server provides server load information to the Load Balancer in the form of system-specific metrics, reporting on the health of the servers. The Load Balancer consultant queries the Metric Server agent residing on each of the servers, assigning weights to the load balancing process using the metrics gathered from the agents. The results are also placed into the service report for Cisco CSS Controller or the server report for Nortel Alteon Controller.

Prerequisites

The Metric Server agent must be installed and running on all servers that are being load balanced.

How to Use Metric Server

Below are the steps to configure Metric Server for the controllers.

- Controller side
  1. Start **ccoserver** or **nalserver**.
  2. For Cisco CSS Controller, add a switch consultant, then add ownercontent.

     For Nortel Alteon Controller, add a switch consultant, then add a service.
  3. Specify the port on which the metric server agent listens. This must match the information specified in the metricserver.cmd file. The default port is 10004. Use the following command:

For Cisco CSS Controller

ccocontrol service set *consultantID*:*ownerContentID*:*serverID* **metricserverport** *portNumber*

For Nortel Alteon Controller

nalcontrol server set *consultantID*:*serviceID*:*serverID* **metricserverport** *portNumber*

4. Issue the system metric command:

   For Cisco CSS Controller

   ccocontrol ownercontent metrics *consultantID:ownerContentID metricName importance*

   For Nortel Alteon Controller

   nalcontrol service metrics *consultantID:serviceID metricName importance*

   where *metricName* is the name of the metric server script.

   The system metric script resides on the backend server and runs on each of the servers in the configuration under the specified ownercontent or service. Two scripts, **cpuload** and **memload** are provided, or you can create custom system metric scripts. The script contains a command that must return a numeric value. This numeric value represents a load measurement, not an availability value.

   Limitation: For Windows systems, if the name of your system metric script has an extension other than .exe, you must specify the full name of the file; for example, mySystemScript.bat. This is a Java code limitation.

5. Issue the command for your controller as follows:

   For Cisco CSS Controller

   ccocontrol consultant start

   For Nortel Alteon Controller

   nalcontrol consultant start

Note: Ensure Security —

- On the controller machine, create key files using the **lbkeys create** command. See "Remote Method Invocation (RMI)" on page 213 for more information on lbkeys.
- On the server machine, copy the resulting key file to the following directory:
  - AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/admin/key**
  - Windows operating systems: **<***install_root***>ibm\edge\lb\admin\key**

  Verify that the key file's permissions enable the file to be readable by the root.

- Metric Server agent (server machine side)
  1. Install the Metric Server package from the Load Balancer install.
  2. Check the **metricserver** script in the appropriate directory to verify that you are using the desired RMI port. The default RMI port is 10004, and the location for the script is:
     - For UNIX and Linux based operating systems, the directory is **/usr/bin**.
     - For Windows systems, the directory is **<***install_root***>ibm\edge\lb\bin**.

Note: The RMI port value specified must be the same value as the RMI port value for the Metric Server on the controller machine.

3. The following two scripts are provided: **cpuload** (returns the percentage of cpu in use ranging from 0-100) and **memload** (returns the percentage of memory in use ranging from 0-100). These scripts reside in the following directory:
   - For UNIX and Linux based operating systems, the directory is **/opt/ibm/edge/lb/ms/script**.
   - For Windows systems, the directory is **<***install_root***>ibm\edge\lb\ms\script**.

   Optionally, you can write your own customized metric script files that define the command that the Metric Server will issue on the server machines. Ensure that any custom scripts are executable and located in the following directory:
   - For UNIX and Linux based operating systems, the directory is **/opt/ibm/edge/lb/ms/script**.
   - For Windows systems, the directory is **<***install_root***>ibm\edge\lb\ms\script**.

   Custom scripts **must** return a numeric load value.

   Note: A custom metric script must be a valid program or script with a .bat or .cmd extension. Specifically, for AIX, HP-UX, Linux, and Solaris operating systems, scripts must begin with the shell declaration; otherwise, they might not properly run.
4. Start the agent by issuing the **metricserver** command.
5. To stop the Metric Server agent, type **metricserver stop**.

To have Metric Server run on an address other than the local host, edit the metricserver file on the load-balanced server machine. After **java** in the metricserver file, insert the following:

```
-Djava.rmi.server.hostname=OTHER_ADDRESS
```

In addition, before the "if" statements in the metricserver file, add this: `hostname OTHER_ADDRESS`.

For Windows systems: Alias the *OTHER_ADDRESS* on the Microsoft stack. To alias an address on the Microsoft stack, see the section on aliasing an address on the Microsoft stack for a metric server.

Workload manager advisor

WLM is code that runs on MVS mainframes. It can be queried to ask about the load on the MVS machine.

When MVS Workload Management has been configured on your OS/390 system, the controllers can accept capacity information from WLM and use it in the load balancing process. Using the WLM advisor, the controllers periodically open connections through the WLM port on each server in the consultant host table and accept the capacity integers returned. Because these integers represent the amount of capacity that is still available and the consultants expects values representing the loads on each machine, the capacity integers are inverted by the advisor and normalized into load values (for example, a large capacity integer but a small load

value both represent a healthier server). There are several important differences between the WLM advisor and other controller advisors:

1. Other advisors open connections to the servers using the same port on which flows normal client traffic. The WLM advisor opens connections to the servers using a port different from normal traffic. The WLM agent on each server machine must be configured to listen on the same port on which the controller WLM Advisor is started. The default WLM port is 10007.
2. It is possible to use both protocol-specific advisors along with the WLM advisor. The protocol-specific advisors will poll the servers on their normal traffic ports, and the WLM advisor will poll the system load using the WLM port.

Using binary logging to analyze server statistics

The binary logging feature allows server information to be stored in binary files. These files can then be processed to analyze the server information that has been gathered over time.

The following information is stored in the binary log for each server defined in the configuration.

- parent (ownercontentID for Cisco CSS Controller; serviceID for Nortel Alteon Controller)
- server ID
- server address
- server port
- server weight
- number of metrics configured for this server
- list of metric values

The consultant must be running to log information in the binary logs.

Use the **xxxcontrol consultant binarylog** command set to configure binary logging.

- binarylog start
- binarylog stop
- binarylog report
- binarylog set interval <seconds>
- binarylog set retention <hours>

The start option starts logging server information to binary logs in the logs directory. One log is created at the start of every hour with the date and time as the name of the file.

The stop option stops logging server information to the binary logs. The log service is stopped by default.

The set interval option controls how often information is written to the logs. The consultant sends server information to the log server every consultant interval. The information is written to the logs only if the specified log interval seconds have elapsed since the last record was written to the log. By default, the log interval is set to 60 seconds.

There is some interaction between the settings of the consultant interval and the log interval. Because the log server is provided with information no faster than the consultant interval seconds, setting the log interval less than the consultant interval effectively sets it to the same as the consultant interval.

This logging technique allows you to capture server information at any granularity. You can capture all changes to server information that are seen by the consultant for calculating server weights; however, this amount of information is probably not required to analyze server usage and trends. Logging server information every 60 seconds gives you snapshots of server information over time. Setting the log interval very low can generate huge amounts of data.

The set retention option controls how long log files are kept. Log files older than the retention hours specified are deleted by the log server. This occurs only if the log server is being called by the consultant, so if you stop the consultant, old log files are not deleted.

A sample Java program and command file are provided in the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/samples/BinaryLog**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\samples\BinaryLog**

This sample shows how to retrieve all the information from the log files and print it to the screen. It can be customized to do any type of analysis you want with the data.

Following is an example using the supplied script and program:

```
xxxlogreport 2002/05/01 8:00 2002/05/01 17:00
```

This produces a report of the controller's server information from 8:00 AM to 5:00 PM on May 1, 2002.

Using scripts to generate an alert or record server failure

Load Balancer provides user exits that trigger scripts that you can customize. You can create the scripts to perform automated actions, such as alerting an Administrator when servers are marked down or simply record the event of the failure. Sample scripts, which you can customize, are in the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/samples**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\samples**

To run the files, copy them to the following directory, and rename each file according to the directions contained in the script:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/bin**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\bin**

The following sample scripts are provided, where **xxx** is **cco** for Cisco CSS Controller, and **nal** for Nortel Alteon Controller:

- **xxxserverdown** — a server is marked down by the controller.
- **xxxserverUp** — a server is marked back up by the controller.
- **xxxallserversdown** — all servers are marked down for a particular service.

Part 8. Administering and troubleshooting Load Balancer

This part provides information on administering and troubleshooting Load Balancer. It contains the following chapters:

Chapter 23. Operating and managing Load Balancer

Note: When reading this chapter, in the general sections that are not specific to one component, if you are *not* using the Dispatcher component, then substitute "dscontrol" and "dsserver" with the following:

- For CBR, use **cbrcontrol** and **cbrserver**
- For Site Selector, use **sscontrol** and **ssserver**
- For Cisco CSS Controller, use **ccocontrol** and **ccoserver**
- For Nortel Alteon Controller, use **nalcontrol** and **nalserver**

This chapter explains how to operate and manage Load Balancer and includes the following sections:

Remote administration of Load Balancer

Load Balancer provides two different ways to run its configuration programs on a separate machine from the one on which the Load Balancer resides. Communication between the configuration programs (dscontrol, cbrcontrol, sscontrol, ccocontrol, nalcontrol) and the server (dsserver, cbrserver, and so on) can be performed by using either one of the following methods:

- Java Remote Method Invocation (RMI)
- Web-based administration

The advantage to remote administration using RMI is that performance is faster than Web-based administration.

The advantages to using Web-based administration is that it provides secure, authenticated, remote administration, and it can communicate to the Load Balancer machine even when a firewall is present. Also, this administration method does *not* require installation and use of authentication keys (lbkeys) on the remote client machine that is communicating with the Load Balancer machine.

Remote Method Invocation (RMI)

For RMI, the command to connect to a Load Balancer machine for remote administration is **dscontrol host:***remote_host*.

If the RMI call comes from a machine other than the local machine, a public key/private key authentication sequence must occur before the configuration command is accepted.

Communication between the control programs running on the same machine as the component servers are not authenticated.

Use the following command to generate public and private keys to be used for remote authentication:

lbkeys [create | delete]

This command runs only on the same machine as the Load Balancer.

Using the **create** option creates a private key in the servers key directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/key**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\key**

The script also creates public keys in the administration keys directory for each of the Load Balancer components:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/admin/keys**
- Windows operating systems: **<***install_root***>ibm\edge\lb\admin\keys**

The file name for the public key is: *component-ServerAddress-RMIport*. These public keys must then be transported to the remote clients and placed in the administration keys directory.

For a Load Balancer machine with hostname address 10.0.0.25 using the default RMI port for each component, the **lbkeys create** command generates the following files:

- The private key:
  - AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/key/authorization.key**
  - Windows operating systems: **<***install_root***>ibm\edge\lb\servers\key\authorization.key**
- The public keys:
  - AIX, HP-UX, Linux, and Solaris operating systems:
    - /opt/ibm/edge/lb/admin/keys/**dispatcher-10.0.0.25-10099.key**
    - /opt/ibm/edge/lb/admin/keys/**cbr-10.0.0.25-11099.key**
    - /opt/ibm/edge/lb/admin/keys/**ss-10.0.0.25-12099.key**
    - /opt/ibm/edge/lb/admin/keys/**cco-10.0.0.25-13099.key**
    - /opt/ibm/edge/lb/admin/keys/**nal-10.0.0.25-14099.key**
  - Windows operating systems:
    - <*install_root*>ibm\edge\lb\admin\keys**dispatcher-10.0.0.25-10099.key**
    - <*install_root*>ibm\edge\lb\admin\keys**cbr-10.0.0.25-11099.key**
    - <*install_root*>ibm\edge\lb\admin\keys**ss-10.0.0.25-12099.key**
    - <*install_root*>ibm\edge\lb\admin\keys**cco-10.0.0.25-13099.key**
    - <*install_root*>ibm\edge\lb\admin\keys**nal-10.0.0.25-14099.key**

The administration fileset has been installed on another machine. The public key files must be placed in the following directory on the remote client machine:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/admin/keys**

- Windows operating systems: **<***install_root***>ibm\edge\lb\admin\keys**

The remote client will now be authorized to configure Load Balancer on 10.0.0.25.

These same keys must be used on all remote clients that you want to authorize to configure Load Balancer on 10.0.0.25.

If you were to run the **lbkeys create** command again, a new set of public/private keys would be generated. This would mean that all remote clients who tried to connect using the previous keys would not be authorized. The new key would have to be placed in the correct directory on those clients you want to reauthorize.

The **lbkeys delete** command deletes the private and public keys on the server machine. If these keys are deleted, no remote clients will be authorized to connect to the servers.

For both lbkeys create and lbkeys delete there is a **force** option. The force option suppresses the command prompts that ask if you wish to overwrite or delete the existing keys.

After you establish the RMI connection, you can communicate between the configuration programs using dscontrol, cbrcontrol, sscontrol, ccocontrol, nalcontrol, dswizard, cbrwizard, and sswizard commands from a command prompt. You can also configure Load Balancer using the GUI by typing lbadmin from a command prompt.

Note: Due to changes to security packages in the Java version, Load Balancer keys generated for releases prior to v5.1.1 may not be compatible with the keys for the current release, so you must regenerate your keys when you install a new release.

Web-based administration

Requirements

To use Web-based administration, the following is required on the **client machine** that performs remote administration:

- JRE 1.3.0 (or higher)
- For information on supported browsers, refer to the following Web page: `http://www.ibm.com/support/docview.wss?rs=180&uid=swg27006921`

  Note: If you are using Netscape, do not resize (Minimize, Maximize, Restore Down, and so on) the Netscape browser window in which the Load Balancer GUI appears. Because Netscape reloads a page every time browser windows are resized, this will cause a disconnect from the host to occur. You will need to reconnect to the host each time you resize the window.

The following is required on the **host machine** that you are accessing in order to perform remote Web-based administration:

- Caching Proxy V6
- Perl 5.5 (or higher)

Configuring Caching Proxy

- For Caching Proxy, IBM Key Management utility (iKeyman) or other utility is required to create SSL server certificates. (See *Caching Proxy Administration Guide* for information on how to create the certificates.)
- In the "Load Balancer Web-based Administration" section of the Caching Proxy configuration file (ibmproxy.conf), add the following directives after protection domains are defined, but before the mapping rules:

  For Windows systems —

```
Protect /lb-admin/lbwebaccess PROT-ADMIN
Exec  /lb-admin/lbwebaccess C:\PROGRA~1\IBM\edge\lb\admin\lbwebaccess.pl
Pass  /lb-admin/help/*      C:\PROGRA~1\IBM\edge\lb\admin\help\*
Pass  /lb-admin/*.jar       C:\PROGRA~1\IBM\edge\lb\admin\lib\*.jar
Pass  /lb-admin/*           C:\PROGRA~1\IBM\edge\lb\admin\*
Pass  /documentation/lang/*  C:\PROGRA~1\IBM\edge\lb\documentation\lang\*
```

  where ***lang*** is your language subdirectory (for example, en_US)

  For AIX, HP-UX, Linux, and Solaris operating systems —

```
Protect /lb-admin/lbwebaccess PROT-ADMIN
Exec  /lb-admin/lbwebaccess  /opt/ibm/edge/lb/admin/lbwebaccess.pl
Pass  /lb-admin/help/*       /opt/ibm/edge/lb/admin/help/*
Pass  /lb-admin/*.jar        /opt/ibm/edge/lb/admin/lib/*.jar
Pass  /lb-admin/*            /opt/ibm/edge/lb/admin/*
Pass  /documentation/lang/*   /opt/ibm/edge/lb/documentation/lang/*
```

  Note: On HP-UX systems, the lbwebaccess.pl script assumes the Perl binary is located in the /usr/bin/ directory. (The first line of the script contains `#!/usr/bin/perl`.) Update this directory path to wherever the Perl application is located. Another option is to create a symbolic link. For example, if Perl is installed at /opt/perl/bin/perl, run the command:

```
ln -s /opt/perl/bin/perl /usr/bin/perl
```

Running and accessing Web-based administration

In order to run Web-based administration, it must be started on the Load Balancer host machine: Issue **lbwebaccess** from the command prompt of the host machine.

The userID and password to the host machine that you are accessing remotely is also required. The userID and password are the same as the Caching Proxy administration userID and password.

To bring up Load Balancer's Web-based administration, access the following URL on the Web browser from the remote location:

`http://host_name/lb-admin/lbadmin.html`

Where *host_name* is the name of the machine you are accessing in order to communicate with Load Balancer.

When the Web page is loaded, the Load Balancer GUI will appear in the browser window for you to perform remote Web-based administration.

From the Load Balancer GUI, you can also issue configuration control commands. In order to issue a command from the GUI:

1. highlight the Host node from the GUI tree
2. select **Send command...** from the Host pop-up menu

3. in the command entry field, type the command that you want to run. For example: **executor report**. The results and history of the commands run in the current session appear in the window provided.

Refreshing configuration remotely

With remote Web-based administration, if there are multiple administrators updating the Load Balancer configuration from other locations, you will need to refresh the configuration in order to view (for example) the cluster, port or server that has been added (or deleted) by another administrator. Remote Web-based administration GUI provides a **Refresh Configuration** and **Refresh all Configurations** function.

From the Web-based GUI, to refresh the configuration

- for one Host: right-click a **Host** node in the GUI tree structure and select **Refresh Configuration**
- for all Hosts: select **File** from the menu and then select **Refresh All Configurations**

Using Load Balancer logs

For Dispatcher, CBR, and Site Selector

Load Balancer posts entries to a server log, a manager log, a metric monitor log (logging communications with Metric Server agents), and a log for each advisor you use.

Note: Additionally, for the Dispatcher component only, entries can be made to a subagent (SNMP) log.

Note: The Content Based Routing (CBR) component is not available on platforms that run a 64-bit JVM, except for HP-UX ia64. On HP-UX ia64, the CBR component runs as a 32-bit application. You can use the CBR forwarding method of Load Balancer's Dispatcher component to provide content-based routing without the use of Caching Proxy. See “Dispatcher's content-based routing (cbr forwarding method)” on page 41 for more information.

You can set the logging level to define the expansiveness of the messages written to the log. At level 0, errors are logged and Load Balancer also logs headers and records of events that happen only once (for example, a message about an advisor starting to be written to the manager log). Level 1 includes ongoing information, and so on, with level 5 including every message produced to aid in debugging a problem if necessary. The default for the manager, advisor, server, or subagent logs is 1.

You can also set the maximum size of a log. When you set a maximum size for the log file, the file will wrap; when the file reaches the specified size, the subsequent entries are written at the top of the file, overwriting the previous log entries. You cannot set the log size to a value that is smaller than the current one. Log entries are timestamped so you can tell the order in which they were written.

The higher you set the log level, the more carefully you should choose the log size. At level 0, it is probably safe to leave the log size to the default of 1MB; however, when logging at level 3 and above, you should limit the size without making it too small to be useful.

- To configure the logging level or maximum log size for a server log, use the **dscontrol set** command. (To display the server log settings, use the **dscontrol logstatus** command.)
- To configure the logging level or maximum log size for a manager log, use the **dscontrol manager** command.
- To configure the logging level or maximum log size for the metric monitor log that logs communication with Metric Server agents, use the **dscontrol manager metric set** command.
- To configure the logging level or maximum log size for an advisor log, use the **dscontrol advisor** command.
- To configure the logging level or maximum log size for a subagent log, use the **dscontrol subagent** command. (Only the Dispatcher component uses the SNMP subagent.)

Changing the log file paths

By default, the logs generated by Load Balancer are stored in the logs directory of the Load Balancer installation. To change this path, set the *lb_logdir* variable in the dsserver script.

AIX, HP-UX, Linux , and Solaris systems: The dsserver script is found in /usr/bin directory. In this script, the variable *lb_logdir* is set to the default directory. You can modify this variable to specify your log directory. Example:

LB_LOGDIR=/path/to/my/logs/

Windows systems: The dsserver file is found in the <*install_root*>ibm\edge\lb\bin directory. In the dsserver file, the variable *lb_logdir* is set to the default directory. You can modify this variable to specify your log directory. Example:

set LB_LOGDIR=c:\path\to\my\logs

For all operating systems, make sure that there are no spaces on either side of the equal sign and that the path ends in a slash ("/" or "\" as appropriate).

Binary logging

Note: Binary logging does not apply to the Site Selector component.

The binary logging feature of Load Balancer uses the same log directory as the other log files. See "Using binary logging to analyze server statistics" on page 194.

For Cisco CSS Controller and Nortel Alteon Controller

You can set the logging level to define the expansiveness of the messages written to the log. At level 0, errors are logged and Load Balancer also logs headers and records of events that happen only once (for example, a message about an advisor starting to be written to the consultant log). Level 1 includes ongoing information, and so on, with level 5 including every message produced to aid in debugging a problem if necessary. The default for the logs is 1.

You can also set the maximum size of a log. When you set a maximum size for the log file, the file will wrap; when the file reaches the specified size, the subsequent entries will be written at the top of the file, overwriting the previous log entries. You cannot set the log size to a value that is smaller than the current one. Log entries are timestamped so you can tell the order in which they were written.

The higher you set the log level, the more carefully you should choose the log size. At level 0, it is probably safe to leave the log size to the default of 1MB; however, when logging at level 3 and above, you should limit the size without making it too small to be useful.

Controller logs

Cisco CSS Controller and Nortel Alteon Controller have logs as follows:

- controller log (**controller set** command)
- consultant log (**consultant set** command)
- highavailability log (**highavailability set** command)
- metriccollector log (**metriccollector set** command)
- binary log (**consultant binarylog** command)

The following is an example of configuring the logging level and maximum log size for the metric monitor log that logs communication with Metric Server agents:

```
xxxcontrol metriccollector set consultantID:serviceID:metricName
   loglevel x logsize y
```

Changing the log file paths

By default, the logs generated by the controllers are stored in the logs directory of the controller installation. To change this path, set the *xxx_logdir* variable in the xxxserver script.

AIX, HP-UX, Linux , and Solaris systems: The xxxserver script is found in /usr/bin directory. In this script, the variable *xxx_logdir* is set to the default directory. You can modify this variable to specify your log directory. Example:

xxx_LOGDIR=/path/to/my/logs/

Windows systems: The xxxserver file is found in the <*install_root*>ibm\edge\lb\bin directory. In the xxxserver file, the variable *xxx_logdir* is set to the default directory. You can modify this variable to specify your log directory. Example:

set xxx_LOGDIR=c:\path\to\my\logs

For all operating systems, make sure that there are no spaces on either side of the equal sign and that the path ends in a slash ("/" or "\" as appropriate).

Binary logging

The binary logging feature of Load Balancer uses the same log directory as the other log files. See "Using binary logging to analyze server statistics" on page 194.

Using the Dispatcher component

This section explains how to operate and manage the Dispatcher component.

Starting and Stopping Dispatcher

- Type **dsserver** on a command line to start Dispatcher.
- Type **dsserver stop** on a command line to stop Dispatcher.

Using stale timeout value

For Load Balancer, connections are considered stale when there has been no activity on that connection for the number of seconds specified in stale timeout. When the number of seconds has been exceeded with no activity, Load Balancer will remove that connection record from its tables, and subsequent traffic for that connection is discarded.

At the port level, for example, you can specify the stale timeout value on the **dscontrol port set staletimeout** command.

Stale timeout can be set at the executor, cluster, and port levels. At the executor and cluster levels, the default is 300 seconds and it filters down to the port. At the port level, the default depends on the port. Some well defined ports have different default stale timeout values. For example, the telnet port 23 has a default of 259,200 seconds.

Some services may also have staletimeout values of their own. For example, LDAP (Lightweight Directory Access Protocol) has a configuration parameter called idletimeout. When idletimeout seconds have been exceeded, an idle client connection will be forcibly closed. Idletimeout may also be set to 0, which means that the connection will never be forcibly closed.

Connectivity problems can occur when Load Balancer's stale timeout value is smaller than the service's timeout value. In the case of LDAP, the Load Balancer staletimeout value defaults to 300 seconds. If there is no activity on the connection for 300 seconds, Load Balancer will remove the connection record from its tables. If the idletimeout value is larger than 300 seconds (or set to 0), the client may still believe that it has a connection to the server. When the client sends packets, the packets will be discarded by Load Balancer. This causes LDAP to hang when a request is made to the server. To avoid this problem, set the LDAP idletimeout to a nonzero value that is the same or smaller than the Load Balancer staletimeout value.

Using fintimeout and staletimeout to control cleanup of connection records

A client sends a FIN packet after it has sent all its packets so that the server will know that the transaction is finished. When Dispatcher receives the FIN packet, it marks the transaction from active state to FIN state. When a transaction is marked FIN, the memory reserved for the connection can be cleared.

To improve the performance of connection record allocation and reuse, use the **executor set fintimeout** command to control the period during which Dispatcher should keep connections in the FIN state, active in the Dispatcher tables and accepting traffic. When a connection in the FIN state exceeds **fintimeout**, it is removed from the Dispatcher tables and ready for reuse. You can change the FIN timeout using the **dscontrol executor set fincount** command.

Use the **dscontrol executor set staletimeout** command to control the period during which Dispatcher should keep connections in the Established state when no traffic has been seen active in the Dispatcher tables and accepting traffic. See “Using stale timeout value” for more information.

Reporting GUI — the Monitor menu option

Various charts can be displayed based on information from the executor and relayed to the manager. (The GUI Monitor menu option requires that the manager function is running):

- Connections per second per server (multiple servers could be shown on the same graph)
- Relative weighting values per server on a particular port
- Average connection duration per server on a particular port

Using Simple Network Management Protocol with the Dispatcher component

A network management system is a program that runs continuously and is used to monitor, reflect status of, and control a network. Simple Network Management Protocol (SNMP), a popular protocol for communicating with devices in a network, is the current network management standard. The network devices typically have an SNMP ***agent*** and one or more subagents. The SNMP agent talks to the ***network management station*** or responds to command line SNMP requests. The SNMP ***subagent*** retrieves and updates data and gives that data to the SNMP agent to communicate back to the requester.

Dispatcher provides an SNMP ***Management Information Base*** (ibmNetDispatcherMIB) and an SNMP subagent. This allows you to use any network management system, such as — Tivoli® NetView®, Tivoli Distributed Monitoring, or HP OpenView — to monitor the Dispatcher's health, throughput, and activity. The MIB data describes the Dispatcher being managed and reflects current Dispatcher status. The MIB gets installed in the **..lb/admin/MIB** subdirectory.

Note: The MIB, ibmNetDispatcherMIB.02, will not load using Tivoli NetView xnmloadmib2 program. To fix this problem, comment out the NOTIFICATION-GROUP section of the MIB. That is, insert "- -" in front of the line "indMibNotifications Group NOTIFICATION-GROUP", and the 6 lines which follow.

The network management system uses SNMP GET commands to look at MIB values on other machines. It then can notify you if specified threshold values are exceeded. You can then affect Dispatcher performance, by modifying configuration data for Dispatcher, to proactively tune or fix Dispatcher problems before they become Dispatcher or Web server outages.

SNMP commands and protocol

The system usually provides an SNMP agent for each network management station. The user sends a GET command to the SNMP agent. In turn, this SNMP agent sends a GET command to retrieve the specified MIB variable values from a subagent responsible for those MIB variables.

Dispatcher provides a subagent that updates and retrieves MIB data. The subagent responds with the appropriate MIB data when the SNMP agent sends a GET command. The SNMP agent communicates the data to the network management station. The network management station can notify you if specified threshold values are exceeded.

The Dispatcher SNMP support includes an SNMP subagent that uses Distributed Program Interface (DPI) capability. DPI is an interface between an SNMP agent and

its subagents. Windows operating system uses the Windows extension agent as an interface between an SNMP agent and its subagents.

Enabling SNMP on AIX, HP-UX, Linux, and Solaris systems

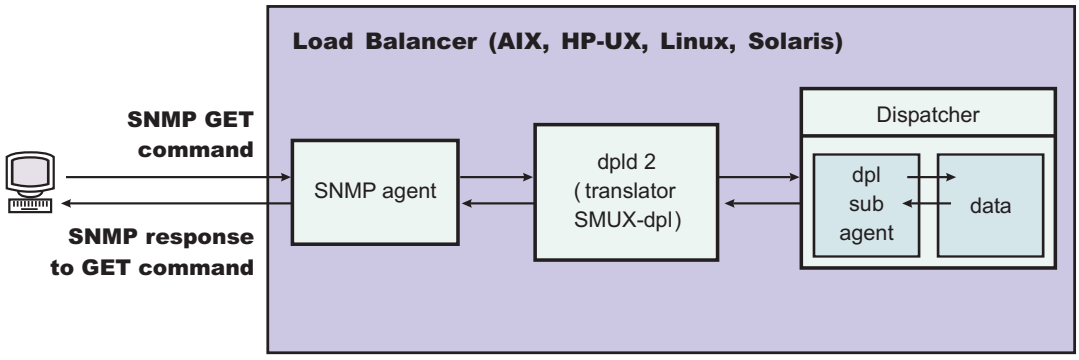


Figure 37. SNMP commands for AIX, HP-UX, Linux, and Solaris operating systems

AIX systems provides an SNMP agent that uses SNMP Multiplexer protocol (SMUX) and provides DPID2, which is an additional executable that works as a translator between DPI and SMUX.

For HP-UX systems, you must obtain an SNMP agent that is SMUX-enabled because HP-UX does not provide one. Load Balancer provides DPID2 for HP-UX systems.

Linux systems provides an SNMP agent that uses SMUX. Most of the Linux versions (for example, Red Hat) come with a UCD SNMP package. UCD SNMP version 4.1 or later has SMUX enabled agents. Load Balancer provides DPID2 for Linux systems.

Note: For SuSE Linux systems, you must obtain an SNMP agent that is SMUX-enabled because SuSE does not provide one.

For Solaris systems, you must obtain an SNMP agent that is SMUX-enabled because Solaris does not provide one. Load Balancer provides DPID2 for Solaris systems in the /opt/ibm/edge/lb/servers/samples/SNMP directory.

The DPI agent must run as a root user. Before you run the DPID2 daemon, update the /etc/snmpd.peers file and the /etc/snmpd.conf file as follows:

For AIX and Solaris systems:

- In the /etc/snmpd.peers file, add the following entry for dpid:

```
"dpid2"    1.3.6.1.4.1.2.3.1.2.2.1.1.2   "dpid_password"
```

- In the /etc/snmpd.conf, add the following entry for dpid:

```
smux     1.3.6.1.4.1.2.3.1.2.2.1.1.2    dpid_password      #dpid
```

For Linux systems:

- In the /etc/snmpd.peers file (if it does not exist in the system then create one), add the following entry for dpid:

```
"dpid2"    1.3.6.1.4.1.2.3.1.2.2.1.1.2   "dpid_password"
```

- In the /etc/snmp/snmpd.conf, add the following entry for dpid:

```
smuxpeer     .1.3.6.1.4.1.2.3.1.2.2.1.1.2    dpid_password
```

  Also, you must comment all lines in the snmpd.conf file that begin with the following words: com2sec, group, view or access.

Enable SNMP on HP-UX systems

To install HP-UX SNMP support:

1. If you do not have a version of GNU SED installed, obtain it from the HP Web site, http://www.hp.com.
2. Obtain ucd-snmp-4.2.4.tar.gz from the following Web page, http://sourceforge.net/project/showfiles.php?group_id=12694.
3. Ensure that you have "gcc" and "gmake or make" installed on your machine. If not, you must install them.
4. Unzip the ucd-snmp-4.2.4.tar.gz file and then untar all of the source files in the directory.
5. Go to the directory where the source files are kept and then do the following:
   a. run `./configure --with-mib-modules=smux`
   b. make
   c. Run the next two commands as root:
      1) `umask 022`
      2) `make install`
   d. `export SNMPCONFPATH=/etc/snmp`
   e. `start /usr/local/sbin/snmpd -s` (This starts the SNMP agent)
   f. `start dpid2` (This starts the DPI translator)
   g. `dscontrol subagent start` (This starts the Dispatcher subagent)

Enable SNMP on SuSE Linux systems

In order to use Load Balancer SNMP with SuSE Linux sysems, you must do the following:

1. Remove the installed ucd-snmp rpm from the SuSE machine.
2. Get ucd-snmp-4.2.4.tar.gz from **http://sourceforge.net/project/showfiles.php?group_id=12694**.
3. Make sure you have "gcc" and "gmake or make" installed on your SuSE machine (you must install them if they are not there).
4. Unzip the ucd-snmp-4.2.4.tar.gz file and then untar all of the source files in the directory.
5. Go to the directory where the source files are kept and then do the following:
   a. `run ./configure ––with–mib–modules=smux`
   b. `make`
   c. Run the next two commands as root:
      1) `umask 022 #`
      2) `make install`
   d. `export SNMPCONFPATH=/etc/snmp`
   e. `start /usr/local/sbin/snmpd –s`
   f. `start dpid2`

Refresh snmpd (if it is already running) so that it will reread the snmpd.conf file:

```
refresh -s snmpd
```

Start the DPID SMUX peer:

```
dpid2
```

The daemons must be started in the following order:

1. SNMP agent
2. DPI translator
3. Dispatcher subagent

Enabling SNMP on Solaris systems

To install Solaris SNMP support:

1. Kill the running Solaris SNMP daemon (snmpdx and snmpXdmid).
2. Rename files as follows:

   /etc/rc3.d/S76snmpdx to **/etc/rc3.d/K76snmpdx**

   /etc/rc3.d/S77dmi to **/etc/rc3.d/K77dmi**
3. Download the following packages from http://www.sunfreeware.com/:
   - libgcc-3.0.3-sol8-sparc-local (SMClibgcc)
   - openssl-0.9.6c-sol8-sparc-local (SMCosslc)
   - popt-1.6.3-sol8-sparc-local (SMCpopt)
4. Install the downloaded packages using pkgadd.
5. Download ucd-snmp-4.2.3-solaris8.tar.gz from http://sourceforge.net/project/showfiles.php?group_id=12694
6. Gunzip and untar the ucd-snmp-4.2.3-solaris8.tar.gz at root directory (/)
7. Issue the following commands:

   export LD_LIBRARY_PATH=$LD_LIBRARY_PATH: /usr/local/lib:/usr/local/ssl/lib:/usr/lib

   export PATH=/usr/local/sbin:/usr/local/bin:$PATH

   export SNMPCONFPATH =/etc/snmp

   export MIBDIRS=/usr/local/share/snmp/mibs

   cp /opt/ibm/edge/lb/servers/samples/SNMP/dpid2 /usr/local/sbin/dpid2
8. It it does not already exist, create /etc/snmpd.peers. Insert the following into snmpd.peers:

   ```
   "dpid2" 1.3.6.1.4.1.2.3.1.2.2.1.1.2     "dpid_password"
   ```
9. If it does not already exist, create /etc/snmp/snmpd.conf. Insert the following into snmpd.conf:

   ```
   smuxpeer        1.3.6.1.4.1.2.3.1.2.2.1.1.2     dpid_password
   ```
10. Start /usr/local/sbin/snmpd.
11. Start /usr/local/sbin/dpid2.

Notes:

1. The following packages are in package format.
   - libgcc-3.0.3-sol8-sparc-local (SMClibgcc)
   - openssl-0.9.6c-sol8-sparc-local (SMCosslc)
   - popt-1.6.3-sol8-sparc-local (SMCpopt)

   On the http://sunfreeware.com/ Web site, the names have an extension of .gz, so do not try to gunzip/untar them. Instead, use pkgadd *packageName*.

2. When you are adding the smuxpeer entry in /etc/snmp/snmpd.conf, make sure no space is added to the **dpid_password** string.
3. The Load Balancer SNMP feature is tested with smux-enabled ucd-snmp version 4.2.3. Future releases of ucd-snmp with smux should work with similar setup.

Enabling SNMP on Windows operating system

To install the Windows SNMP support:

1. Click **Start** > **Control Panel** > **Add/Remove Programs**.
2. Click **Add/Remove Windows Components**.
3. In the Windows Component Wizard, click **Management and Monitoring Tools** (but do not select or clear its check box), then click **Details**
4. Select the **Simple Network Management Protocol** checkbox, click OK.
5. Click Next.

Providing a community name for SNMP

With the executor running, use the **dscontrol subagent start** [*communityname*] command to define the community name used between the Windows OS Extension agent and the SNMP agent.

IMPORTANT: On Windows 2003, by default SNMP does not respond to any community names presented. In such case, the SNMP subagent will not respond to any SNMP requests. To ensure that the SNMP subagent will respond to the community name, you must set SNMP Service Properties with the appropriate community name and destination host(s). Configure SNMP security properties as follows:

1. Open Computer Management
2. In the console tree, click **Services**
3. In the details pane, click **SNMP Service**
4. On the action menu, click **Properties**
5. On the Security tab, under Accepted community names, click **Add**
6. Under Community Rights, select a permission level for this host to process SNMP requests from the selected community (at least **Read Only** permission)
7. In Community Name, type a case-sensitive community name, the same as you provided to the Load Balancer Subagent (default community name: public), and then click **Add**
8. Specify whether or not to accept SNMP packets from a host. Choose one of the following options:
   - To accept SNMP requests from any host on the network, regardless of identity: click **Accept SNMP packets from any host**. (With this option, a person or entity must be verified by means of authentication, based on criteria such as a password or a certificate.)
   - To limit acceptance of SNMP packets: click **To limit acceptance of SNMP packets**, click **Accept SNMP packets from these hosts**, then click **Add**. Type the appropriate host name, IP or IPX address, and then click **Add**, after each entry.
9. Restart the SNMP Service in order for the change to take effect

Traps

SNMP communicates by sending and receiving *traps*, messages sent by managed devices to report exception conditions or the occurrence of significant events, such as a threshold having been reached.

The subagent uses the following traps:

- indHighAvailStatus
- indSrvrGoneDown
- indDOSAttack
- indDOSAttackDone

The **indHighAvailStatus** trap announces that the value of the high-availability status state variable (hasState) has changed. The possible values of hasState are:

-idle This machine is load balancing and is not trying to establish contact with its partner Dispatcher.

-listen High availability has just started and the Dispatcher is listening for its partner.

-active This machine is load balancing.

-standby
This machine is monitoring the active machine.

-preempt
This machine is in a transitory state during the switch from primary to backup.

-elect The Dispatcher is negotiating with its partner regarding who will be the primary or backup.

-no_exec
The executor is not running

The **indSrvrGoneDown** trap announces that the weight for the server specified by the csID (cluster ID), psNum (port number), and ssID (server ID) portion of the Object Identifier has gone to zero. The last known number of active connections for the server is sent in the trap. This trap indicates that, as far as the Dispatcher can determine, the specified server has gone down.

The **indDOSAttack** trap indicates that numhalfopen, the number of half-open connections consisting only of SYN packets, has exceeded the maxhhalfopen threshold for the port specified by the csID (cluster ID) and psNum (port number) portion of the Object Identifier. The number of servers configured on the port is sent in the trap. This trap indicates that Load Balancer may be experiencing a Denial Of Service Attack.

The **indDOSAttackDone** trap indicates that numhalfopen, the number of half-open connections consisting only of SYN packets, has fallen below the maxhalfopen threshold for the port specified by the csID and psNum portion of the Object Identifier. The number of servers configured on the port is sent in the trap. When Load Balancer determines that the possible Denial of Service attack is over, this trap is sent after an indDOSAttack trap is sent.

For AIX, HP-UX, Linux, and Solaris operating systems, due to a limitation in the SMUX API, the enterprise identifier reported in traps from the ibmNetDispatcher subagent may be the enterprise identifier of dpid2, instead of the enterprise identifier of ibmNetDispatcher, 1.3.6.1.4.1.2.6.144. However, the SNMP management utilities are able to determine the source of the trap because the data will contain an object identifier from within the ibmNetDispatcher MIB.

Turning the SNMP support on and off from the dscontrol command

The **dscontrol subagent start** command turns the SNMP support on. The **dscontrol subagent stop** command turns the SNMP support off.

For more information about the dscontrol command, see "dscontrol subagent — configure SNMP subagent" on page 338.

Using ipchains or iptables to reject all traffic to harden the Load Balancer machine (Linux systems)

Built into the Linux kernel is a firewall facility called ipchains. When Load Balancer and ipchains run concurrently, Load Balancer sees packets first, followed by ipchains. This allows the use of ipchains to harden a Linux Load Balancer machine, which could be, for example, a Load Balancer machine that is used to load balance firewalls.

When ipchains or iptables are configured as completely restricted (no inbound or outbound traffic permitted), the packet-forwarding portion of Load Balancer continues to function normally.

Note that ipchains and iptables ***cannot*** be used to filter incoming traffic before it is load balanced.

Some additional traffic must be permitted for all of Load Balancer to function properly. Some examples of this communication are:

- Advisors communicate between the Load Balancer machine and the backend servers.
- Load Balancer pings backend servers, reach targets, and high availability partner Load Balancer machines.
- User interfaces (graphical user interface, command line, and wizards) use RMI.
- backend servers must respond to pings from the Load Balancer machine.

In general, an appropriate ipchains strategy for the Load Balancer machines is to disallow all traffic, except that which is to or from the backend servers, the partner high availability Load Balancer, any reach targets, or any configuration hosts.

It is not recommended to activate iptables when running Load Balancer on Linux kernel version 2.4.10.x. Activation on this Linux kernel version can result in performance degradation over time.

To deactivate iptables, list the modules (`lsmod`) to see which modules are using ip_tables and ip_conntrack, then remove them by issuing `rmmod ip_tables` and `rmmod ip_conntrack`. When you reboot the machine these modules will be added again, so you need to repeat these step each time you reboot.

For more information, see "Problem: Linux iptables can interfere with the routing of packets" on page 268.

Using the Content Based Routing component

This section explains how to operate and manage the CBR component of Load Balancer.

Note: The Content Based Routing (CBR) component is not available on platforms that run a 64-bit JVM, except for HP-UX ia64. On HP-UX ia64, the CBR component runs as a 32-bit application. You can use the CBR forwarding method of Load Balancer's Dispatcher component to provide content-based routing without the use of Caching Proxy. See "Dispatcher's content-based routing (cbr forwarding method)" on page 41 for more information.

Starting and Stopping CBR

- Type **cbrserver** on a command line to start CBR.
- Type **cbrserver stop** on a command line to stop CBR.

CBR and Caching Proxy collaborate using the Caching Proxy plug-in API to handle HTTP and HTTPS (SSL) request. Caching Proxy must be running on the same machine in order for CBR to begin load balancing servers. Set up CBR and Caching Proxy as described in "CBR configuration example" on page 83.

Controlling CBR

After starting CBR, you can control it using either of the following methods:

- Configure CBR through the **cbrcontrol** command. The complete syntax of this command is described in Chapter 26, "Command reference for Dispatcher and CBR," on page 287. Some example uses are listed here.
- Configure CBR using the graphical user interface (GUI). Type **lbadmin** on the command line to open the GUI. See "GUI" on page 77 for more information on how to configure CBR using the GUI.

Using CBR logs

The logs used by CBR are similar to those used in Dispatcher. For more information, see "Using Load Balancer logs" on page 217.

Note:
In previous releases, for CBR you could change the log directory path in the Caching Proxy configuration file. Now you can change the directory path where the log gets stored in the cbrserver file. See "Changing the log file paths" on page 219.

Using the Site Selector component

Starting and stopping Site Selector

- Type **ssserver** on a command line to start Site Selector.
- Type **ssserver stop** on a command line to stop Site Selector.

Controlling Site Selector

After starting Site Selector, you can control it using either of the following methods:

- Configure Site Selector through the **sscontrol** command. The complete syntax of this command is described in Chapter 27, "Command reference for Site Selector," on page 341. Some example uses are listed here.
- Configure Site Selector using the graphical user interface (GUI). Type **lbadmin** on the command line to open the GUI. See "GUI" on page 96 for more information on how to configure Site Selector using the GUI.

Using Site Selector logs

The logs used by Site Selector are similar to those used in Dispatcher. For more description, see "Using Load Balancer logs" on page 217.

Using the Cisco CSS Controller component

Starting and stopping Cisco CSS Controller

1. Type **ccoserver** on a command line to start Cisco CSS Controller.
2. Type **ccoserver stop** on a command line to stop Cisco CSS Controller.

Controlling Cisco CSS Controller

After starting Cisco CSS Controller, you can control it using either of the following methods:

- Configure Cisco CSS Controller through the **ccocontrol** command. The complete syntax of this command is described in Chapter 28, "Command reference for Cisco CSS Controller," on page 367. Some example uses are listed here.
- Configure Cisco CSS Controller using the graphical user interface (GUI). Type **lbadmin** on the command line to open the GUI. See "GUI" on page 115 for more information on how to configure Cisco CSS Controller using the GUI.

Using Cisco CSS Controller logs

The logs used by Cisco CSS Controller are similar to those used in Dispatcher. For more description, see "Using Load Balancer logs" on page 217.

Using the Nortel Alteon Controller component

Starting and stopping Nortel Alteon Controller

1. Type **nalserver** on a command line to start Nortel Alteon Controller.
2. Type **nalserver stop** on a command line to stop Nortel Alteon Controller.

Controlling Nortel Alteon Controller

After starting Nortel Alteon Controller, you can control it using either of the following methods:

- Configure Nortel Alteon Controller through the **nalcontrol** command. The complete syntax of this command is described in Chapter 29, "Command reference for Nortel Alteon Controller," on page 385. Some example uses are listed here.
- Configure Nortel Alteon Controller using the graphical user interface (GUI). Type **lbadmin** on the command line to open the GUI. See "GUI" on page 134 for more information on how to configure Nortel Alteon Controller using the GUI.

Using Nortel Alteon Controller logs

The logs used by Nortel Alteon Controller are similar to those used in Dispatcher. For more description, see "Using Load Balancer logs" on page 217.

Using the Metric Server component

Starting and stopping Metric Server

Metric Server provides server load information to the Load Balancer. Metric Server resides on each of the servers that are being load balanced.

Linux and UNIX systems:

- On each server machine where Metric Server resides, type **metricserver start** on a command line to start Metric Server.
- On each server machine where Metric Server resides, type **metricserver stop** on a command line to stop Metric Server.

Windows systems:

Click **Start** > **Control Panel** > **Administrative Tools** > **Services**. Right-click **IBM Metric Server** and select Start. To stop the service, follow the same steps and select Stop.

Using Metric Server logs

Change the log level in the Metric Server startup script. You can specify a log level range of 0 through 5, similar to the log level range in Load Balancer logs. This will generate an agent log in the **...ms/logs** directory.

Chapter 24. Troubleshooting

This chapter helps you detect and resolve problems associated with Load Balancer.

- Before you call IBM service, see "Gathering troubleshooting information."
- Find the symptom you are experiencing in "Troubleshooting tables" on page 235.

Gathering troubleshooting information

Use the information in this section to gather the data that IBM service requires. The information is divided into the following subjects.

General information (always required)

For the Dispatcher component only, there is a problem determination tool that automatically gathers operating system-specific data and component-specified configuration files. To run this tool, type **lbpd** from the appropriate directory:

For AIX, HP-UX, Linux, and Solaris operating systems: /opt/ibm/edge/lb/servers/bin

For Windows systems: <*install_root*>ibm\edge\lb\servers\bin

This problem determination tool packages the data into files as follows:

For AIX, HP-UX, Linux, and Solaris operating systems: /opt/ibm/edge/lb/lbpmr.tar

For Windows systems: lbpmr.jar, which is created in the directory from which you run the **lbpd** tool.

Before you call IBM service, have the following information available.

- For Dispatcher only, the lbpmr file that is generated by the problem determination tool discussed above.
- In a high availability environment, configuration files from both Load Balancer machines. On all operating systems, use the script you use to load the configuration, or issue this command:

  ```
  dscontrol file save primary.cfg
  ```

  This command places the configuration file in the following directory:
  - AIX, HP-UX, Linux, and Solaris operating systems: /opt/ibm/edge/lb/servers/configuration/*component*
  - Windows operating systems: <*install_root*>ibm\edge\lb\servers\configuration*component*

- The operating system you are running and the version of that operating system. To determine the version of Load Balancer that you are running, refer to the product information in IBM Installation Manager.
- Issue the following command to obtain the current level of Java:

  ```
  java -fullversion
  ```
- Are you using Token Ring or Ethernet?
- Issue one of these commands to get protocol statistics and TCP/IP connection information:

  AIX, HP-UX, Linux, and Solaris operating systems:

  ```
  netstat -ni
  ```

  On Windows systems:

  ```
  ipconfig /all
  ```

  This is required from all servers and Load Balancer.
- Issue one of these command to get information on the route tables:

  On AIX, HP-UX, Linux, and Solaris systems:

  ```
  netstat -nr
  ```

  On Windows systems:

  ```
  route print
  ```

  This is required from all servers and Load Balancer.

High availability (HA) problems

Gather the following required information for problems in an HA environment.

- Set hamon.log at loglevel 5: `dscontrol set loglevel 5`.
- Set reach.log at loglevel 5: `dscontrol manager reach set loglevel 5`.
- Obtain the scripts, located as follows:

  AIX, HP-UX, Linux, and Solaris systems: /opt/ibm/edge/lb/servers/bin

  Windows systems: <*install_root*>ibm\edge\lb\servers\bin

  The script names are:

  goActive

  goStandby

  goIdle (if present)

  goInOp (if present)

  Also include the configuration files. See "General information (always required)" on page 231.

Advisor problems

Gather the following required information for advisor problems; for example, when advisors are mistakenly marking servers as down.

- Set the advisor log at loglevel 5:

  ```
  dscontrol advisor loglevel http 80 5
  ```

  or

  ```
  dscontrol advisor loglevel advisorName port loglevel
  ```

  or

  ```
  dscontrol advisor loglevel advisorName cluster:port loglevel
  ```

  or

  ```
  nalcontrol metriccollector set consultantID:serviceID:metricName
  loglevel value
  ```

This creates a log named ADV_*advisorName* log; for example, ADV_http.log. This log is located as follows:

AIX, HP-UX, Linux, and Solaris platforms: /opt/ibm/edge/lb/servers/logs/*component*

Windows platforms: <*install_root*>ibm\edge\lb\servers\logs*component*

Where *component* is:

dispatcher = Dispatcher

cbr = Content Based Routing

cco = Cisco CSS Controller

nal = Nortel Alteon Controller

ss = Site Selector

Note: When writing custom advisors, it is helpful to use the ADVLOG(*loglevel*,*message*) to verify the advisor is working correctly.

The ADVLOG call prints statements to the advisors log file when the level is less than the logging level associated with the advisors. A logging level of 0 will cause the statement to always be written. You cannot use ADVLOG from the constructor. The log file is not created until immediately after the custom advisor's constructor has completed because the log file name depends on information that is set in the constructor.

There is another way to debug your custom advisor that will avoid this limitation. You can use System.out.println(*message*) statements to print messages to a window. Edit the dsserver script and change `javaw` to `java` for the print statements to appear in the window. The window used to start dsserver must be kept open for the prints to appear. If you are using Windows platforms, you must stop the Dispatcher from running as a service and manually start it from a window to see the messages.

Refer to *Programming Guide for Edge Components* for more information on ADVLOG.

Content Based Routing problems

Gather the following required information for Content Based Routing problems.

- Issue this command to obtain the version: `cbrcontrol executor status`.
- Obtain the following files:
  - ibmproxy.conf, located as follows:

    AIX, HP-UX, Linux, and Solaris: /etc/

    Windows systems: <*install_root*>ibm\edge\cachingproxy\etc\en_US\
  - CBR configuration file, located as follows:

    AIX, HP-UX, Linux, and Solaris operating systems: /opt/ibm/edge/lb/servers/configurations/cbr

    Windows systems: <*install_root*>ibm\edge\lb\servers\configurations\cbr
  - Ensure that the correct entries are made in ibmproxy.conf. See "Step 1. Configure Caching Proxy to use CBR" on page 79.

Not able to hit the cluster

If you are not able to hit the cluster, it is possible that neither or both of the Load Balancer machines have the cluster aliased. To determine which machine owns the cluster:

1. On the same subnet and *not* on a Load Balancer machine or server:

   ```
   ping cluster
   arp -a
   ```

   If you are using Dispatcher's nat or cbr forwarding methods, ping the return address also.
2. Look through the arp output and match the MAC (16–digit hexadecimal address) to one of the netstat -ni outputs to determine which machine physically owns the cluster.
3. Use the following commands to interpret the output from both machines to see if they both have the cluster address.

   On AIX and HP-UX systems: `netstat -ni`

   On Linux and Solaris systems: `ifconfig -a`

   On Windows systems: `ipconfig /all`

If you do not get a response from the ping, and you are not using ULB, it is possible that neither machine has the cluster IP address aliased to its interface; for example, en0, tr0, and so forth.

All else fails

If you are unable to solve routing problems and all else has failed, issue the following command to run a trace on the network traffic:

- On AIX systems, from the Load Balancer machine:

  ```
  iptrace -a -s failingClientIPAddress -d clusterIPAddress -b iptrace.trc
  ```

  Run the trace, recreate the problem, then kill the process.
- On HP-UX systems:

  ```
  tcpdump -i lan0 host cluster and host client
  ```

  You may need to download tcpdump from one of the HP-UX GNU software archive sites.
- On Linux systems:

  ```
  tcpdump -i eth0 host cluster and host client
  ```

  Run the trace, recreate the problem, then kill the process.
- On Solaris:

  ```
  snoop -v clientIPAddress destinationIPAddress > snooptrace.out
  ```
- On Windows systems, a sniffer is required. Use the same inputs as for a filter.

You can also increase different log levels (for example, manager log, advisor log and so forth.) and investigate their output.

Upgrades

To identify a problem that is already fixed in a service release fix or patch, check for upgrades. To obtain a list of Edge Components defects fixed, refer to the WebSphere Application Server Web site Support page: http://www.ibm.com/software/webservers/appserv/was/support/. From the Support page, follow the link to the corrective service download site.

Java code

The correct version of Java code is installed as part the Load Balancer installation.

Helpful links

See "Reference Information" on page xv for links to support and library Web pages. The Web support page contains a link to Self-help information in the form of Technotes.

Troubleshooting tables

Refer to the following for:

- Dispatcher troubleshooting information — Table 12
- CBR troubleshooting information — Table 13 on page 240
- Site Selector troubleshooting information — Table 14 on page 241
- Cisco CSS Controller troubleshooting information — Table 15 on page 243
- Nortel Alteon Controller troubleshooting information — Table 16 on page 244
- Metric Server troubleshooting information — Table 17 on page 245

Table 12. Dispatcher troubleshooting table

Symptom	Possible Cause	Go to...
Dispatcher not running correctly	Conflicting port numbers	"Checking Dispatcher port numbers" on page 246
Configured a collocated server and it will not respond to load balanced requests	Wrong or conflicting address	"Problem: Dispatcher and server will not respond" on page 249
Connections from client machines not being served or connections timing out	• Wrong routing configuration • NIC not aliased to the cluster address • Server does not have loopback device aliased to the cluster address • Extra route not deleted • Port not defined for each cluster	"Problem: Dispatcher requests are not being balanced" on page 249
Client machines are not being served or are timing out	High availability not working	"Problem: Dispatcher high-availability function is not working" on page 250
Unable to add heartbeat (Windows platform)	Source address is not configured on an adapter	"Problem: Unable to add heartbeat (Windows platform)" on page 250
Advisors not working correctly with wide area	Advisors are not running on remote machines	"Problem: Advisors not working correctly" on page 250
On a backend server running Windows Server 2008, memload.exe crashes	The Windows Server 2008 registry might not be populated with the performance keys that these tools require. This application crash would be reported from the cpuload application.	"Problem: On a Windows Server 2008 backend server, memload.exe crashes" on page 250

Table 12. Dispatcher troubleshooting table (continued)

Symptom	Possible Cause	Go to...
Dispatcher, Microsoft IIS, and SSL are not working or will not continue	Unable to send encrypted data across protocols	"Problem: Dispatcher, Microsoft IIS, and SSL do not work (Windows platform)" on page 251
Connection to remote machine refused	Older version of the keys is still being used	"Problem: Dispatcher connection to a remote machine" on page 251
The dscontrol or lbadmin command fails with 'Server not responding' or 'unable to access RMI server' message	1. Commands fail due to socksified stack. Or commands fail due to not starting dsserver 2. RMI ports are not set correctly 3. Host file has incorrect local host	"Problem: dscontrol or lbadmin command fails" on page 251
"Cannot Find the File..." error message, when running Netscape as default browser to view online help (Windows platform)	Incorrect setting for HTML file association	"Problem: "Cannot find the file..." error message when trying to view online Help (Windows platform)" on page 252
Graphical user interface does not start correctly	Insufficient paging space	"Problem: Graphical user interface (GUI) does not start correctly" on page 252
Error running Dispatcher with Caching Proxy installed	Caching Proxy file dependency	"Problem: Error running Dispatcher with Caching Proxy installed" on page 252
Graphical user interface does not display correctly.	Resolution is incorrect.	"Problem: Graphical user interface (GUI) does not display correctly" on page 252
Help panels sometimes disappear behind other windows	Java limitation	"Problem: On Windows platform, help windows sometimes disappear behind other open windows" on page 252
Load Balancer cannot process and forward a frame	Need a unique MAC address for each NIC	"Problem: Load Balancer cannot process and forward a frame" on page 252
Blue screen appears	No installed and configured network card	"Problem: A blue screen displays when you start the Load Balancer executor" on page 253
Path to Discovery prevents return traffic	The cluster is aliased on the loopback	"Problem: Path to Discovery prevents return traffic with Load Balancer" on page 253
High availability in the Wide Area mode of Load Balancer does not work.	Remote Dispatcher must be defined as a server in a cluster on local Dispatcher	"Problem: High availability in the Wide Area mode of Load Balancer does not work" on page 254

Table 12. Dispatcher troubleshooting table (continued)

Symptom	Possible Cause	Go to...
GUI hangs (or unexpected behavior) when trying to load a large configuration file.	Java does not have access to enough memory to handle such a large change to the GUI	"Problem: GUI hangs (or unexpected behavior) when trying to load a large configuration file" on page 254
IP addresses not resolving correctly over the remote connection	When using a remote client over a secure socks implementation, fully qualified domain names or host names might not resolve to the correct IP address	"Problem: IP addresses not resolving correctly over the remote connection" on page 255
Korean Load Balancer interface displays overlapping or undesirable fonts on AIX and Linux systems	Default fonts must be changed	"Problem: Korean Load Balancer interface displays overlapping or undesirable fonts on AIX and Linux systems" on page 255
On Windows systems, after aliasing the MS Loopback adapter, when issuing certain commands such as hostname, the OS will incorrectly respond with the alias address	In the network connections list, the newly added alias must not be listed above the local address	"Problem: On Windows systems, alias address is returned instead of local address when issuing commands such as hostname" on page 256
Unexpected GUI behavior when using Windows platform paired with Matrox AGP video card	Problem occurs when using Matrox AGP video cards while running the Load Balancer GUI	"Problem: On Windows platform, unexpected GUI behavior when using Matrox AGP video cards" on page 256
Unexpected behavior, such as system hang, when executing "rmmod ibmlb" on Linux systems	Problem occurs when manually removing the Load Balancer kernel module (ibmlb).	"Problem: Unexpected behavior when executing "rmmod ibmlb" (Linux systems)" on page 256
Slow response time when running commands on the Dispatcher machine	Slow response time can be due to machine overloading from a high volume of client traffic	"Problem: Slow response time running commands on Dispatcher machine" on page 256
For Dispatcher's mac forwarding method, SSL or HTTPS advisor not registering server loads	Problem occurs because the SSL server application not configured with the cluster IP address	"Problem: SSL or HTTPS advisor not registering server loads (when using mac-forwarding)" on page 257
Disconnect from host when using remote Web administration through Netscape	Disconnect from host will occur when resize the browser window	"Problem: Disconnect from host occurs when resize Netscape browser window while using Web administration" on page 257
On Windows platform, corrupted Latin-1 national characters appear in command prompt	Change font properties of command prompt window	"Problem: On Windows systems, corrupted Latin-1 national characters appear in command prompt window" on page 257

Table 12. Dispatcher troubleshooting table (continued)

Symptom	Possible Cause	Go to...
On HP-UX platform, the following message occurs: java.lang.OutOfMemoryError unable to create new native thread	Some HP-UX installations by default allow 64 threads per process. This is insufficient.	"Problem: On HP-UX, Java out of memory or thread error occurs" on page 257
On Windows platform, advisors and reach targets mark all servers down	Task offloading is not disabled or may need to enable ICMP.	"Problem: On Windows systems, advisors and reach targets mark all servers down" on page 258
On Windows platform, problem resolving IP address to hostname when more than one address is configured to an adapter	The IP address you want as your hostname must appear first in the registry.	"Problem: On Windows platform, resolving IP address to host name when more than one address is configured to an adapter" on page 258
On Windows platform, advisors not working in a high availability setup after a network outage	When the system detects a network outage, it clears its Address Resolution Protocol (ARP) cache	"Problem: On Windows systems, after network outage, advisors not working in a high availability setup" on page 259
On Linux systems, "IP address add" command and multiple cluster loopback aliases are incompatible	When aliasing more than one address on the loopback device, should use **ifconfig** command, not **ip address add**	"Problem: On Linux systems, do not use "IP address add" command when aliasing multiple clusters on the loopback device" on page 260
Error message: "Router address not specified or not valid for port method" when trying to add a server	Checklist of information to determine the problem that has occurred when adding a server	"Problem: "Router address not specified or not valid for port method" error message" on page 260
On Solaris systems, Load Balancer processes end when you exit the terminal session window from which they started	Use the **nohup** command to prevent the processes that you started from receiving a hangup signal when you exit the terminal session.	"Problem: On Solaris systems, Load Balancer processes end when you exit the terminal window from which they started" on page 261
Slow down occurs when loading Load Balancer configurations	The delay might be due to Domain Name System (DNS) calls that are made to resolve and verify the server address.	"Problem: Delay occurs while loading a Load Balancer configuration" on page 261
On Windows systems, the following error message appears: There is an IP address conflict with another system on the network	If high availability is configured, cluster addresses may be configured on both machines for a brief period which causes this error message to appear.	"Problem: On Windows systems, an IP address conflict error message appears" on page 261
Both primary and backup machines are active in a high availability configuration	This problem may occur when the go scripts do not run on either primary or backup machine.	"Problem: Both primary and backup machines are active in a high availability configuration" on page 261

Table 12. Dispatcher troubleshooting table (continued)

Symptom	Possible Cause	Go to...
Client requests fail when Dispatcher attempts to return large page responses	Client requests that result in large page responses timeout if the maximum transmit unit (MTU) is not set properly on the Dispatcher machine when using nat or cbr forwarding.	"Problem: Client requests fail when attempting the return of large page responses" on page 262
On Windows systems, "Server not responding" error occurs when issuing a dscontrol or lbadmin command	When more than one IP address exists on a Windows system and the host file does not specify the address to associate with the hostname.	"Problem: On Windows systems, "Server not responding" error occurs when issuing dscontrol or lbadmin" on page 262
High availability Dispatcher machines may fail to synchronize on Linux for S/390 on qeth devices	When using high availability on Linux for S/390 with the qeth network driver, the active and standby Dispatchers may fail to synchronize.	"Problem: High availability Dispatcher machines may fail to synchronize on Linux for S/390 systems on qeth drivers" on page 262
Tips on configuring the high availability feature for Load Balancer	The tips will help alleviate high availability problems such as: • Connections dropped after takeover • Partner machines unable to synchronize • Requests erroneously directed to the backup partner machine	"Problem: Tips on configuring high availability" on page 263
Dispatcher MAC forwarding configuration limitations with zSeries and S/390 platforms	On Linux, there are limitations when using zSeries or S/390 servers that have Open System Adapter (OSA) cards. Possible workarounds are provided.	"Problem: On Linux, Dispatcher configuration limitations when using zSeries or S/390 servers that have Open System Adapter (OSA) cards" on page 264
On some Red Hat Linux versions, a memory leak occurs when running Load Balancer configured with the manager and advisors	The IBM Java SDK versions of the JVM and the Native POSIX Thread Library (NPTL) shipped with some Linux distributions, such as Red Hat Enterprise Linux 3.0, can cause the memory leak to occur.	"Problem: On some Linux versions, a memory leak occurs when running Dispatcher configured with the manager and advisors" on page 266
On SUSE Linux Enterprise Server 9, Dispatcher report indicates that packets are forwarded (packet-count increases), however packets never actually reach the backend server	The iptables NAT module is loaded. There is a possible, but unconfirmed, error in this version of iptables that causes strange behavior when interacting with Dispatcher.	"Problem: On SUSE Linux Enterprise Server 9, Dispatcher forwards packets, but the packets do not reach the backend server" on page 267

Table 12. Dispatcher troubleshooting table (continued)

Symptom	Possible Cause	Go to...
On Windows systems, when using Dispatcher's high availability feature, problems might occur during takeover	If the goScript that configures the cluster IP address on the active machine runs before the goScript to unconfigure the IP cluster address on the backup machine, problems might occur.	"Problem: On Windows system, IP address conflict message appears during high availability takeover" on page 267
On Linux systems, iptables can interfere with the routing of packets	Linux iptables can interfere with load balancing of traffic and must be disabled on the Load Balancer machine.	"Problem: Linux iptables can interfere with the routing of packets" on page 268
A Java fileset warning message appears when installing service fixes or installing natively, using system packaging tools	The product installation consists of several packages which are not required to be installed on the same machine, so each of these packages installs a Java fileset. When installed on the same machine a warning messages stating that the Java fileset is also owned by another fileset.	
Upgrading the Java fileset provided with the Load Balancer installations	If a problem is found with the Java file set, you should report the problem to IBM Service so that you can receive an upgrade for the Java file set that was provided with the Load Balancer installation.	"Upgrading the Java file set provided with the Load Balancer installation" on page 268
Persistent connections might drop during high availability takeover on a Windows platform	On Microsoft Windows operating systems, persistent connections might drop during a high availability takeover. This problem exists only when you have a collocated server that uses the MAC forwarding method.	"Problem: Persistent connections might drop during high availability takeover" on page 269

Table 13. CBR Troubleshooting table

Symptom	Possible Cause	Go to...
CBR not running correctly	Conflicting port numbers	"Checking CBR port numbers" on page 246
The cbrcontrol or lbadmin command fails with 'Server not responding' or 'unable to access RMI server' message	Commands fail due to socksified stack. Or commands fail due to not starting cbrserver	"Problem: cbrcontrol or lbadmin command fails" on page 269
Requests are not being load balanced	Caching Proxy was started before the executor was started	"Problem: Requests not being load balanced" on page 269

Table 13. CBR Troubleshooting table (continued)

On Solaris, the cbrcontrol executor start command fails with 'Error: Executor was not started.' message	Command fails because the system IPC defaults may need to be modified, or link to library is incorrect.	"Problem: On Solaris systems, cbrcontrol executor start command fails" on page 270
URL rule does not work	Syntactical or configuration error	"Problem: Syntactical or configuration error" on page 270
Unexpected GUI behavior when using Windows systems paired with Matrox AGP video card	Problem occurs when using Matrox AGP video cards while running the Load Balancer GUI	"Problem: On Windows platform, unexpected GUI behavior when using Matrox AGP video cards" on page 270
GUI hangs (or unexpected behavior) when trying to load a large configuration file.	Java does not have access to enough memory to handle such a large change to the GUI	"Problem: GUI hangs (or unexpected behavior) when trying to load a large configuration file" on page 254
Disconnect from host when using remote Web administration through Netscape	Disconnect from host will occur when resize the browser window	"Problem: Disconnect from host occurs when resize Netscape browser window while using Web administration" on page 270
On Windows platform, corrupted Latin-1 national characters appear in command prompt	Change font properties of command prompt window	"Problem: On Windows platform, corrupted Latin-1 national characters appear in command prompt window" on page 270
On HP-UX platform, the following message occurs: java.lang.OutOfMemoryError unable to create new native thread	Some HP-UX installations by default allow 64 threads per process. This is insufficient.	"Problem: On HP-UX, Java out of memory/ thread error occurs" on page 271
On Windows platform, advisors and reach targets mark all servers down	Task offloading is not disabled or may need to enable icmp.	"Problem: On Windows systems, advisors and reach targets mark all servers down" on page 271
On Windows platform, problem resolving IP address to host name when more than one address is configured to an adapter	The IP address you want as your hostname must appear first in the registry.	"Problem: On Windows systems, resolving IP address to host name when more than one address is configured to an adapter" on page 271
On Solaris systems, Load Balancer processes end when you exit the terminal session window from which they started	Use the **nohup** command to prevent the processes that you started from receiving a hangup signal when you exit the terminal session.	"Problem: On Solaris systems, Load Balancer processes end when you exit the terminal window from which they started" on page 261

Table 14. Site Selector troubleshooting table

Symptom	Possible Cause	Go to...
Site Selector not running correctly	Conflicting port number	"Checking Site Selector port numbers" on page 247

Table 14. Site Selector troubleshooting table (continued)

Symptom	Possible Cause	Go to...
Site Selector does not round-robin incoming requests from Solaris client	Solaris systems run a "name service cache daemon"	"Problem: Site Selector does not round-robin traffic from Solaris clients" on page 272
The sscontrol or lbadmin command fails with 'Server not responding' or 'unable to access RMI server' message	Commands fail due to socksified stack. Or commands fail due to not starting ssserver.	"Problem: sscontrol or lbadmin command fails" on page 272
ssserver fails to start on Windows platform	Windows systems do not require the host name to be in the DNS.	"Problem: The ssserver is failing to start on Windows platform" on page 272
Machine with duplicate routes not load balancing correctly — name resolution appears to fail	Site Selector machine with multiple adapters attached to the same subnet	"Problem: Site Selector with duplicate routes not load balancing correctly" on page 273
Unexpected GUI behavior when using Windows platform paired with Matrox AGP video card	Problem occurs when using Matrox AGP video cards while running the Load Balancer GUI	"Problem: On Windows platform, unexpected GUI behavior when using Matrox AGP video cards" on page 273
GUI hangs (or unexpected behavior) when trying to load a large configuration file.	Java does not have access to enough memory to handle such a large change to the GUI	"Problem: GUI hangs (or unexpected behavior) when trying to load a large configuration file" on page 254
Disconnect from host when using remote Web administration through Netscape	Disconnect from host will occur when resize the browser window	"Problem: Disconnect from host occurs when resize Netscape browser window while using Web administration" on page 273
On Windows platform, corrupted Latin-1 national characters appear in command prompt	Change font properties of command prompt window	"Problem: On Windows platform, corrupted Latin-1 national characters appear in command prompt window" on page 273
On HP-UX platform, the following message occurs: java.lang.OutOfMemoryError unable to create new native thread	Some HP-UX installations by default allow 64 threads per process. This is insufficient.	"Problem: On HP-UX, Java out of memory/thread error occurs" on page 273
On Windows platform, advisors and reach targets mark all servers down	Task offloading is not disabled or may need to enable icmp.	"Problem: On Windows systems, advisors and reach targets mark all servers down" on page 274
On Solaris systems, Load Balancer processes end when you exit the terminal session window from which they started	Use the **nohup** command to prevent the processes that you started from receiving a hangup signal when you exit the terminal session.	"Problem: On Solaris systems, Load Balancer processes end when you exit the terminal window from which they started" on page 261

Table 15. Controller for Cisco CSS Switches troubleshooting table

Symptom	Possible Cause	Go to...
ccoserver will not start	Conflicting port numbers	"Checking Cisco CSS Controller port numbers" on page 248
The ccocontrol or lbadmin command fails with 'Server not responding' or 'unable to access RMI server' message	Commands fail due to socksified stack. Or commands fail due to not starting ccoserver.	"Problem: ccocontrol or lbadmin command fails" on page 274
receive error: **Cannot create registry on port 13099**	Expired product license	"Problem: Cannot create registry on port 13099" on page 274
Unexpected GUI behavior when using Windows platform paired with Matrox AGP video card	Problem occurs when using Matrox AGP video cards while running the Load Balancer GUI	"Problem: On Windows platform, unexpected GUI behavior when using Matrox AGP video cards" on page 275
Received a connection error when adding a consultant	Configuration settings are incorrect on the switch or the controller	"Problem: Received a connection error when adding a consultant" on page 275
Weights are not being updated on the switch	Communication between the controller or the switch is unavailable or interrupted	"Problem: Weights are not being updated on the switch" on page 275
Refresh command did not update the consultant configuration	Communication between the switch and the controller is unavailable or interrupted	"Problem: Refresh command did not update the consultant configuration" on page 275
GUI hangs (or unexpected behavior) when trying to load a large configuration file.	Java does not have access to enough memory to handle such a large change to the GUI	"Problem: GUI hangs (or unexpected behavior) when trying to load a large configuration file" on page 254
Disconnect from host when using remote Web administration through Netscape	Disconnect from host will occur when resize the browser window	"Problem: Disconnect from host occurs when resize Netscape browser window while using Web administration" on page 275
On Windows platform, corrupted Latin-1 national characters appear in command prompt	Change font properties of command prompt window	"Problem: On Windows platform, corrupted Latin-1 national characters appear in command prompt window" on page 276
On HP-UX platform, the following message occurs: java.lang.OutOfMemoryError unable to create new native thread	Some HP-UX installations by default allow 64 threads per process. This is insufficient.	"Problem: On HP-UX, Java out of memory/ thread error occurs" on page 276
On Solaris systems, Load Balancer processes end when you exit the terminal session window from which they started	Use the **nohup** command to prevent the processes that you started from receiving a hangup signal when you exit the terminal session.	"Problem: On Solaris systems, Load Balancer processes end when you exit the terminal window from which they started" on page 261

Table 16. Nortel Alteon Controller troubleshooting table

Symptom	Possible Cause	Go to...
nalserver will not start	Conflicting port numbers	"Checking Nortel Alteon Controller port numbers" on page 248
The nalcontrol or lbadmin command fails with 'Server not responding' or 'unable to access RMI server' message	Commands fail due to socksified stack. Or commands fail due to not starting nalserver.	"Problem: nalcontrol or lbadmin command fails" on page 276
receive error: **Cannot create registry on port 14099**	Expired product license	"Problem: Cannot create registry on port 14099" on page 277
Unexpected GUI behavior when using Windows platform paired with Matrox AGP video card	Problem occurs when using Matrox AGP video cards while running the Load Balancer GUI	"Problem: On Windows platform, unexpected GUI behavior when using Matrox AGP video cards" on page 277
GUI hangs (or unexpected behavior) when trying to load a large configuration file.	Java does not have access to enough memory to handle such a large change to the GUI	"Problem: GUI hangs (or unexpected behavior) when trying to load a large configuration file" on page 254
Disconnect from host when using remote Web administration through Netscape	Disconnect from host will occur when resize the browser window	"Problem: Disconnect from host occurs when resize Netscape browser window while using Web administration" on page 277
Received a connection error when adding a consultant	Configuration settings are incorrect on the switch or the controller	"Problem: Received a connection error when adding a consultant" on page 277
Weights are not being updated on the switch	Communication between the controller or the switch is unavailable or interrupted	"Problem: Weights are not being updated on the switch" on page 278
Refresh command did not update the consultant configuration	Communication between the switch and the controller is unavailable or interrupted	"Problem: Refresh command did not update the consultant configuration" on page 278
On Windows platform, corrupted Latin-1 national characters appear in command prompt	Change font properties of command prompt window	"Problem: On Windows systems, corrupted Latin-1 national characters appear in command prompt window" on page 278
On HP-UX platform, the following message occurs: java.lang.OutOfMemoryError unable to create new native thread	Some HP-UX installations by default allow 64 threads per process. This is insufficient.	"Problem: On HP-UX, Java out of memory/ thread error occurs" on page 278
On Solaris systems, Load Balancer processes end when you exit the terminal session window from which they started	Use the **nohup** command to prevent the processes that you started from receiving a hangup signal when you exit the terminal session.	"Problem: On Solaris systems, Load Balancer processes end when you exit the terminal window from which they started" on page 261

Table 17. Metric Server troubleshooting table

Symptom	Possible Cause	Go to...
Metric Server IOException on Windows platform running .bat or .cmd user metric files	Full metric name is required	"Problem: Metric Server IOException on Windows platform running .bat or .cmd user metric files" on page 278
Metric Server not reporting the load information to the Load Balancer machine	Possible causes include: • no key files on Metric Server machine • host name of Metric Server machine not registered with local nameserver • the /etc/hosts file has the local hostname resolving to the loopback address 127.0.0.1	"Problem: Metric Server not reporting loads to Load Balancer machine" on page 279
Metric Server log reports "Signature is necessary for access to agent" when key files transferred to server	Key file fails authorization due to corruption.	"Problem: Metric Server log reports "Signature is necessary for access to agent"" on page 279
On AIX systems, when running Metric Server under heavy stress on a multi-processor system (AIX 5.1), `ps -vg` command output may become corrupted	APAR IY33804 corrects this known AIX problem	"Problem: On AIX systems, while running Metric Server under heavy stress, ps –vg command output may become corrupted" on page 279
Configuring Metric Server in a two-tier configuration with Site Selector load-balancing across high-availability Dispatchers	Metric Server (residing in the second-tier) is not configured to listen on a new IP address.	"Problem: Configuring Metric Server in a two-tier configuration with Site Selector load-balancing across high-availability Dispatchers" on page 279
Scripts (metricserver, cpuload, memload) running on multi-CPU Solaris machines produce unwanted console messages	This behavior is due to the use of the VMSTAT system command to gather CPU and memory statistics from the kernel.	"Problem: Scripts, running on multi-CPU Solaris machines, produce unwanted console messages" on page 281
On Solaris systems, Load Balancer processes end when you exit the terminal session window from which they started	Use the **nohup** command to prevent the processes that you started from receiving a hangup signal when you exit the terminal session.	"Problem: On Solaris systems, Load Balancer processes end when you exit the terminal window from which they started" on page 261
Metric value returns -1 after starting Metric Server	This problem may be caused due to the key files losing their integrity during the transfer of the key files to the client.	"Problem: After starting Metric Server, metric value returns -1" on page 281

Checking Dispatcher port numbers

If you are experiencing problems running Dispatcher, it may be that one of your applications is using a port number that the Dispatcher normally uses. Be aware that the Dispatcher server uses the following port numbers:

- 10099 to receive commands from dscontrol
- 10004 to send metric queries to Metric Server
- 10199 for the RMI server port

If another application is using one of the Dispatcher's port numbers, you can either change the Dispatcher's port numbers *or* change the application's port number.

Change the Dispatcher's port numbers by doing the following:

- To change the port used to receive commands
  - Modify the LB_RMIPORT variable at the top of the dsserver file to the port that you want Dispatcher to receive commands.
- To change the port used to receive metric reports from Metric Server
  - Modify the RMI_PORT variable in the metricserver file to the port that you want Dispatcher to communicate with Metric Server.
  - Provide the metric_port argument when the manager is started. See the description of the **dscontrol manager start** command syntax "dscontrol manager — control the manager" on page 313

Change the application's RMI port number by doing the following:

- To change the port used by the application
  - Modify the LB_RMISERVERPORT variable in the dsserver file to the port that you want the application to use. (The default value of the RMI port used by the application is 10199.)

Note: For Windows platform, dsserver and metricserver files are in the <*install_root*>ibm\edge\lb\bin directory. For other platforms, these file are in the /usr/bin/ directory.

Checking CBR port numbers

If you are experiencing problems running CBR, it may be that one of your applications is using a port number that CBR normally uses. Be aware that CBR uses the following port number:

- 11099 to receive commands from cbrcontrol
- 10004 to send metric queries to Metric Server
- 11199 for the RMI server port

Note: The Content Based Routing (CBR) component is not available on platforms that run a 64-bit JVM, except for HP-UX ia64. On HP-UX ia64, the CBR component runs as a 32-bit application. You can use the CBR forwarding method of Load Balancer's Dispatcher component to provide content-based routing without the use of Caching Proxy. See "Dispatcher's content-based routing (cbr forwarding method)" on page 41 for more information.

If another application is using one of the CBR's port numbers, you can either change the CBR's port numbers *or* change the application's port number.

Change the CBR's port numbers by doing the following:

- To change the port used to receive commands
  - Modify the LB_RMIPORT variable at the top of the cbrserver file to the port that you want CBR to receive commands.
- To change the port used to receive metric reports from Metric Server
  - Modify the RMI_PORT variable in the metricserver file to the port that you want CBR to communicate with Metric Server.
  - Provide the metric_port argument when the manager is started. See the description of the **manager start** command syntax "dscontrol manager — control the manager" on page 313

Change the application's RMI port number by doing the following:

- To change the port used by the application
  - Modify the LB_RMISERVERPORT variable at the top of the cbrserver file to the port that you want the application to use. (The default value of the RMI port used by the application is 11199.)

Note: For Windows platform, cbrserver and metricserver files are in the <*install_root*>ibm\edge\lb\bin directory. For other platforms, these file are in the /usr/bin/ directory.

Checking Site Selector port numbers

If you are experiencing problems running the Site Selector component, it may be that one of your applications is using a port number that Site Selector normally uses. Be aware that Site Selector uses the following port numbers:

- 12099 to receive commands from sscontrol
- 10004 to send metric queries to Metric Server
- 12199 for the RMI server port

If another application is using one of the Site Selector's port numbers, you can either change the Site Selector's port numbers *or* change the application's port number.

Change the Site Selector's port numbers by doing the following:

- To change the port used to receive commands,
  - Modify the LB_RMIPORT variable at the top of the ssserver file to the port that you want Site Selector to receive commands.
- To change the port used to receive metric reports from Metric Server
  - Modify the RMI_PORT variable in the metricserver file to the port that you want Site Selector to communicate with Metric Server.
  - Provide the metric_port argument when the manager is started. See the description of the **manager start** command syntax "sscontrol manager — control the manager" on page 350

Change the application's RMI port number by doing the following:

- To change the port used by the application
  - Modify the LB_RMISERVERPORT variable at the top of the ssserver file to the port that you want the application to use. (The default value of the RMI port used by the application is 12199.)

Note: For Windows platform, ssserver and metricserver files are in the <*install_root*>ibm\edge\lb\bin directory. For other platforms, these file are in the /usr/bin/ directory.

Checking Cisco CSS Controller port numbers

If you are experiencing problems running the Cisco CSS Controller component, it may be that another application is using one of the port numbers used by Cisco CSS Controller's ccoserver. Be aware that Cisco CSS Controller uses the following port numbers:

13099 to receive commands from ccocontrol

10004 to send metric queries to Metric Server

13199 for the RMI server port

If another application is using one of the Cisco CSS Controller's port numbers, you can either change the port numbers for Cisco CSS Controller *or* change the application's port number.

Change the Cisco CSS Controller's port numbers by doing the following:

- To change the port used to receive commands from ccocontrol, modify the CCO_RMIPORT variable in the ccoserver file. Change from 13099 to the port on which you want Cisco CSS Controller to receive ccocontrol commands.
- To change the port used to receive metric reports from Metric Server:
  1. Modify the RMI_PORT variable in the metricserver file. Change 10004 to the port on which you want Cisco CSS Controller to communicate with Metric Server.
  2. Provide the metric_port argument when you start the consultant.

Change the application's RMI port number by doing the following:

- To change the port used by the application
  - Modify the CCO_RMISERVERPORT variable at the top of the ccoserver file to the port that you want the application to use. (The default value of the RMI port used by the application is 13199.)

Note: For Windows platform, ccoserver and metricserver files are in the <*install_root*>ibm\edge\lb\bin directory. For other platforms, these file are in the /usr/bin directory.

Checking Nortel Alteon Controller port numbers

If you are experiencing problems running the Nortel Alteon Controller component, it may be that another application is using one of the port numbers used by Nortel Alteon Controller's nalserver. Be aware that Nortel Alteon Controller uses the following port numbers:

14099 to receive commands from nalcontrol

10004 to send metric queries to Metric Server

14199 for the RMI server port

If another application is using one of the Nortel Alteon Controller's port numbers, you can either change the port numbers for Nortel Alteon Controller *or* change the port numbers for the applicaton.

Change the port numbers for Nortel Alteon Controller by doing the following:

- To change the port used to receive commands from nalcontrol, modify the NAL_RMIPORT variable in the nalserver file. Change from 14099 to the port on which you want Nortel Alteon Controller to receive nalcontrol commands.
- To change the port used to receive metric reports from Metric Server:
  1. Modify the RMI_PORT variable in the metricserver file. Change 10004 to the port on which you want Nortel Alteon Controller to communicate with Metric Server.
  2. Provide the metric_port argument when you start the consultant.

Change the application's RMI port number by doing the following:

- To change the port used by the application
  - Modify the NAL_RMISERVERPORT variable at the top of the nalserver file to the port that you want the application to use. (The default value of the RMI port used by the application is 14199.)

Note: For Windows platform, nalserver and metricserver files are in the <*install_root*>ibm\edge\lb\bin directory. For other platforms, these file are in the /usr/bin directory.

Solving common problems—Dispatcher

Problem: Dispatcher will not run

This problem can occur when another application is using one of the ports used by the Dispatcher. For more information, go to "Checking Dispatcher port numbers" on page 246.

Problem: Dispatcher and server will not respond

This problem occurs when another address is being used other than the address specified. When collocating the Dispatcher and server, be sure that the server address used in the configuration is the NFA address or is configured as collocated. Also, check the host file for the correct address.

Problem: Dispatcher requests are not being balanced

This problem has symptoms such as connections from client machines not being served or connections timing out. Check the following to diagnose this problem:

1. Have you configured the nonforwarding address, clusters, ports, and servers for routing? Check the configuration file.
2. Is the network interface card aliased to the cluster address? For AIX, HP-UX, Linux, and Solaris operating systems, use `netstat -ni` to check.
3. Does the loopback device on each server have the alias set to the cluster address? For AIX, HP-UX, Linux, and Solaris operating systems, use `netstat -ni` to check.
4. Is the extra route deleted? For AIX, HP-UX, Linux, and Solaris operating systems, use `netstat -nr` to check.
5. Use the **dscontrol cluster status** command to check the information for each cluster you have defined. Make sure you have a port defined for each cluster.
6. Use the **dscontrol server report ::** command to make sure that your servers are neither down nor set to a weight of zero.

For Windows and other platforms, see also "Setting up server machines for load balancing" on page 55.

Problem: Dispatcher high-availability function is not working

This problem appears when a Dispatcher high-availability environment is configured and connections from the client machines are not being served or are timing out. Check the following to correct or diagnose the problem:

- Make sure you have created the goActive, goStandby, and goInOp scripts, and place them in the bin directory where Dispatcher is installed. For more information on these scripts, see "Using scripts" on page 167
- For **AIX**, **HP-UX**, **Linux,** and **Solaris** systems, make sure the goActive, goStandby, and goInOp scripts have `execute permission` set.
- For Windows systems, be sure to configure the nonforwarding address using the **executor configure** command.

The following steps are an effective way to test that high availability scripts are functioning properly:

1. gather a report by issuing `netstat -an` and `ifconfig -a` from the machine
2. run the `goActive` script
3. run the `goStandby` script
4. once again, gather a report by issuing `netstat -an` and `ifconfig -a` commands

The two reports are identical if the scripts are properly configured.

Problem: Unable to add heartbeat (Windows platform)

This Windows platform error occurs when the source address is not configured on an adapter. Check the following to correct or diagnose the problem.

- Be sure to configure the nonforwarding address using either the token-ring or Ethernet interface and issuing either of the following commands:

  ```
  dscontrol executor configure <ip address>
  ```

Problem: Advisors not working correctly

If you are using wide area support, and your advisors do not seem to work correctly, make sure that they are started on both the local and the remote Dispatchers.

An ICMP ping is issued to the servers before the advisor request. If a firewall exists between Load Balancer and the servers, ensure that pings are supported across the firewall. If this setup poses a security risk to your network, modify the `java` statement in `dsserver` to turn off all pings to the servers by adding the `java` property:

```
LB_ADV_NO_PING="true"
java  -DLB_ADV_NO_PING="true"
```

See "Using remote advisors with Dispatcher's wide area support" on page 186.

Problem: On a Windows Server 2008 backend server, memload.exe crashes

When Load Balancer is connecting to a backend server that runs Windows Server 2008, the metric collection features might cause the memload.exe application to stop running unexpectedly.

The crash occurs because the Windows Server 2008 registry might not be populated with the performance keys that these tools require. This application crash would be reported from the cpuload application.

Refer to the following Knowledge Base topic from Microsoft for steps on how to address this problem: http://support.microsoft.com/kb/300956

Problem: Dispatcher, Microsoft IIS, and SSL do not work (Windows platform)

When using Dispatcher, Microsoft IIS, and SSL, if they do not work together, there may be a problem with enabling SSL security. For more information about generating a key pair, acquiring a certificate, installing a certificate with a key pair, and configuring a directory to require SSL, see the *Microsoft Information and Peer Web Services* documentation.

Problem: Dispatcher connection to a remote machine

Dispatcher uses keys to allow you to connect to a remote machine and configure it. The keys specify an RMI port for the connection. It is possible to change the RMI port for security reasons or conflicts. When you change the RMI ports, the filename of the key is different. If you have more than one key in your keys directory for the same remote machine, and they specify different RMI ports, the command line will only try the first one it finds. If it is the incorrect one, the connection will be refused. The connection will not occur unless you delete the incorrect key.

Problem: dscontrol or lbadmin command fails

1. The dscontrol command returns: **Error: Server not responding**. Or, the lbadmin command returns: **Error: unable to access RMI server**. These errors can result when your machine has a socksified stack. To correct this problem, edit the socks.cnf file to contain the following lines:

   ```
   EXCLUDE-MODULE java
   EXCLUDE-MODULE javaw
   ```

2. The administration consoles for Load Balancer interfaces (command line, graphical user interface, and wizards) communicate with dsserver using remote method invocation (RMI). The default communication uses three ports; each port is set in the dsserver start script:
   - 10099 to receive commands from dscontrol
   - 10004 to send metric queries to Metric Server
   - 10199 for the RMI server port

   This can cause problems when one of the administration consoles runs on the same machine as a firewall or through a firewall. For example, when Load Balancer runs on the same machine as a firewall, and you issue dscontrol commands, you might see errors such as **Error: Server not responding**.

   To avoid this problem, edit the dsserver script file to set the port used by RMI for the firewall (or other application). Change the line: LB_RMISERVERPORT=10199 to LB_RMISERVERPORT=*yourPort*. Where *yourPort* is a different port.

   When complete, restart dsserver and open traffic for ports 10099, 10004, 10199, and 10100, or for the chosen port for the host address from which the administration console will be run.
3. These errors can also occur if you have not already started **dsserver**.
4. If there are multiple adapters on the machine, you must designate which adapter that dsserver is to use by adding the following in the dsserver script:`java.rmi.server.hostname=<host_name or IPaddress>`

   For example: java -Djava.rmi.server.hostname="10.1.1.1"

Problem: “Cannot find the file..." error message when trying to view online Help (Windows platform)

For Windows platforms, when using Netscape as your default browser, the following error message may result: “Cannot find the file '<filename>.html' (or one of its components). Make sure the path and filename are correct and that all required libraries are available."

The problem is due to an incorrect setting for HTML file association. The solution is the following:

1. Click **My Computer**, click **Tools**, select **Folder Options**, and click **File Types** tab
2. Select “Netscape Hypertext Document"
3. Click **Advanced** button, select **open**, click **Edit** button
4. Enter *NSShell* in the **Application:** field (not the Application Used to Perform Action: field), and click **OK**

Problem: Graphical user interface (GUI) does not start correctly

The graphical user interface (GUI), which is lbadmin, requires a sufficient amount of paging space to function correctly. If insufficient paging space is available, the GUI might not start up completely. If this occurs, check your paging space and increase it if necessary.

Problem: Error running Dispatcher with Caching Proxy installed

If you uninstall Load Balancer to reinstall another version and get an error when you attempt to start the Dispatcher component, check to see if Caching Proxy is installed. Caching Proxy has a dependency on one of the Dispatcher files; this file will uninstall only when Caching Proxy is uninstalled.

To avoid this problem:

1. Uninstall Caching Proxy.
2. Uninstall Load Balancer.
3. Reinstall both Load Balancer and Caching Proxy.

Problem: Graphical user interface (GUI) does not display correctly

If you experience a problem with the appearance of the Load Balancer GUI, check the setting for the operating system's desktop resolution. The GUI is best viewed at a resolution of 1024x768 pixels.

Problem: On Windows platform, help windows sometimes disappear behind other open windows

On Windows platform, when you first open help windows, they sometimes disappear into the background behind existing windows. If this occurs, click on the window to bring it forward again.

Problem: Load Balancer cannot process and forward a frame

On Solaris each network adapter has the same MAC address by default. This works properly when each adapter is on a different IP subnet; however, in a

switched environment, when multiple NICs with the same MAC and the same IP subnet address communicate with the same switch, the switch sends all traffic bound for the single MAC (and both IPs) down the same wire. Only the adapter that last put a frame on the wire sees the IP packets bound for both adapters. Solaris might discard packets for a valid IP address that arrived on the "wrong" interface.

If all network interfaces are not designated for Load Balancer as configured in ibmlb.conf, and if the NIC that is not defined in ibmlb.conf receives a frame, Load Balancer does not have the ability to process and forward the frame.

To avoid this problem, you must override the default and set a unique MAC address for each interface. Use this command:

```
ifconfig interface ether macAddr
```

For example:

```
ifconfig eri0 ether 01:02:03:04:05:06
```

Problem: A blue screen displays when you start the Load Balancer executor

On Windows platform, you must have a network card installed and configured before starting the executor.

Problem: Path to Discovery prevents return traffic with Load Balancer

The AIX operating system contains a networking parameter called path MTU discovery. During a transaction with a client, if the operating system determines that it must use a smaller maximum transmission unit (MTU) for the outgoing packets, path MTU discovery has AIX create a route to remember that data. The new route is for that specific client IP and records the necessary MTU to reach it.

When the route is being created, a problem might occur on the servers resulting from the cluster being aliased on the loopback. If the gateway address for the route falls in the subnet of the cluster/netmask, AIX systems create the route on the loopback. This happens because that was the last interface aliased with that subnet.

For example, if the cluster is 9.37.54.69 and a 255.255.255.0 netmask is used, and the intended gateway is 9.37.54.1, AIX systems use the loopback for the route. This causes the server's responses to never leave the machine, and the client times out waiting. The client typically sees one response from the cluster, then the route is created and the client receives nothing more.

To address this problem, enter the following command:

```
/usr/sbin/no -p -o udp_pmtu_discover=0
/usr/sbin/no -p -o tcp_pmtu_discover=0
```

This command will make the values persistent, and the values will apply to both current and future reboot values.

Problem: High availability in the Wide Area mode of Load Balancer does not work

When you set up a Wide Area Load Balancer, you must define the remote Dispatcher as a server in a cluster on your local Dispatcher. Typically, you use the non-forwarding address (NFA) of the remote Dispatcher as the destination address of the remote server. If you do this, and then set up high availability on the remote Dispatcher, it will fail. This happens because the local Dispatcher always points to the primary on the remote side when you use its NFA to access it.

To get around this problem:

1. Define an additional cluster on the remote Dispatcher. It is not necessary to define ports or servers for this cluster.
2. Add this cluster address to your goActive and goStandy scripts.
3. On your local Dispatcher, define this cluster address as a server, instead of the NFA of the remote primary Dispatcher.

When the remote primary Dispatcher comes up, it will alias this address on its adapter, allowing it to accept traffic. If a failure occurs, the address moves to the backup machine and the backup continues to accept traffic for that address.

Problem: GUI hangs (or unexpected behavior) when trying to load a large configuration file

When using lbadmin or Web administration (lbwebaccess) to load a large configuration file (roughly 200 or more **add** commands), the GUI may hang or display unexpected behavior, such as responding to screen changes at an extremely slow rate of speed.

This occurs because Java does not have access to enough memory to handle such a large configuration.

There is an option on the runtime environment that can be specified to increase the memory allocation pool available to Java.

The option is -Xmx*n* where *n* is the maximum size, in bytes, for the memory allocation pool. n must be a multiple of 1024 and must be greater than 2MB. The value *n* may be followed by k or K to indicate kilobytes, or m or M to indicate megabytes. For example, `-Xmx128M` and `-Xmx81920k` are both valid. The default value is 64M.

For example, to add this option, edit the lbadmin script file, modifying "javaw" to "javaw -Xmx*n*" as follows. (For AIX systems, modify "java" to "java -Xmx*n*"):

- **AIX systems**

```
javaw -Xmx256m -cp $LB_CLASSPATH $LB_INSTALL_PATH $LB_CLIENT_KEYS
com.ibm.internet.nd.framework.FWK_Main 1>/dev/null 2>&1 &
```

- **HP-UX systems**

```
java -Xmx256m -cp $LB_CLASSPATH $LB_INSTALL_PATH $LB_CLIENT_KEYS
com.ibm.internet.nd.framework.FWK_Main 1>/dev/null 2>&1 &
```

- **Linux systems**

```
javaw -Xmx256m -cp $LB_CLASSPATH $LB_INSTALL_PATH $LB_CLIENT_KEYS
com.ibm.internet.nd.framework.FWK_Main 1>/dev/null 2>&1 &
```

- **Solaris systems**

```
java -Xmx256m -cp $LB_CLASSPATH $LB_INSTALL_PATH $LB_CLIENT_KEYS
com.ibm.internet.nd.framework.FWK_Main 1>/dev/null 2>&1 &
```

- **Windows systems**

```
START javaw -Xmx256m -cp %LB_CLASSPATH% %LB_INSTALL_PATH%
 %LB_CLIENT_KEYS% com.ibm.internet.nd.framework.FWK_Main
```

There is no recommended value for *n* , but it should be greater than the default option. A good place to start would be with twice the default value.

Problem: lbadmin disconnects from server after updating configuration

If Load Balancer administration (lbadmin) disconnects from the server after you update the configuration, check the version of dsserver on the server that you are attempting to configure, and ensure that it is the same as your version of lbadmin or dscontrol.

Problem: IP addresses not resolving correctly over the remote connection

When using a remote client over a secure socks implementation, fully qualified domain names or host names might not resolve to the correct IP address in IP address format notation The socks implementation might add specific, socks-related data to the DNS resolution.

If the IP addresses are not resolving correctly over the remote connection, specify the IP address in the IP address notation format.

Problem: Korean Load Balancer interface displays overlapping or undesirable fonts on AIX and Linux systems

To correct overlapping or undesirable fonts in the Korean Load Balancer interface:

On AIX systems

1. Stop all Java processes on the AIX system.
2. Open the font.properties.ko file in an editor. This file is located in *home*/jre/lib where *home* is the Java home.
3. Search for this string:

```
-Monotype-TimesNewRomanWT-medium-r-normal
--*-%d-75-75-*-*-ksc5601.1987-0
```

4. Replace all instances of the string with:

```
-Monotype-SansMonoWT-medium-r-normal
--*-%d-75-75-*-*-ksc5601.1987-0
```

5. Save the file.

On Linux systems

1. Stop all Java processes on the system.
2. Open the font.properties.ko file in an editor. This file is located in *home*/jre/lib where *home* is the Java home.
3. Search for this string (with no spaces):

```
-monotype-
timesnewromanwt-medium-r-normal--*-%d-75-75-p-*-microsoft-symbol
```

4. Replace all instances of the string with:

```
-monotype-sansmonowt-medium-r-normal--*-%d-75-75-p-*-microsoft-symbol
```

5. Save the file.

Problem: On Windows systems, alias address is returned instead of local address when issuing commands such as hostname

On Windows systems, after aliasing the MS Loopback adapter, when issuing certain commands such as hostname, the OS will incorrectly respond with the alias address instead of the local address. To correct this problem, in the network connections list, the newly added alias must be listed below the local address. This will ensure that the local address is accessed prior to the loopback alias.

To check the network connections list:

1. Click **Start** > **Settings** > **Network and Dial-up Connections**
2. From the **Advanced** menu option, select **Advanced Settings...**
3. Ensure the **Local Area Connection** is listed first in the **Connections** box
4. If necessary, use the ordering buttons on the right to move entries up or down in the list

Problem: On Windows platform, unexpected GUI behavior when using Matrox AGP video cards

On Windows platform when using a Matrox AGP card, unexpected behavior can occur in the Load Balancer GUI. When clicking the mouse, a block of space slightly larger than the mouse pointer can become corrupted causing possible highlighting reversal or images to shift out of place on the screen. Older Matrox cards have not shown this behavior. There is no known fix when using Matrox AGP cards.

Problem: Unexpected behavior when executing "rmmod ibmlb" (Linux systems)

On Linux systems, if dsserver is still running during the manual removal of the Load Balancer kernel module, unexpected behavior, such as system hang or javacores, can occur. When manually removing the Load Balancer kernel module, you must first stop dsserver.

If "dsserver stop" does not work, stop the java process with SRV_KNDConfigServer. Stop the process by finding its process identifier using `ps -ef | grep SRV_KNDConfigServer` command and then ending the process using `kill process_id` command.

You can safely run the "rmmod ibmlb" command to remove the Load Balancer module from the kernel.

Problem: Slow response time running commands on Dispatcher machine

If you are running the Dispatcher component for load balancing, it is possible to overload the computer with client traffic. The Load Balancer kernel module has the highest priority, and if it is constantly handling client packets, the rest of the system may become unresponsive. Running commands in user space may take a very long time to complete, or may never complete.

If this happens, you should begin to restructure your setup to avoid overloading the Load Balancer machine with traffic. Alternatives include spreading the load across several Load Balancer machines, or replacing the machine with a stronger and faster computer.

When trying to decide if the slow response time on the machine is due to high client traffic, consider whether this occurs during client peak traffic times. Misconfigured systems that cause routing loops can also cause the same symptoms. But before changing the Load Balancer setup, determine whether the symptoms may be due to high client load.

Problem: SSL or HTTPS advisor not registering server loads (when using mac-forwarding)

When using mac-based forwarding method, Load Balancer will send packets to the servers using the cluster address which is aliased on the loopback. Some server applications (such as SSL) require that configuration information (such as certificates) are based on the IP address. The IP address must be the cluster address which is configured on the loopback in order to match the contents of the incoming packets. If the IP address of the cluster is not used when configuring the server application, then the client request will not get properly forwarded to the server.

Problem: Disconnect from host occurs when resize Netscape browser window while using Web administration

If you are using remote Web administration to configure Load Balancer, do not resize (Minimize, Maximize, Restore Down, and so on) the Netscape browser window in which the Load Balancer GUI appears. Because Netscape reloads a page every time browser windows are resized, this will cause a disconnect from host. You will need to reconnect to host each time you resize the window. If you are performing remote Web administration on a Windows platform, use Internet Explorer.

Problem: On Windows systems, corrupted Latin-1 national characters appear in command prompt window

In a command prompt window on the Windows operating system, some national characters of the Latin-1 family might appear corrupted. For example, the letter "a" with a tilde may display as a pi symbol. To fix this, you must change the font properties of the command prompt window. To change the font, do the following:

1. Click the icon in the upper left corner of the command prompt window
2. Select Properties, then click the Font tab
3. The default font is Raster fonts; change this to Lucida Console and click OK

Problem: On HP-UX, Java out of memory or thread error occurs

Some HP-UX 11i installations are pre-configured to allow only 64 threads per process. However, some Load Balancer configurations require more than this amount. For HP-UX systems, set the threads per process to at least 256. To increase this value, use the "sam" utility to set the max_thread_proc kernel parameter. If heavy use is expected, you might need to increase max_thread_proc beyond 256.

To increase the max_thread_proc parameter, do the following:

1. From the command line, type: sam
2. Select **Kernel Configuration > Configurable Parameters**
3. From the scroll bar, select **max_thread_proc**
4. Press Spacebar to highlight **max_thread_proc**
5. Press Tab one time, then press the right-arrow key until you select **Actions**

6. Press Enter to display the **Actions** menu, then press **M** to select Modify Configurable Parameter. (If you do not see this option, highlight **max_thread_proc**)
7. Press Tab until you select the **Formula/Value** field
8. Type a value of 256 or greater.
9. Click **OK**
10. Press Tab one time, then select **Actions**
11. Press **K** for Process New Kernel..
12. Select **Yes**
13. Reboot your system

Problem: On Windows systems, advisors and reach targets mark all servers down

When configuring your adapter on a Load Balancer machine, you must ensure that the following two settings are correct for the advisor to work:

- Disable Task Offloading, which is most commonly used on 3Com adapter cards.
  - To disable Task offloading: Go to Start > Settings > Control Panel > Network and Dial-up Connections, then select the adapter.
  - In the pop-up window, click Properties.
  - Click Configure, then select the Advanced tab.
  - In the property pane, select the Task Offload property, then select disable in the value field.
- Enable Protocol 1 (ICMP) for IP protocols if you are enabling TCP/IP filtering. If ICMP is not enabled, the ping test to the backend server will not succeed. To check whether ICMP is enabled:
  - Go to Start > Settings > Control Panel > Network and Dial-up Connections, then select the adapter.
  - In the pop-up window, click Properties.
  - From the components pane, select Internet Protocol (TCP/IP), then click Properties.
  - Click Advanced, then select the Options tab.
  - Select TCP/IP filtering in the options pane, then click Properties.
  - If you have selected **Enable TCP/IP Filtering** and **permit only** for IP protocols, you must add IP Protocol 1. This must be added in addition to the existing TCP and UDP ports that you enabled.

Problem: On Windows platform, resolving IP address to host name when more than one address is configured to an adapter

On Windows platform, when configuring an adapter with more than one IP address, configure the IP address that you want affiliated to the host name first in the registry.

Because Load Balancer is dependent on InetAddress.getLocalHost() in many instances (for example, lbkeys create), multiple IP addresses aliased to a single adapter might cause problems. To avoid this problem, list the IP address to which you want your host name to resolve first in the registry. For example:

1. Start Regedit
2. Modify the following value names as follows:

- HKEY_LOCAL_MACHINE -> SYSTEM -> ControlSet001 -> Services -> *YourInterfaceAddress*} -> Parameters -> Tcpip-> IPAddress
  - Place the IP address to which you want the host name to resolve first.
- HKEY_LOCAL_MACHINE -> SYSTEM -> ControlSet001 -> Services -> Tcpip -> Parameters -> Interfaces -> *YourInterfaceAddress* -> IPAddress
  - Place the IP address to which you want the host name to resolve first.
- HKEY_LOCAL_MACHINE -> SYSTEM -> ControlSet002 -> Services -> *YourInterfaceAddress* -> Parameters -> Tcpip -> IPAddress
  - Place the IP address to which you want the host name to resolve first.
- HKEY_LOCAL_MACHINE -> SYSTEM -> ControlSet002 -> Services -> Tcpip -> Parameters -> Interfaces -> *YourInterfaceAddress* -> IPAddress
  - Place the IP address to which you want the host name to resolve first.
- HKEY_LOCAL_MACHINE -> SYSTEM -> CurrentControlSet -> Services-> *YourInterfaceAddress* -> Parameters -> Tcpip- > IPAddress
  - Place the IP address to which you want the host name to resolve first.
- HKEY_LOCAL_MACHINE -> SYSTEM -> CurrentControlSet -> Services-> Tcpip -> Parameters -> Interfaces -> *YourInterfaceAddress* -> IPAddress
  - Place the IP address to which you want the host name to resolve first.

3. Reboot
4. Check that your host name resolves to the correct IP address. For example, ping *yourhostname*.

Problem: On Windows systems, after network outage, advisors not working in a high availability setup

By default, when the Windows operating system detects a network outage, it clears its address resolution protocol (ARP) cache, including all static entries. After the network is available, the ARP cache is repopulated by ARP requests sent on the network.

With a high availability configuration, both servers take over primary operations when a loss of network connectivity affects one or both. When the ARP request is sent to repopulate the ARP cache, both servers respond, which causes the ARP cache to mark the entry as not valid. Therefore, the advisors are not able to create a socket to the backup servers.

Preventing the Windows operating system from clearing the ARP cache when there is a loss of connectivity solves this problem. Microsoft has published an article that explains how to accomplish this task. This article is on the Microsoft Web site, located in the Microsoft Knowledge Base, article number 239924: http://support.microsoft.com/default.aspx?scid=kb;en-us;239924.

The following is a summary of the steps, described in the Microsoft article, to prevent the system from clearing the ARP cache:

1. Use the Registry editor (regedit or regedit32) to open the registry.
2. View the following key in the registry:
   `HKEY_LOCAL_MACHINE\System\CurrentControlSet\Services\Tcpip\Parameters`
3. Add the following registry value: Value Name: DisableDHCPMediaSense Value Type: REG_DWORD.
4. After the key is added, edit the value and set it to 1.
5. Reboot the machine for the change to take effect.

Note: This affects the ARP cache regardless of the DHCP setting.

Problem: On Linux systems, do not use "IP address add" command when aliasing multiple clusters on the loopback device

Certain considerations must be taken when using Linux kernel 2.4.x servers and Dispatcher's MAC forwarding method. If the server has a cluster address configured on the loopback device using the **ip address add** command, only one cluster address can be aliased.

When aliasing multiple clusters to the loopback device use the **ifconfig** command, for example:

```
ifconfig lo:num clusterAddress netmask 255.255.255.255 up
```

Additionally, there are incompatibilities between the ifconfig method of configuring interfaces and the ip method of configuring interfaces. Best practice suggests that a site choose one method and use that method exclusively.

Problem: "Router address not specified or not valid for port method" error message

When adding servers to your Dispatcher configuration, the following error message can result: "Error: Router address not specified or not valid for port method".

Use this checklist to determine the problem:

- Ensure you have applied the latest maintenance level.
- Ensure you are using an IBM distribution of Java (except on the Solaris platforms).
- Ensure you are not configured to use DHCP on Windows systems.
- If the forwarding method is mac (default), then the server, cluster and at least one supported NIC must be on the same subnet. For example, you cannot define a cluster of 10.1.1.1 and a server of 130.2.3.4 because they are not on the same subnet.

  Note: If the forwarding method is nat or cbr, the servers do not need to be on the same subnet as the cluster.
- If all are on the same subnet, and you have aliased the cluster, ensure you alias the cluster on a NIC that routes to this subnet. For example, if en0 is defined for 13.2.3.4 and en1 is defined for 9.1.2.3 and the cluster definition is 9.5.7.3, you must configure the cluster on en1. The default interface is en0.
- On Linux platforms, ensure you have loaded the correct kernel by looking in the /usr/lpp/ibm/internet/nd/logs/dispatcher directory for the loadoutput.log file. Check this file for any reported errors.

The default for the router parameter is 0, which indicates the server is local. When you set the server's router address to something other than 0, this indicates that it is a remote server, on a different subnet. For more information on the router parameter on the server add command, see “dscontrol server — configure servers” on page 330.

If the server that you are adding is located on a different subnet, the router parameter should be the address of the router to be used on the local subnet to communicate with the remote server.

Problem: On Solaris systems, Load Balancer processes end when you exit the terminal window from which they started

On Solaris systems, after starting Load Balancer scripts (such as dsserver or lbadmin) from a terminal window, if you exit from that window, the Load Balancer process also exits.

To resolve this problem, start the Load Balancer scripts with the **nohup** command. For example: **nohup dsserver**. This command prevents the processes started from the terminal session from receiving a hangup signal from the terminal when it exits, allowing the processes to continue even after the terminal session has ended. Use the **nohup** command in front of any Load Balancer scripts that you want to continue to process beyond the end of a terminal session.

Problem: Delay occurs while loading a Load Balancer configuration

Loading a Load Balancer configuration might take a long time due to Domain Name System (DNS) calls that are made to resolve and verify the server address.

If the DNS of the Load Balancer machine is configured incorrectly, or if DNS in general takes a long time, this will cause a slow down in loading the configuration due to the Java processes that are sending DNS requests on the network.

A workaround for this is to add your server addresses and hostnames to your local `/etc/hosts` file.

Problem: On Windows systems, an IP address conflict error message appears

If high availability is configured, the cluster addresses may be configured on both machines for a brief period and cause the following error message to occur: There is an IP address conflict with another system on the network. In this case, you can safely ignore the message. It is possible for a cluster address to be briefly configured on both high availability machines at the same time, especially during startup of either machine, or when a takeover has been initiated.

Check the go* scripts to ensure they are correctly configuring and unconfiguring cluster addresses. If you have invoked a configuration file and have go* scripts installed, ensure you do not have any "executor configure" command statements for your cluster addresses in your configuration file, as this will conflict with the configure and unconfigure commands in the go* scripts.

For more information on go* scripts when configuring high availability, see “Using scripts” on page 167.

Problem: Both primary and backup machines are active in a high availability configuration

This problem may occur when the go scripts do not run on either primary or backup machine. The go scripts cannot run if dsserver is not started on both machines. Check both machines and make sure dsserver is running.

Problem: Client requests fail when attempting the return of large page responses

Client requests that result in large page responses time out if the maximum transmit unit (MTU) is not set properly on the Dispatcher machine. For Dispatcher component's cbr and nat forwarding methods, this can occur because Dispatcher defaults the MTU value, rather than negotiate the value.

The MTU is set on each operating system based on the type of communication media (for example, Ethernet or Token-Ring). Routers from the local segment might have a smaller MTU set if they connect to a different type of communication media. Under normal TCP traffic, an MTU negotiation occurs during the connection setup, and the smallest MTU is used to send data between the machines.

Dispatcher does not support MTU negotiation for Dispatcher's cbr or nat forwarding method because it is actively involved as an endpoint for TCP connections. For cbr and nat forwarding, Dispatcher defaults the MTU value to `1500`. This value is the typical MTU size for standard Ethernet, so most customers do not need to adjust this setting.

When using Dispatcher's cbr or nat forwarding method, if you have a router to the local segment that has a lower MTU, you must set the MTU on the Dispatcher machine to match the lower MTU.

To resolve this problem, use the following command to set the maximum segment size (mss) value: `dscontrol executor set mss` *`new_value`*

For example:

```
dscontrol executor set mss 1400
```

The default for mss is `1460`.

The mss setting does not apply for Dispatcher's mac forwarding method or any non-Dispatcher component of Load Balancer.

Problem: On Windows systems, "Server not responding" error occurs when issuing dscontrol or lbadmin

When more than one IP address is on a Windows system and the **hosts** file does not specify the address to associate with the host name, the operating system chooses the smallest address to associate with the host name.

To resolve this problem, update the `c:\Windows\system32\drivers\etc\`**`hosts`** file with your machine host name and the IP address that you want to associate with the host name.

IMPORTANT: The IP address cannot be a cluster address.

Problem: High availability Dispatcher machines may fail to synchronize on Linux for S/390 systems on qeth drivers

When using high availability on Linux for S/390 machines with the qeth network driver, the active and standby Dispatchers may fail to synchronize. This problem might be limited to Linux Kernel 2.6.

If this problem occurs, use the following workaround:

Define a channel-to-channel (CTC) network device between the active and standby Dispatcher images and add a heartbeat between the two CTC endpoint IP addresses.

Problem: Tips on configuring high availability

With the high availability function for Load Balancer, a partner machine can takeover load balancing if the primary partner fails or is shut down. To maintain connections between the high availability partners, connection records are passed between the two machines. When the backup partner takes over the load balancing function, the cluster IP address is removed from the backup machine and added to the new primary machine. There are numerous timing and configuration considerations that can affect this takeover operation.

The tips listed in this section can help alleviate problems that arise from high availability configuration problems such as:

- Connections dropped after takeover
- Partner machines unable to synchronize
- Requests erroneously directed to the backup partner machine

The following tips are helpful for successful configuration of high availability on your Load Balancer machines.

- The positioning of the high availability commands in your script files can make a significant difference.

  Examples of high availability commands are:

  ```
  dscontrol highavailability heartbeat add ...
  dscontrol highavailability backup add ...
  dscontrol highavailability reach add ...
  ```

  In most cases, you must position the high availability definitions at the end of the file. The cluster, port, and server statements must be placed before the high availability statements. This is because when high availability synchronizes, it looks for the cluster, port, and server definitions when a connection record is received.

  If the cluster, port, and server do not exist, the connection record is dropped. If a takeover occurs and the connection record has not been replicated on the partner machine, the connection fails.

  The exception to this rule is when using collocated servers that are configured with the MAC-forwarding method. In this case, the high availability statements must come before the collocated server statements. If the high availability statements are not before the collocated server statements, Load Balancer receives a request for the collocated server, but it appears the same as an incoming request for the cluster and is load balanced. This can lead to a looping of the packets on the network and lead to excess traffic. When the high availability statements are placed before the collocated server, Load Balancer knows that it should not forward incoming traffic unless it is in the ACTIVE state.
- On z/OS or OS/390 operating systems, the hypervisor controls the interface and multiplexes the real interface among the guest operating systems. The hypervisor permits only one guest at a time to register itself for an IP address, and there is an update window. This means that when the cluster IP is removed

from the backup machine, you might have to add a delay before trying to add the cluster IP to the primary machine; otherwise, it fails and incoming connections are not processed.

To correct this behavior, add a sleep delay in the goActive script. The amount of time needed to sleep is deployment dependent. It is recommended that you start with a sleep delay time of 10.

- High availability partners must be able to ping each other and must be on the same subnet.

  By default, the machines attempt to communicate with each other every one half second and will detect a failure after four failed attempts. If you have a busy machine, this might cause failovers to occur when the system is still functioning properly. You can increase the number of times until failure by issuing:

  ```
  dscontrol executor set hatimeout <value>
  ```

- When the partners synchronize, all the connection records are sent from the active machine to the backup machine. The synchronization must complete within the default limit of 50 seconds.

  To accomplish this, old connections must not remain in memory for an extended amount of time. In particular, there have been issues with LDAP ports and large `staletimeout` periods (in excess of one day). Setting a large `staletimeout` period causes old connections to remain in memory, which causes more connection records to be passed at synchronization, and also more memory usage on both machines.

  If the synchronization fails with a reasonable `staletimeout` period, you can increase the synchronization timeout by issuing:

  ```
  e xm 33 5 new_timeout
  ```

  This command is not stored in the configuration file when it is saved, so you must manually add it to the configuration file if you want this setting to persist between shutdowns.

  The timeout value is stored in one half seconds; therefore, the default value for `new_timeout` is 100 (50 seconds).

- When a partner machine takes over the workload, it issues a gratuitous ARP response to tell machines on the same subnet of the new hardware address associated with the cluster IP address. You must ensure that your routers honor gratuitous ARPs and update their cache, or the requests will be sent to the inactive partner.

Note: For information on configuring the high availability feature see "High availability" on page 164.

Problem: On Linux, Dispatcher configuration limitations when using zSeries or S/390 servers that have Open System Adapter (OSA) cards

In general, when using the MAC forwarding method, servers in the Load Balancer configuration must all be on the same network segment regardless of the platform. Active network devices such as router, bridges, and firewalls interfere with Load Balancer. This is because Load Balancer functions as a specialized router, modifying only the link-layer headers to its next and final hop. Any network topology in which the next hop is not the final hop is not valid for Load Balancer.

Note: Tunnels, such as channel-to-channel (CTC) or inter-user communication vehicle (IUCV), are often supported. However, Load Balancer must forward across the tunnel directly to the final destination, it cannot be a network-to-network tunnel.

There is a limitation for zSeries and S/390 servers that share the OSA card, because this adapter operates differently than most network cards. The OSA card has its own virtual link layer implementation, which has nothing to do with ethernet, that is presented to the Linux and z/OS hosts behind it. Effectively, each OSA card looks just like ethernet-to-ethernet hosts (and not to the OSA hosts), and hosts that use it will respond to it as if it is ethernet.

The OSA card also performs some functions that relate to the IP layer directly. Responding to ARP (address resolution protocol) requests is one example of a function that it performs. Another is that shared OSA routes IP packets based on destination IP address, instead of on ethernet address as a layer 2 switch. Effectively, the OSA card is a bridged network segment unto itself.

Load Balancer that runs on an S/390 Linux or zSeries Linux host can forward to hosts on the same OSA or to hosts on the ethernet. All the hosts on the same shared OSA are effectively on the same segment.

Load Balancer can *forward out* of a shared OSA because of the nature of the OSA bridge. The bridge knows the OSA port that owns the cluster IP. The bridge knows the MAC address of hosts directly connected to the ethernet segment. Therefore, Load Balancer can MAC-forward across one OSA bridge.

However, Load Balancer cannot forward into a shared OSA. This includes the Load Balancer on an S/390 Linux when the backend server is on a different OSA card than the Load Balancer. The OSA for the backend server advertises the OSA MAC address for the server IP, but when a packet arrives with the ethernet destination address of the server's OSA and the IP of the cluster, the server's OSA card does not know which of its hosts, if any, should receive that packet. The same principles that permit OSA-to-ethernet MAC-forwarding to work out of one shared OSA do not hold when trying to forward into a shared OSA.

Workaround:

In Load Balancer configurations that use zSeries or S/390 servers that have OSA cards, there are two approaches you can take to work around the problem that has been described.

1. Using platform features

   If the servers in the Load Balancer configuration are on the same zSeries or S/390 platform type, you can define point-to-point (CTC or IUCV) connections between Load Balancer and each server. Set up the endpoints with private IP addresses. The point-to-point connection is used for Load Balancer-to-server traffic only. Then add the servers with the IP address of the server endpoint of the tunnel. With this configuration, the cluster traffic comes through the Load Balancer OSA card and is forwarded across the point-to-point connection where the server responds through its own default route. The response uses the server's OSA card to leave, which might or might not be the same card.
2. Using Load Balancer's GRE feature

   If the servers in the Load Balancer configuration are not on the same zSeries or S/390 platform type, or if it is not possible to define a point-to-point connection between Load Balancer and each server, it is recommended that you use Load

Balancer's Generic Routing Encapsulation (GRE) feature, which is a protocol that permits Load Balancer to forward across routers.

When using GRE, the client->cluster IP packet is received by Load Balancer, encapsulated, and sent to the server. At the server, the original client->cluster IP packet is excapsulated, and the server responds directly to the client. The advantage with using GRE is that Load Balancer sees only the client-to-server traffic, not the server-to-client traffic. The disadvantage is that it lowers the maximum segment size (MSS) of the TCP connection due to encapsulation overhead.

To configure Load Balancer to forward with GRE encapsulation, add the servers using the following command:

```
dscontrol server add cluster_add:port:backend_server router
backend_server
```

Where router backend_server is valid if Load Balancer and the backend server are on the same IP subnet. Otherwise, specify the valid next-hop IP address as the router.

To configure Linux systems to perform native GRE excapsulation, for each backend server, issue the following commands:

```
modprobe ip_gre
ip tunnel add grelb0 mode gre ikey 3735928559
ip link set grelb0 up
ip addr add cluster_addr dev grelb0
```

Note: Do not define the cluster address on the loopback of the backend servers. When using z/OS backend servers, you must use z/OS-specific commands to configure the servers to perform GRE excapsulation.

Problem: On some Linux versions, a memory leak occurs when running Dispatcher configured with the manager and advisors

When running Load Balancer configured with the manager and advisor features, large memory leaks can occur on some Red Hat Linux versions. The Java memory leak increases if you configure a small time-interval setting for the advisor.

The IBM Java SDK versions of the JVM and the Native POSIX Thread Library (NPTL) shipped with some Linux distributions, such as Red Hat Enterprise Linux 3.0, can cause the memory leak to occur. The enhanced threading library NPTL is shipped with some distributions of Linux systems, such as Red Hat Enterprise Linux 3.0, that support NPTL.

Refer to `http://www.ibm.com/developerworks/java/jdk/linux/tested.html` for the latest information on Linux systems and the IBM Java SDK shipped with these systems.

As a problem determination tool, use the `vmstat` or `ps` command to detect memory leaks.

To fix the memory leak, issue the following command before running the Load Balancer machine to disable the NPTL library:

```
export LD_ASSUME_KERNEL=2.4.10
```

Problem: On SUSE Linux Enterprise Server 9, Dispatcher forwards packets, but the packets do not reach the backend server

On Suse Linux Enterprise Server 9, when using the MAC forwarding method, the Dispatcher report might indicate that the packet was forwarded (packet count increases); however, the packet never reaches the backend server.

You might observe one or both of the following when this problem occurs:

- On the Dispatcher machine-side, the following message displays:

  ```
  ip_finish_output2: No header cache and no neighbour!
  ```

- On the client-side, the following message displays:

  ```
  ICMP Destination unreachable: Fragmentation Needed
  ```

This problem might occur due to the iptables NAT module that is loaded. On SLES 9, there is a possible, but unconfirmed, error in this version of iptables that causes strange behavior when interacting with Dispatcher.

Solution:

Unload the iptables NAT module and Connection Tracking module.

For example:

```
# lsmod | grep ip
        iptable_filter          3072  0
        iptable_nat            22060  0
        ip_conntrack           32560  1 iptable_nat
        ip_tables              17280  2
iptable_filter,iptable_nat
        ipv6                  236800  19
        # rmmod iptable_nat
        # rmmod ip_conntrack
```

Remove the modules in the order of their usage. Specifically, you can remove a module only if the reference count (last column in lsmod output) is zero. If you have configured any rules in iptables, you must remove them. For example: `iptables -t nat -F`.

The iptable_nat module uses ip_conntrack, so you must first remove iptable_nat module, and then remove ip_conntrack module.

Note: Just trying to list rules configured on a table loads up the corresponding module; for example: `iptables -t nat -L`. Make sure that you do not run this after the modules are removed.

Problem: On Windows system, IP address conflict message appears during high availability takeover

On Windows systems, if you are running Load Balancer's high availability feature, goScripts are used to configure the cluster IP on the active Load Balancer and to unconfigure the cluster IP on the backup system when a takeover occurs. If the goScript that configures the cluster IP address on the active machine runs before the goScript to unconfigure the IP cluster address on the backup machine, problems might occur. You might see a popup window that tells you that the system has detected an IP address conflict. If you run the `ipconfig \all` command, you might also see that there is a 0.0.0.0 IP address on the machine.

Solution:

Issue the following command to manually unconfigure the cluster IP address from the primary machine:

```
dscontrol executor unconfigure clusterIP
```

This removes the 0.0.0.0 address from the Windows IP stack.

After the high availability partner releases the cluster IP address, issue the following command to manually add the cluster IP back:

```
dscontrol executor configure clusterIP
```

After this command is issued, look for the cluster IP address on the Windows IP stack again by issuing the following command:

```
ipconfig /all
```

Problem: Linux iptables can interfere with the routing of packets

Linux iptables can interfere with load balancing of traffic and must be disabled on the Dispatcher machine.

Issue the following command to determine if iptables are loaded:

```
lsmod | grep ip_tables
```

The output from the preceding command might be similar to this:

```
ip_tables         22400   3
iptable_mangle,iptable_nat,iptable_filter
```

Issue the following command for each iptable listed in the output to display the rules for the tables:

```
iptables -t <short_name> -L
```

For example:

```
iptables -t mangle -L
iptables -t nat    -L
iptables -t filter -L
```

If iptable_nat is loaded, it must be unloaded. Because iptable_nat has a dependency on iptable_conntrack, iptable_conntrack also must be removed. Issue the following command to unload these two iptables:

```
rmmod iptable_nat iptable_conntrack
```

Upgrading the Java file set provided with the Load Balancer installation

During the Load Balancer installation process, a Java file set also gets installed. Load Balancer will be the only application that uses the Java version which installs with the product. You should not upgrade this version of the Java file set on your own. If there are problem which requires an upgrade for the Java file set, you should report the problem to IBM Service so the Java file set which is shipped within Load Balancer will be upgraded with an official fix level.

Problem: Persistent connections might drop during high availability takeover

On Microsoft Windows operating systems, persistent connections might drop during a high availability takeover. This problem exists only when you have a collocated server that uses the MAC forwarding method.

When the cluster IP address is deleted, either from the ethernet interface or the loopback interface, any connections on that IP address are released. When the operating system receives a packet on a connection that has been released, it sends a RST response back to the client and the connection is terminated.

If you cannot tolerate connections being dropped during a high availability takeover, you must not use a collocated server on Windows operating systems when you use the MAC forwarding method.

Solving common problems—CBR

Problem: CBR will not run

This problem can occur when another application is using one of the ports used by CBR. For more information, go to "Checking CBR port numbers" on page 246.

Problem: cbrcontrol or lbadmin command fails

1. The cbrcontrol command returns: **Error: Server not responding**. Or, the lbadmin command returns: **Error: unable to access RMI server**. These errors can result when your machine has a socksified stack. To correct this problem, edit the socks.cnf file to contain the following lines:

   ```
   EXCLUDE-MODULE java
   EXCLUDE-MODULE javaw
   ```

2. The administration consoles for Load Balancer interfaces (command line, graphical user interface, and wizards) communicate with cbrserver using remote method invocation (RMI). The default communication uses three ports; each port is set in the cbrserver start script:
   - 11099 to receive commands from cbrcontrol
   - 10004 to send metric queries to Metric Server
   - 11199 for the RMI server port

   This can cause problems when one of the administration consoles runs on the same machine as a firewall or through a firewall. For example, when Load Balancer runs on the same machine as a firewall, and you issue cbrcontrol commands, you might see errors such as **Error: Server not responding**.

   To avoid this problem, edit the cbrserver script file to set the port used by RMI for the firewall (or other application). Change the line: LB_RMISERVERPORT=11199 to LB_RMISERVERPORT=*yourPort*. Where *yourPort* is a different port.

   When complete, restart cbrserver and open traffic for ports 11099, 10004, 11199, and 11100, or for the chosen port for the host address from which the administration console will be run.
3. These errors can also occur if you have not already started **cbrserver**.

Problem: Requests not being load balanced

Caching Proxy and CBR have been started, but requests are not being load balanced. This error can occur if you start Caching Proxy before starting the

executor. If this happens, the stderr log for Caching Proxy will contain the following error message: "ndServerInit: Could not attach to executor." To avoid this problem, start the executor before starting Caching Proxy.

Problem: On Solaris systems, cbrcontrol executor start command fails

On Solaris systems, the **cbrcontrol executor start** command returns: “Error: Executor was not started." This error occurs if you do not configure the IPC (Inter-process Communication) for the system so that the maximum size of a shared memory segment and semaphore IDs are bigger than the operating system's default. In order to increase the size of the shared memory segment and semaphore IDs, you must edit the **/etc/system** file. For more information on how to configure this file, see the section on modifying the system defaults for IPCs (Inter-process Communication).

Problem: Syntactical or configuration error

If the URL rule does not work, this can be a result of either a syntactical or configuration error. For this problem check the following:

- Verify the rule is configured correctly. See Appendix B, “Content rule (pattern) syntax,” on page 409, for details.
- Issue a **cbrcontrol rule report** for this rule, and check the ‘Times Fired' column to see if it has incremented according to the number of requests made. If it has incremented correctly, recheck the server configuration.
- If the rule is not being fired, add an ‘always true' rule. Issue a **cbrcontrol rule report** on the ‘always true' rule to verify that it is getting fired.

Problem: On Windows platform, unexpected GUI behavior when using Matrox AGP video cards

On Windows platform when using a Matrox AGP card, unexpected behavior can occur in the Load Balancer GUI. When clicking the mouse, a block of space slightly larger than the mouse pointer can become corrupted causing possible highlighting reversal or images to shift out of place on the screen. Older Matrox cards have not shown this behavior. There is no known fix when using Matrox AGP cards.

Problem: Disconnect from host occurs when resize Netscape browser window while using Web administration

If you are using remote Web administration to configure Load Balancer, do not resize (Minimize, Maximize, Restore Down, and so on) the Netscape browser window in which the Load Balancer GUI appears. Because Netscape reloads a page every time browser windows are resized, this will cause a disconnect from host. You will need to reconnect to host each time you resize the window. If you are performing remote Web administration on a Windows platform, use Internet Explorer.

Problem: On Windows platform, corrupted Latin-1 national characters appear in command prompt window

In a command prompt window on the Windows operating system, some national characters of the Latin-1 family might appear corrupted. For example, the letter "a" with a tilde may display as a pi symbol. To fix this, you must change the font properties of the command prompt window. To change the font, do the following:

1. Click the icon in the upper left corner of the command prompt window

2. Select Properties, then click the Font tab
3. The default font is Raster fonts; change this to Lucida Console and click OK

Problem: On HP-UX, Java out of memory/ thread error occurs

Some HP-UX 11i installations are pre-configured to allow only 64 threads per process. However, some Load Balancer configurations require more than this amount. For HP-UX systems, set the threads per process to at least 256. To increase this value, use the "sam" utility to set the max_thread_proc kernel parameter. If heavy use is expected, you might need to increase max_thread_proc beyond 256.

To increase max_thread_proc, refer to the steps to increase the max_thread_proc parameter.

Problem: On Windows systems, advisors and reach targets mark all servers down

When configuring your adapter on a Load Balancer machine, you must ensure that the following two settings are correct for the advisor to work:

- Disable Task Offloading, which is most commonly used on 3Com adapter cards.
- Enable Protocol 1 (ICMP) for IP protocols if you are enabling TCP/IP filtering. If ICMP is not enabled, the ping test to the backend server will not succeed.

Refer to the section on disabling task offloading for instructions on configuring this setting.

Problem: On Windows systems, resolving IP address to host name when more than one address is configured to an adapter

On Windows platform, when configuring an adapter with more than one IP address, configure the IP address that you want affiliated to the host name first in the registry.

Because Load Balancer is dependent on InetAddress.getLocalHost() in many instances (for example, lbkeys create), multiple IP addresses aliased to a single adapter might cause problems. To avoid this problem, list the IP address to which you want your host name to resolve first in the registry.

To address this issue, reorder the adapters in the Advanced Settings for the Control Panel's Network Connections option. For example:

1. Open the Control Panel.
2. Open the Network Connections option.
3. From the menu bar, select **Advanced > Advanced Settings...**
4. Reorder the adapters that are listed in the Advanced Settings panel.

Solving common problems—Site Selector

Problem: Site Selector will not run

This problem can occur when another application is using one of the ports used by Site Selector. For more information, go to “Checking Site Selector port numbers” on page 247.

Problem: Site Selector does not round-robin traffic from Solaris clients

Symptom: Site Selector component does not round-robin incoming requests from Solaris clients.

Possible cause: Solaris systems run a name service cache daemon. If this daemon is running, the subsequent resolver request is answered from this cache instead of querying Site Selector.

Solution: Turn off the name service cache daemon on the Solaris machine.

Problem: sscontrol or lbadmin command fails

1. The sscontrol command returns: **Error: Server not responding**. Or, the lbadmin command returns: **Error: unable to access RMI server**. These errors can result when your machine has a socksified stack. To correct this problem, edit the socks.cnf file to contain the following lines:

   ```
   EXCLUDE-MODULE java
   EXCLUDE-MODULE javaw
   ```

2. The administration consoles for Load Balancer interfaces (command line, graphical user interface, and wizards) communicate with ssserver using remote method invocation (RMI). The default communication uses three ports; each port is set in the ssserver start script:
   - 12099 to receive commands from sscontrol
   - 10004 to send metric queries to Metric Server
   - 12199 for the RMI server port
   - 53 for sending and receiving DNS traffic

   This can cause problems when one of the administration consoles runs on the same machine as a firewall or through a firewall. For example, when Load Balancer runs on the same machine as a firewall, and you issue sscontrol commands, you might see errors such as **Error: Server not responding**.

   To avoid this problem, edit the ssserver script file to set the port used by RMI for the firewall (or other application). Change the line: LB_RMISERVERPORT=10199 to LB_RMISERVERPORT=*yourPort*. Where *yourPort* is a different port.

   When complete, restart ssserver and open traffic for ports 12099, 10004, 12199, and 12100, or for the chosen port for the host address from which the administration console will be run.
3. These errors can also occur if you have not already started **ssserver**.

Problem: The ssserver is failing to start on Windows platform

Site Selector must be able to participate in a DNS. All the machines involved in the configuration should also be participants of this system. Windows systems do not always require the configured host name to be in the DNS. Site Selector requires that its host name be define in the DNS to start properly.

Verify this host is defined in the DNS. Edit the ssserver.cmd file and remove the "w" from "javaw". This should provide more information about errors.

Problem: Site Selector with duplicate routes not load balancing correctly

Site Selector's name server does not bind to any one address on the machine. It will respond to requests destined for any valid IP on the machine. Site Selector relies on the operating system to route the response back to the client. If the Site Selector machine has multiple adapters and any number of them are attached to the same subnet, it is possible the O/S will send the response to the client from a different address than it was received. Some client applications will not accept a response received from an address other than where it was sent. As a result, the name resolution will appear to fail.

Problem: On Windows platform, unexpected GUI behavior when using Matrox AGP video cards

On Windows platform when using a Matrox AGP card, unexpected behavior can occur in the Load Balancer GUI. When clicking the mouse, a block of space slightly larger than the mouse pointer can become corrupted causing possible highlighting reversal or images to shift out of place on the screen. Older Matrox cards have not shown this behavior. There is no known fix when using Matrox AGP cards.

Problem: Disconnect from host occurs when resize Netscape browser window while using Web administration

If you are using remote Web administration to configure Load Balancer, do not resize (Minimize, Maximize, Restore Down, and so on) the Netscape browser window in which the Load Balancer GUI appears. Because Netscape reloads a page every time browser windows are resized, this will cause a disconnect from host. You will need to reconnect to host each time you resize the window. If you are performing remote Web administration on a Windows platform, use Internet Explorer.

Problem: On Windows platform, corrupted Latin-1 national characters appear in command prompt window

In a command prompt window on the Windows operating system, some national characters of the Latin-1 family might appear corrupted. For example, the letter "a" with a tilde may display as a pi symbol. To fix this, you must change the font properties of the command prompt window. To change the font, do the following:

1. Click the icon in the upper left corner of the command prompt window
2. Select Properties, then click the Font tab
3. The default font is Raster fonts; change this to Lucida Console and click OK

Problem: On HP-UX, Java out of memory/thread error occurs

Some HP-UX 11i installations are pre-configured to allow only 64 threads per process. However, some Load Balancer configurations require more than this amount. For HP-UX systems, set the threads per process to at least 256. To increase this value, use the "sam" utility to set the max_thread_proc kernel parameter. If heavy use is expected, you might need to increase max_thread_proc beyond 256.

To increase max_thread_proc, refer to the steps to increase the max_thread_proc parameter.

Problem: On Windows systems, advisors and reach targets mark all servers down

When configuring your adapter on a Load Balancer machine, you must ensure that the following two settings are correct for the advisor to work:

- Disable Task Offloading, which is most commonly used on 3Com adapter cards.
- Enable Protocol 1 (ICMP) for IP protocols if you are enabling TCP/IP filtering. If ICMP is not enabled, the ping test to the backend server will not succeed.

Refer to the section on disabling task offloading for instructions.

Solving common problems—Cisco CSS Controller

Problem: ccoserver will not start

This problem can occur when another application is using one of the ports used by the Cisco CSS Controller's ccoserver. For more information, see "Checking Cisco CSS Controller port numbers" on page 248.

Problem: ccocontrol or lbadmin command fails

1. The ccocontrol command returns: **Error: Server not responding**. Or, the lbadmin command returns: **Error: unable to access RMI server**. These errors can result when your machine has a socksified stack. To correct this problem, edit the socks.cnf file to contain the following lines:

   ```
   EXCLUDE-MODULE java
   EXCLUDE-MODULE javaw
   ```

2. The administration consoles for Load Balancer interfaces (command line and graphical user interface) communicate with ccoserver using remote method invocation (RMI). The default communication uses three ports; each port is set in the ccoserver start script:
   - 13099 to receive commands from ccocontrol
   - 10004 to send metric queries to Metric Server
   - 13199 for the RMI server port

   This can cause problems when one of the administration consoles runs on the same machine as a firewall or through a firewall. For example, when Load Balancer runs on the same machine as a firewall, and you issue ccocontrol commands, you might see errors such as **Error: Server not responding**.

   To avoid this problem, edit the ccoserver script file to set the port used by RMI for the firewall (or other application). Change the line: CCO_RMISERVERPORT=14199 to CCO_RMISERVERPORT=*yourPort*. Where *yourPort* is a different port.

   When complete, restart ccoserver and open traffic for ports 13099, 10004, 13199, and 13100, or for the chosen port for the host address from which the administration console will be run.
3. These errors can also occur if you have not already started **ccoserver**.

Problem: Cannot create registry on port 13099

This problem can occur when a valid product license is missing. When you attempt to start ccoserver, you receive the following message:

```
Your license has expired.  Contact your local IBM
representative or authorized IBM reseller.
```

To correct this problem:

1. If you have already attempted to start ccoserver, type **ccoserver stop**.
2. Copy your valid license to the following directory:
   - AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/conf**
   - Windows operating systems: **<***install_root***>ibm\edge\lb\servers\conf**
3. Type **ccoserver** to start the server.

Problem: On Windows platform, unexpected GUI behavior when using Matrox AGP video cards

On Windows platform when using a Matrox AGP card, unexpected behavior can occur in the Load Balancer GUI. When clicking the mouse, a block of space slightly larger than the mouse pointer can become corrupted causing possible highlighting reversal or images to shift out of place on the screen. Older Matrox cards have not shown this behavior. There is no known fix when using Matrox AGP cards.

Problem: Received a connection error when adding a consultant

You might experience a connection error, due to incorrect configuration settings, when adding a consultant. To fix this problem:

- Ensure that the Address or community specified exactly match the values configured on the switch.
- Ensure that connectivity between the controller and the switch is available.
- Ensure that the community has read-write permission on the switch. The controller will attempt to enable the `ApSvcLoadEnable` (SNMP) variable when testing the connection to verify write access.

Problem: Weights are not being updated on the switch

To fix this problem

- If you are using the Active connections or Connection rate metrics, issue `ccocontrol service SWID:OCID:serviceIO report`. Verify the metric values are changing according to throughput traffic on the switch.
- Increase the loglevel of the consultant log and look for SNMP TimeOut occurrences. If timeouts are occurring, possible solutions include:
  - Reducing the load on the switch.
  - Decreasing the network delay between the switch and the controller.
- Stop and restart the consultant.

Problem: Refresh command did not update the consultant configuration

Increase the consultant loglevel and retry the command. If it fails again, search the log for SNMP timeout or other SNMP communication errors.

Problem: Disconnect from host occurs when resize Netscape browser window while using Web administration

If you are using remote Web administration to configure Load Balancer, do not resize (Minimize, Maximize, Restore Down, and so on) the Netscape browser window in which the Load Balancer GUI appears. Because Netscape reloads a page every time browser windows are resized, this will cause a disconnect from

host. You will need to reconnect to host each time you resize the window. If you are performing remote Web administration on a Windows platform, use Internet Explorer.

Problem: On Windows platform, corrupted Latin-1 national characters appear in command prompt window

In a command prompt window on the Windows operating system, some national characters of the Latin-1 family might appear corrupted. For example, the letter "a" with a tilde may display as a pi symbol. To fix this, you must change the font properties of the command prompt window. To change the font, do the following:

1. Click the icon in the upper left corner of the command prompt window
2. Select Properties, then click the Font tab
3. The default font is Raster fonts; change this to Lucida Console and click OK

Problem: On HP-UX, Java out of memory/ thread error occurs

Some HP-UX 11i installations are pre-configured to allow only 64 threads per process. However, some Load Balancer configurations require more than this amount. For HP-UX systems, set the threads per process to at least 256. To increase this value, use the "sam" utility to set the max_thread_proc kernel parameter. If heavy use is expected, you might need to increase max_thread_proc beyond 256.

To increase max_thread_proc, refer to the steps to increase the max_thread_proc parameter.

Solving common problems—Nortel Alteon Controller

Problem: nalserver will not start

This problem can occur when another application is using one of the ports used by the Nortel Alteon Controller's nalserver. For more information, see “Checking Nortel Alteon Controller port numbers” on page 248.

Problem: nalcontrol or lbadmin command fails

1. The nalcontrol command returns: **Error: Server not responding**. Or, the lbadmin command returns: **Error: unable to access RMI server**. These errors can result when your machine has a socksified stack. To correct this problem, edit the socks.cnf file to contain the following lines:

   ```
   EXCLUDE-MODULE java
   EXCLUDE-MODULE javaw
   ```

2. The administration consoles for Load Balancer interfaces (command line and graphical user interface) communicate with nalserver using remote method invocation (RMI). The default communication uses three ports; each port is set in the nalserver start script:
   - 14099 to receive commands from nalcontrol
   - 10004 to send metric queries to Metric Server
   - 14199 for the RMI server port

   This can cause problems when one of the administration consoles runs on the same machine as a firewall or through a firewall. For example, when Load Balancer runs on the same machine as a firewall, and you issue nalcontrol commands, you might see errors such as **Error: Server not responding**.

To avoid this problem, edit the nalserver script file to set the port used by RMI for the firewall (or other application). Change the line: NAL_RMISERVERPORT=14199 to NAL_RMISERVERPORT=*yourPort*. Where *yourPort* is a different port.

When complete, restart nalserver and open traffic for ports 14099, 10004, 14199, and 14100, or for the chosen port for the host address from which the administration console will be run.

3. These errors can also occur if you have not already started **nalserver**.

Problem: Cannot create registry on port 14099

This problem can occur when a valid product license is missing. When you attempt to start nalserver, you receive the following message:

```
Your license has expired.  Contact your local IBM
representative or authorized IBM reseller.
```

To correct this problem:

1. If you have already attempted to start nalserver, type **nalserver stop**.
2. Copy your valid license to the
   - AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/conf**
   - Windows operating systems: **<***install_root***>ibm\edge\lb\servers\conf**
3. Type **nalserver** to start the server.

Problem: On Windows platform, unexpected GUI behavior when using Matrox AGP video cards

On Windows platform when using a Matrox AGP card, unexpected behavior can occur in the Load Balancer GUI. When clicking the mouse, a block of space slightly larger than the mouse pointer can become corrupted causing possible highlighting reversal or images to shift out of place on the screen. Older Matrox cards have not shown this behavior. There is no known fix when using Matrox AGP cards.

Problem: Disconnect from host occurs when resize Netscape browser window while using Web administration

If you are using remote Web administration to configure Load Balancer, do not resize (Minimize, Maximize, Restore Down, and so on) the Netscape browser window in which the Load Balancer GUI appears. Because Netscape reloads a page every time browser windows are resized, this will cause a disconnect from host. You will need to reconnect to host each time you resize the window. If you are performing remote Web administration on a Windows platform, use Internet Explorer.

Problem: Received a connection error when adding a consultant

You might experience a connection error, due to incorrect configuration settings, when adding a consultant. To fix this problem:

- Ensure that the Address or community specified exactly match the values configured on the switch.
- Ensure that connectivity between the controller and the switch is available.

- Ensure that the community has read-write permission on the switch. The controller will attempt to enable the `ApSvcLoadEnable` (SNMP) variable when testing the connection to verify write access.

Problem: Weights are not being updated on the switch

To fix this problem

- If you are using the Active connections or Connection rate metrics, issue `ccocontrol service SWID:OCID:serviceIO report`. Verify the metric values are changing according to throughput traffic on the switch.
- Increase the loglevel of the consultant log and look for SNMP TimeOut occurrences. If timeouts are occurring, possible solutions include:
  - Reducing the load on the switch.
  - Decreasing the network delay between the switch and the controller.
- Stop and restart the consultant.

Problem: Refresh command did not update the consultant configuration

Increase the consultant loglevel and retry the command. If it fails again, search the log for SNMP timeout or other SNMP communication errors.

Problem: On Windows systems, corrupted Latin-1 national characters appear in command prompt window

In a command prompt window on the Windows platform operating system, some national charactes of the Latin-1 family might appear corrupted. For example, the letter "a" with a tilde may display as a pi symbol. To fix this, you must change the font properties of the command prompt window. To change the font, do the following:

1. Click the icon in the upper left corner of the command prompt window
2. Select Properties, then click the Font tab
3. The default font is Raster fonts; change this to Lucida Console and click OK

Problem: On HP-UX, Java out of memory/ thread error occurs

Some HP-UX 11i installations are pre-configured to allow only 64 threads per process. However, some Load Balancer configurations require more than this amount. For HP-UX systems, set the threads per process to at least 256. To increase this value, use the "sam" utility to set the max_thread_proc kernel parameter. If heavy use is expected, you might need to increase max_thread_proc beyond 256.

To increase max_thread_proc, refer to the steps to increase the max_thread_proc parameter.

Solving common problems—Metric Server

Problem: Metric Server IOException on Windows platform running .bat or .cmd user metric files

You must use the full metric name for user-written metrics on Metric Servers running on Windows platform. For example, you must specify **usermetric.bat** instead of **usermetric**. The name **usermetric** is valid on the command line, but will not work when run from within the runtime environment. If you do not use the full metric name, you will receive a Metric Server IOException. Set the

LOG_LEVEL variable to a value of 3 in the metricserver command file, then check the log output. In this example, the exception appears as:

```
... java.io.IOException: CreateProcess: usermetric error=2
```

Problem: Metric Server not reporting loads to Load Balancer machine

There can be several reasons why Metric Server is not reporting load information to Load Balancer. To determine the cause, perform the following checks:

- Ensure that the key files have been transferred to Metric Server.
- Verify the host name of the Metric Server machine is registered with the local nameserver.

  You can also resolve this problem by specifying the host name in the Java property `java.rmi.server.hostname` in the metricserver script.
- Restart with a higher loglevel and look for errors.
- On the Load Balancer machine, increase the logging level for the Metric Monitor log using the **dscontrol manager metric set** command. Search for errors in the MetricMonitor.log file.

Problem: Metric Server log reports "Signature is necessary for access to agent"

The Metric Server log reports this error message after key files have been transferred to the server.

This error is logged when the key file fails authorization with the paired key due to corruption in the pair. To correct this problem try the following:

- FTP the key file again using the binary transfer method.
- Create new key and redistribute it.

Problem: On AIX systems, while running Metric Server under heavy stress, ps –vg command output may become corrupted

While running Metric Server under heavy stress on a multi-processor AIX platform (4.3.3, 32-bit 5.1, or 64-bit 5.1), output from ps -vg command may be corrupt. For example:

```
55742 - A 88:19 42 18014398509449680  6396 32768 22 36 2.8 1.0 java -Xms
```

The SIZE and/or RSS field of the ps command may show an excessive amount of memory being used.

This is a known AIX kernel problem. Apar IY33804 will correct this problem. Obtain the fix from AIX support at `http://techsupport.services.ibm.com/server/fixes`, or contact your local AIX support representative.

Problem: Configuring Metric Server in a two-tier configuration with Site Selector load-balancing across high-availability Dispatchers

In a two-tier Load Balancer configuration, if Site Selector (first-tier) is load balancing across a pair of Dispatcher high-availability partners (second-tier), there are steps you must complete to configure the metric server component. You must

configure metric server to listen on a new IP address that is specifically for metric server's use. On the two high-availability Dispatcher machines, metric server is active only on the active Dispatcher.

To correctly configure this setup, complete the following steps:

- Configure the metric server to listen on the new local IP. It should not be left to respond on the local NFA address. Refer to "Metric Server" on page 157 for configuration information.
- Because Site Selector should only be communicating with the active Dispatcher, you must start and stop metric server in the high-availability go scripts. To start or stop metric server correctly, alias the new metric server specific IP on the machine. Modify the go scripts to move the metric server IP address (similar to moving cluster addresses) so that the goActive script moves the metric server IP from the loopback to a physical adapter, and the goStandby script does the reverse. After moving the IP address, the goActive script must run the metricserver command to start metric server. The goStandby script must run **metricserver stop** to prevent metric server from talking to Site Selector while in standby mode.
- On Windows platform, refer to "Using scripts" on page 167 for moving the metric server specific IP address.
- The goStandby script changes include operating specific instructions as follows:
  - **HP-UX, Linux, and Solaris systems**: In the section within the goStandby script where the cluster address is moved to the loopback, insert commands to move the metric server specific IP to the loopback. Next, insert the **metricserver stop** command to stop metric server from responding to Site Selector.
  - **AIX systems**: In the section within the goStandby script where the cluster address is moved to the loopback, insert commands to move the metric server specific IP address to the loopback. Next, add a route so you can communicate with the loopback alias. Run the command **route add** *metricserverIP* **127.0.0.1**. Then insert the **metricserver stop** command to prevent metric server from further responding to Site Selector. After Metric Server stops, the final step is to remove the loopback route. To prevent any future confusion, insert **route delete** *metricserverIP*.

    For example:

    ```
    ifconfig en0 delete 9.27.23.61
    ifconfig lo0 alias 9.27.23.61 netmask 255.255.255.0
    route add 9.27.23.61 127.0.0.1
    metricserver stop
    # Sleep either max 60 seconds or until the metricserver stops
    let loopcount=0
    while [[ "$loopcount" -lt "60" && 'ps -ef | grep AgentStop|
             grep -c -v gr ep' -eq "1"]]
    do
     sleep 1
     let loopcount=$loopcount+1
    done
    route delete 9.27.23.61
    ```

  - **Windows** systems: First have the metric server loopback adapter (which is called the Local Area Connection 2 in the following example) installed on your machine with an IP address. Add an unused private network type of address to it, such as 10.1.1.1. After configuring the loopback, make changes to the go scripts. The goStandby script will include the netsh command to move the metric server IP to the metric server loopback adapter. Then run the **metricserver stop** command.

    For example:

```
call netsh interface ip delete address "Local Area Connection" addr=9.27.23.61
call netsh interface ip add address "Local Area Connection 2" addr=9.27.2.3.61
  mask = 255.255.255.0
sleep 3
metricserver stop
```

Problem: Scripts, running on multi-CPU Solaris machines, produce unwanted console messages

When running on multi-CPU Solaris machines, metricserver, cpuload, and memload scripts can produce unwanted console messages. This behavior is due to the use of the VMSTAT system command to gather CPU and memory statistics from the kernel. Some messages that VMSTAT returns indicate that the state of the kernel has changed. The scripts are unable to handle these messages, resulting in unnecessary console messages from the shell.

Examples of these console messages are:

```
/opt/ibm/edge/lb/ms/script/memload[29]: TOTAL=: syntax error
/opt/ibm/edge/lb/ms/script/memload[31]: LOAD=4*100/0: divide by zero
/opt/ibm/edge/lb/ms/script/memload[29]: TOTAL=659664+: more tokens expected
```

These messages can be ignored.

Problem: After starting Metric Server, metric value returns -1

This problem might be the result of the key files losing their integrity during transfer to client.

If you are using FTP to transfer your key files from the Load Balancer machine to the backend server ensure that you are using binary mode to `put` or `get` key files to or from the FTP server.

Part 9. Command Reference

This part provides command reference information for all the Load Balancer components. It contains the following chapters:

Chapter 25. How to read a syntax diagram

The syntax diagram shows you how to specify a command so that the operating system can correctly interpret what you type. Read the syntax diagram from left to right and from top to bottom, following the horizontal line (the main path).

Symbols and punctuation

The following symbols are used in syntax diagrams:

Symbol
Description

►► Marks the beginning of the command syntax.

►◄ Marks the end of the command syntax.

You must include all punctuation such as colons, quotation marks, and minus signs that are shown in the syntax diagram.

Parameters

The following types of parameters are used in syntax diagrams.

Parameter
Description
Required
Required parameters are displayed on the main path.
Optional
Optional parameters are displayed below the main path.

Parameters are classified as keywords or variables. Keywords are displayed in lowercase letters and can be entered in lowercase. For example, a command name is a keyword. Variables are italicized and represent names or values you supply.

Syntax examples

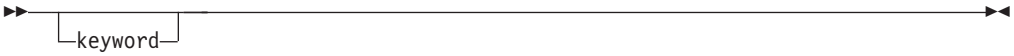
In the following example, the user command is a keyword. The required variable is *user_id*, and the optional variable is *password*. Replace the variables with your own values.

Required keywords: required keywords and variables appear on the main path line.

You must code required keywords and values.

Choose one required item from a stack: If there is more than one mutually exclusive required keyword or variable to choose from, they are stacked vertically in alphanumeric order.

Optional values: Optional keywords and variables appear below the main path line.

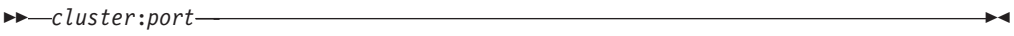


You can choose not to code optional keywords and variables.

Choose one optional keyword from a stack: If there is more than one mutually exclusive optional keyword or variable to choose from, they are stacked vertically in alphanumeric order below the main path line.

Variables: A word in all italics is a *variable*. Where you see a variable in the syntax, you must replace it with one of its allowable names or values, as defined in the text.

Nonalphanumeric characters: If a diagram shows a character that is not alphanumeric (such as colons, quotes, or minus signs), you must code the character as part of the syntax. In this example, you must code *cluster:port*.



Chapter 26. Command reference for Dispatcher and CBR

This chapter describes how to use the Dispatcher **dscontrol** commands. It is also a command reference for CBR.

For previous versions, when the product was known as Network Dispatcher, the Dispatcher control command name was ndcontrol. The Dispatcher control command name is now **dscontrol**. Ensure you update all previous script files to use dscontrol (not ndcontrol) for configuring Dispatcher.

CBR uses a subset of the Dispatcher commands listed in this command reference. When using these syntax diagrams for **CBR**, substitute **cbrcontrol** for dscontrol. For information, see "Configuration differences between CBR and Dispatcher" on page 288.

The following list contains the commands noted in this chapter:

You can enter a minimized version of the dscontrol command parameters. You only need to enter the unique letters of the parameters. For example, to get help on the file save command, you can type **dscontrol he f** instead of **dscontrol help file**.

To start up the command-line interface: issue **dscontrol** to receive a dscontrol command prompt.

To end the command line interface: issue **exit** or **quit**.

The command parameter values must be entered in English characters. The only exceptions are host names (used in cluster, server, and highavailability commands) and file names (used in file commands).

Configuration differences between CBR and Dispatcher

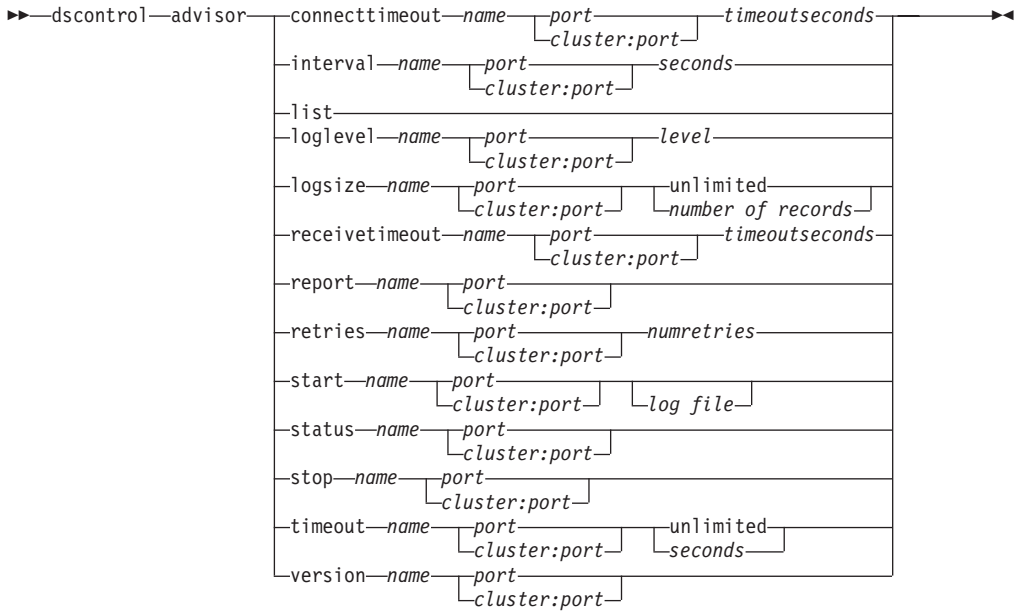
The CBR command line interface is a subset of the command line interface of Dispatcher. For CBR, substitute the **cbrcontrol** command instead of dscontrol to configure the component.

Note: The Content Based Routing (CBR) component is not available on platforms that run a 64-bit JVM, except for HP-UX ia64. On HP-UX ia64, the CBR component runs as a 32-bit application. You can use the CBR forwarding method of Load Balancer's Dispatcher component to provide content-based routing without the use of Caching Proxy. See "Dispatcher's content-based routing (cbr forwarding method)" on page 41 for more information.

Some of the commands that are *omitted* in CBR are listed below.

1. highavailability
2. subagent
3. executor
   - report
   - set nfa <value>
   - set fintimeout <value>
   - set hatimeout <value>
   - set hasynctimeout <value>
   - set porttype <value>
4. cluster
   - report {c}
   - set {c} porttype
5. port
   - add {c:p} porttype
   - add {c:p} protocol
   - set {c:p} porttype
6. rule add {c:p:r} type port
7. server
   - add {c:p:s} router
   - set {c:p:s} router

dscontrol advisor — control the advisor



connecttimeout
Set how long an advisor waits before reporting that a connect to a server for a particular port on a server (a service) fails. For more information, see "Advisor connect timeout and receive timeout for servers" on page 149.

name
The name of the advisor. Possible values include **connect**, **db2**, **dns**, **ftp**, **http**, **https**, **cachingproxy**, **imap**, **ldap**, **ldaps**, **nntp**, **ping**, **pop3**, **self**, **sip**, **smtp**, **ssl**, **ssl2http**, **telnet**, and **wlm**.

See "List of advisors" on page 149 for more information on the advisors that Load Balancer provides.

Names of customized advisors are of the format xxxx, where ADV_*xxxx* is the name of the class that implements the custom advisor. See "Create custom (customizable) advisors" on page 154 for more information.

port
The number of the port that the advisor is monitoring.

cluster:port
The cluster value is optional on the advisor commands, but the port value is required. If the cluster value is not specified, then the advisor will start running on the port for all clusters. If you specify a cluster, then the advisor will start running on the port, but only for the cluster you have specified. See "Starting and stopping an advisor" on page 147 for more information.

The cluster is the address in IP address format or symbolic name. The port is the number of the port that the advisor is monitoring.

timeoutseconds
A positive integer representing the timeout in seconds at which the advisor waits before reporting that a connect to a server fails. The default is 3 times the value specified for the advisor interval.

interval
Set how often the advisor will query the servers for information.

seconds
A positive integer representing the number of seconds between requests to the servers about their current status. The default is 7.

list
Show list of advisors that are currently providing information to the manager.

loglevel
Set the logging level for an advisor log.

level
The number of the level (0 to 5). The default is 1. The higher the number, the more information that is written to the advisor log. The following are the possible values: 0 is None, 1 is Minimal, 2 is Basic, 3 is Moderate, 4 is Advanced, 5 is Verbose.

logsize
Set the maximum size of an advisor log. When you set a maximum size for the log file, the file will wrap; when the file reaches the specified size, the subsequent entries are written from the top of the file, overwriting the previous log entries. Log size cannot be set smaller than the current size of the log. Log entries are time–stamped so you can tell the order in which they were written. The higher you set the log level, the more carefully you should choose the log size, because you can quickly run out of space when logging at the higher levels.

number of records
The maximum size in bytes for the advisor log file. You can specify either a positive number greater than zero, or the word **unlimited**. The log file may not reach the exact maximum size before overwriting because the log entries themselves vary in size. The default value is 1 MB.

receivetimeout
Set how long an advisor waits before reporting that a receive from a particular port on a server (a service) fails. For more information, see "Advisor connect timeout and receive timeout for servers" on page 149.

timeoutseconds
A positive integer representing the timeout in seconds at which the advisor waits before reporting that a receive from a server fails. The default is 3 times the value specified for the advisor interval.

report
Display a report on the state of the advisor.

retry
Retry sets the number of retries that an advisor can make before marking a server down.

numretries
An integer greater than or equal to zero. This value should be no larger than 3. If retries keyword is not configured, the number of retries defaults to zero.

start
Start the advisor. There are advisors for each protocol. The default ports are as follows:

Advisor Name	Protocol	Port
cachingproxy	HTTP (via Caching Proxy)	80
connect	ICMP	12345

Advisor Name	Protocol	Port
db2	private	50000
dns	DNS	53
ftp	FTP	21
http	HTTP	80
https	SSL	443
imap	IMAP	143
ldap	LDAP	389
ldaps	SSL	443
nntp	NNTP	119
ping	PING	0
pop3	POP3	110
self	private	12345
sip	SIP	5060
smtp	SMTP	25
ssl	SSL	443
ssl2http	SSL	443
telnet	Telnet	23
WLM	private	10,007

Note: The FTP advisor should advise only on the FTP control port (21). Do not start an FTP advisor on the FTP data port (20).

log file
File name to which the management data is logged. Each record in the log is time–stamped.

The default file is *advisorname_port*.log, for example, **http_80.log.** To change the directory where the log files are kept, see "Changing the log file paths" on page 219. The default log files for cluster (or site) specific advisors are created with the cluster address, for example, **http_127.40.50.1_80.log**.

status
Display the current status of all the values in an advisor that can be set globally and their defaults.

stop
Stop the advisor.

timeout
Set the number of seconds for which the manager will consider information from the advisor as valid. If the manager finds that the advisor information is older than this timeout period, the manager will not use that information in determining weights for the servers on the port the advisor is monitoring. An exception to this timeout is when the advisor has informed the manager that a specific server is down. The manager will use that information about the server even after the advisor information has timed out.

seconds
A positive number representing the number of seconds or the word **unlimited**. The default value is unlimited.

version
Display the current version of the advisor.

Examples

- To start the http advisor on port 80 for cluster 127.40.50.1:

```
dscontrol advisor start http 127.40.50.1:80
```

- To start the http advisor on port 88 for all clusters:

```
dscontrol advisor start http 88
```

- To stop the http advisor at port 80 for cluster 127.40.50.1:

```
dscontrol advisor stop http 127.40.50.1:80
```

- To set the time (30 seconds) an HTTP advisor for port 80 waits before reporting that a connect to a server fails:

```
dscontrol advisor connecttimeout http 80 30
```

- To set the time (20 seconds) an HTTP advisor for port 80 on cluster 127.40.50.1 waits before reporting that a connect to a server fails:

```
dscontrol advisor connecttimeout http 127.40.50.1:80 20
```

- To set the interval for the FTP advisor (for port 21) to 6 seconds:

```
dscontrol advisor interval ftp 21 6
```

- To display the list of advisors currently providing information to the manager:

```
dscontrol advisor list
```

  This command produces output similar to:

```
--------------------------------------
| ADVISOR | CLUSTER:PORT  |  TIMEOUT  |
--------------------------------------
|  http   |127.40.50.1:80 | unlimited |
|  ftp    |            21 | unlimited |
--------------------------------------
```

- To change the log level of the advisor log to 0 for better performance:

```
dscontrol advisor loglevel http 80 0
```

- To change the ftp advisor log size for port 21 to 5000 bytes:

```
dscontrol advisor logsize ftp 21 5000
```

- To set the time (60 seconds) an HTTP advisor (for port 80) waits before reporting that a receive from a server fails:

```
dscontrol advisor receivetimeout http 80 60
```

- To display a report on the state of the ftp advisor (for port 21):

```
dscontrol advisor report ftp 21
```

  This command produces output similar to:

```
Advisor Report:
---------------
Advisor name ............. Ftp
Port number .............. 21

Cluster address .......... 9.67.131.18
Server address ........... 9.67.129.230
Load ..................... 8

Cluster address .......... 9.67.131.18
Server address ........... 9.67.131.215
Load ..................... -1
```

- To display the current status of values associated with the http advisor for port 80:

```
dscontrol advisor status http 80
```

This command produces output similar to the following:

```
Advisor Status:
---------------
Interval (seconds) ............ 7
Timeout (seconds) ............. Unlimited
Connect timeout (seconds).......21
Receive timeout (seconds).......21
Advisor log filename .......... Http_80.log
Log level ..................... 1
Maximum log size (bytes) ...... Unlimited
Number of retries ............. 0
```

- To set the timeout value for the ftp advisor information on port 21 to 5 seconds:

  ```
  dscontrol advisor timeout ftp 21 5
  ```

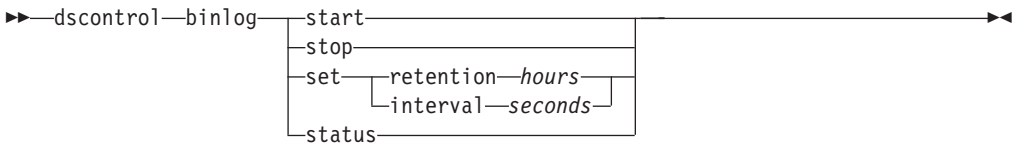
- To display the current version number of the ssl advisor for port 443:

  ```
  dscontrol advisor version ssl 443
  ```

  This command produces output similar to the following:

  ```
  Version: 04.00.00.00 - 07/12/2001-10:09:56-EDT
  ```

dscontrol binlog — control the binary log file



start
Starts the binary log.

stop
Stops the binary log.

set
Sets fields for binary logging. For more information on setting fields for binary logging, see "Using binary logging to analyze server statistics" on page 194.

retention
The number of hours that binary log files are kept. The default value for retention is 24.

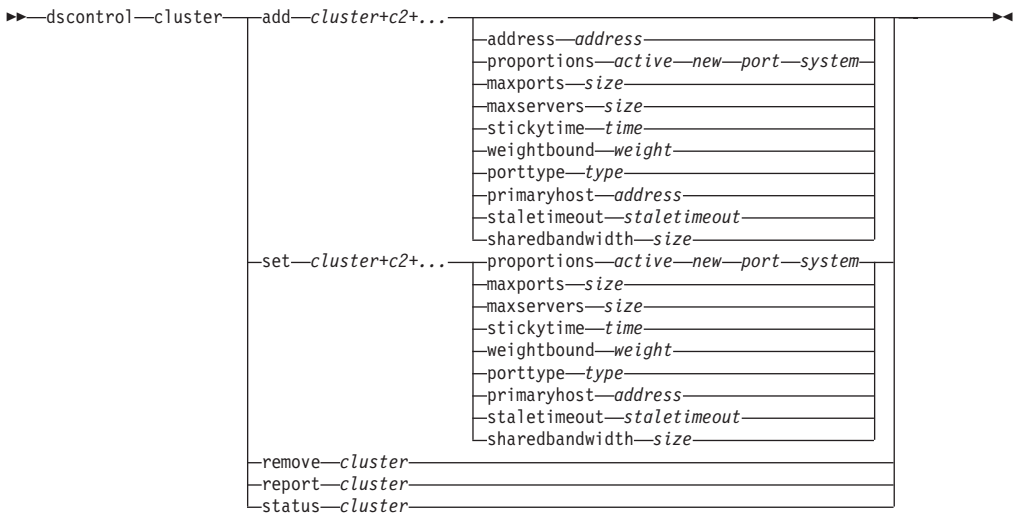
hours
The number of hours.

interval
The number of seconds between log entries. The default value for interval is 60.

seconds
The number of seconds.

status
Shows the retention and intervals of the binary log.

dscontrol cluster — configure clusters



add
Add this cluster. You must define at least one cluster.

cluster
The cluster name or address to which clients connect. The cluster value is either a symbolic name or in IP address format. A cluster value of 0.0.0.0 can be used to specify a wildcard cluster. See "Use wildcard cluster to combine server configurations" on page 191 for more information.

With the exception of the dscontrol cluster add command, you can use a colon (:) to act as a wild card. For example, the following command, `dscontrol cluster set : weightbound 80`, will result in setting a weightbound of 80 to all clusters.

Note: Additional clusters are separated by a plus sign (+).

address
The unique IP address of the TCP machine as either a host name or in IP address format. If the cluster value is unresolvable, you must provide this IP address of the physical machine.

Note: Address only applies to the Dispatcher component.

address
Value of the address of the cluster.

proportions
At the cluster level, set the proportion of importance for active connections (*active*), new connections (*new*), information from any advisors (*port*), and information from a system monitoring program such as Metric Server (*system*) that are used by the manager to set server weights. Each of these values, described below, is expressed as a percentage of the total and they therefore always total 100. For more information see, "Proportion of importance given to status information" on page 142.

active
A number from 0–100 representing the proportion of weight to be given to the active connections. The default is 50.

new
A number from 0–100 representing the proportion of weight to be given to the new connections. The default is 50.

port
A number from 0–100 representing the proportion of weight to be given to the information from advisors. The default is 0.

Note: When an advisor is started and if the port proportion is 0, Load Balancer automatically sets this value to 1 in order for the manager to use the advisor information as input for calculating server weight.

system
A number from 0–100 representing the proportion of weight to be given to the information from the system metrics, such as from Metric Server. The default is 0.

maxports
The maximum number of ports. The default value of maxports is 8.

size
The number of ports allowed.

maxservers
The default maximum number of servers per ports. This may be overridden for individual ports using **port maxservers** . The default value of maxservers is 32.

size
The number of servers allowed on a port.

stickytime
The default stickytime for ports to be created. This may be overridden for individual ports using **port stickytime**. The default value of stickytime is 0.

Note: For the Dispatcher's cbr forwarding method, if you set stickytime (to a nonzero value), then port stickytime is enabled if the port is SSL (not HTTP). If stickytime for ports to be created is non-zero and the new port added is SSL, SSL ID affinity is enabled for the port. To disable SSL ID affinity on the port, you will need to explicitly set the port stickytime to 0.

time
The value of stickytime in seconds.

weightbound
The default port weight bound. This may be overridden for individual ports using **port weightbound**. The default value of weightbound is 20.

weight
The value of weightbound.

porttype
The default port type. This may be overridden for individual ports using **port porttype**.

type
Possible values are **tcp**, **udp**, and **both**.

primaryhost
The NFA address of this Dispatcher machine or the NFA address of the backup

Dispatcher machine. In a mutual high availability configuration, a cluster is associated with either the primary or the backup machine.

If you change the primaryhost of a cluster after the primary and backups are already started and running mutual high availability, you also must force the new primary host to takeover. And, you need to update the scripts and manually unconfigure and configure the cluster correctly. See "Mutual high availability" on page 46 for more information.

address
The address value of the primaryhost. The default is the NFA address of this machine.

staletimeout
The number of seconds during which there can be no activity on a connection before that connection is removed. The default for FTP is 900; the default for Telnet is 259,200. The default for all other protocols is 300. This may be overridden for individual ports using **port staletimeout**. See "Using stale timeout value" on page 220 for more information.

staletimout
The staletimeout value.

sharedbandwidth
The maximum amount of bandwidth (in kilobytes per second) that can be shared at the cluster level. For more information on shared bandwidth, see "Using rules based on reserved bandwidth and shared bandwidth" on page 173 and "Shared bandwidth rule" on page 174.

Note: Shared bandwidth applies to the Dispatcher component.

size
The size of **sharedbandwidth** is an integer value. The default is zero. If the value is zero, then bandwidth cannot be shared at the cluster level.

set
Set the properties of the cluster.

remove
Remove this cluster.

report
Show the internal fields of the cluster.

Note: Report applies to the Dispatcher component.

status
Show current status of a specific cluster.

Examples

- To add cluster address 130.40.52.153:

  ```
  dscontrol cluster add 130.40.52.153
  ```

- To remove cluster address 130.40.52.153:

  ```
  dscontrol cluster remove 130.40.52.153
  ```

- To set the relative importance placed on input (active, new, port, system) received by the manager for servers residing on cluster 9.6.54.12:

  ```
  dscontrol cluster set 9.6.54.12 proportions 60 35 5 0
  ```

- To add a wildcard cluster:

  ```
  dscontrol cluster add 0.0.0.0
  ```

- For a mutual high availability configuration, set cluster address 9.6.54.12 with the NFA of the backup machine (9.65.70.19) as the primary host:

  ```
  dscontrol cluster set 9.6.54.12 primaryhost 9.65.70.19
  ```

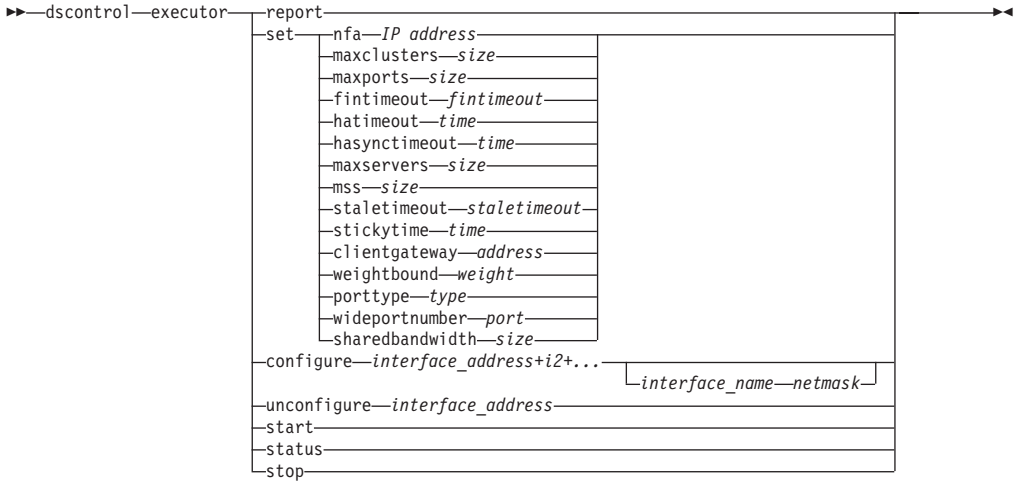
- To show the status for cluster address 9.67.131.167:

  ```
  dscontrol cluster status 9.67.131.167
  ```

  This command produces output similar to:

  ```
   Cluster Status:
  ---------------
  Cluster ................................ 9.67.131.167
  Address ................................ 9.67.131.167
  Number of target ports ................. 3
  Default sticky time .................... 0
  Default stale timeout .................. 30
  Default port weight bound .............. 20
  Maximum number of ports ................ 8
  Default port protocol .................. tcp/udp
  Default maximum number of servers ....... 32
  Proportion given to active connections... 0.5
  Proportion given to new connections...... 0.5
  Proportion given specific to the port.... 0
  Proportion given to system metrics....... 0
  Shared bandwidth (KBytes) .............. 0
  Primary Host Address ................... 9.67.131.167
  ```

dscontrol executor — control the executor



report
Display a statistics snapshot report. For example: total packets received, packets discarded, packets forwarded with errors, and so on.

Note: Report applies to the Dispatcher component.

set
Set the fields of the executor.

nfa
Set the nonforwarding address. Any packet sent to this address will not be forwarded by the Dispatcher machine.

Note: NFA applies to the Dispatcher component.

IP address
The Internet Protocol address as either a symbolic name or in dotted decimal format.

maxclusters
The maximum number of clusters that can be configured. The default value of maxclusters is 100.

size
The maximum number of clusters that can be configured.

maxports
The default value of maxports for clusters to be created. This may be overridden by the **cluster set** or **cluster add** command. The default value of maxports is 8.

size
The number of ports.

fintimeout
The number of seconds to keep a connection in memory after the connection has been put in the FIN state. The default fintimeout value is 30.

fintimeout
The fintimeout value.

Note: Fintimeout applies to the Dispatcher component.

hatimeout
The number of seconds that the executor uses to timeout high availability heartbeats. The default value is 2.

Note: The hatimeout value applies to the Dispatcher component.

time
The hatimeout value.

hasynctimeout
The number of seconds that the executor uses to timeout replication of connection records between the primary and backup machine. The default value is 50.

The timer is used to ensure that the primary and backup machines attempt to synchronize. However, if there are too many connections in existence, and the active machine continues to handle a significant incoming traffic load, then synchronization might not complete before the timer expires. As a result, Load Balancer attempts to resynchronize perpetually, and the two machines never synchronize. If this situation occurs, set hasynctimeout to a larger value than the default to give the two machines enough time to exchange information about existing connections. In order to set this timer, the hasynctimeout command must be issued after the `dscontrol executor start` command but before issuing the high availability commands (`dscontrol highavailability`).

Note: The hasynctimeout value applies to the Dispatcher component.

time
The hasynctimeout value.

maxservers
The default maximum number of servers per port. This may be overridden by the **cluster** or **port** command. The default value of maxservers is 32.

mss
The maximum number of bytes in the data segment of the TCP/UDP connection. The number of bytes in the data segment and the header must add up to less than the number of bytes in the maximum transmission unit (MTU). The default value of mss is 1460.

Note: Maximum segment size only applies to Dispatcher component's nat or cbr forwarding method.

size
The number of servers.

staletimeout
The number of seconds during which there can be no activity on a connection before that connection is removed. The default for FTP is 900; the default for Telnet is 259,200. The default for all other ports is 300. This may be overridden by the **cluster** or **port** command. See "Using stale timeout value" on page 220 for more information.

staletimeout
The staletimeout value.

stickytime
The default port sticky time value for all future clusters. It may be overridden by the **cluster** or **port** command. The default stickytime value is 0.

time
The stickytime value in seconds.

clientgateway
Clientgateway is an IP address used for NAT/NAPT or Dispatcher's content-based routing. It is the router address through which traffic in the return direction is forwarded from Load Balancer to clients. Clientgateway must be set to a nonzero value before adding a port with a forwarding method of NAT/NAPT or Dispatcher's content-based routing. See "Dispatcher's NAT/NAPT (nat forwarding method)" on page 39 and "Dispatcher's content-based routing (cbr forwarding method)" on page 41 for more information.

Note: Clientgateway only applies to the Dispatcher component.

address
The clientgateway address as either a symbolic name or in dotted decimal format. The default is 0.0.0.0.

weightbound
The default port weightbound value for all future ports. It may be overridden by the **cluster** or **port** command. The default weightbound value is 20.

weight
The weightbound value.

porttype
The default port porttype value for all future ports. It may be overridden by the **cluster** or **port** command.

Note: Porttype applies to the Dispatcher component.

type
Possible values are **tcp**, **udp**, and **both**.

wideportnumber
An unused TCP port on each Dispatcher machine. The *wideportnumber* must be the same for all the Dispatcher machines. The default value of wideportnumber is 0, indicating that wide area support is not in use.

Note: Wideportnumber applies to the Dispatcher component.

port
The value of **wideportnumber**.

sharedbandwidth
The maximum amount of bandwidth (in kilobytes per second) that can be shared at the executor level. For more information on shared bandwidth, see "Using rules based on reserved bandwidth and shared bandwidth" on page 173 and "Shared bandwidth rule" on page 174.

Note: Shared bandwidth applies to the Dispatcher component.

size
The size of **sharedbandwidth** is an integer value. The default is zero. If the value is zero, then bandwidth cannot be shared at the executor level.

configure
Configure an address (for example a cluster address, return address, or high

availability heartbeat address) to the network interface card of the Dispatcher machine. This is also known as configuring an alias on the Dispatcher machine.

Note: Configure applies to the Dispatcher component.

interface_address
The address as either a symbolic name or in IP address format.

Note: Additional interface addresses are separated by a plus sign (+).

interface_name netmask
It is only required if the address does not match any subnet for existing addresses. The *interface_name* can be a value such as: en0, eth1, eri0. The *netmask* is the 32-bit mask used to identify the subnetwork address bits in the host portion of an IP address.

unconfigure
Deletes the alias address from the network interface card.

Note: Unconfigure applies to the Dispatcher component.

start
Start the executor.

status
Display the current status of the values in the executor that can be set and their defaults.

stop
Stop the executor.

Note: Stop applies to Dispatcher and CBR.

Examples

- To display the internal counters for Dispatcher:

```
dscontrol executor status

 Executor Status:
----------------
Nonforwarding address ............... 9.67.131.151
Client gateway address .............. 0.0.0.0
Fin timeout ......................... 60
Wide area network port number ....... 0
Shared bandwidth (Kbytes) ........... 0
Default maximum ports per cluster ... 8
Maximum number of clusters .......... 100
Default maximum servers per port .... 32
Default stale timeout ............... 300
Default sticky time ................. 0
Default weight bound ................ 20
Default port type ................... tcp/udp
```

- To set the nonforwarding address to 130.40.52.167:

```
dscontrol executor set nfa 130.40.52.167
```

- To set the maximum number of clusters:

```
dscontrol executor set maxclusters 4096
```

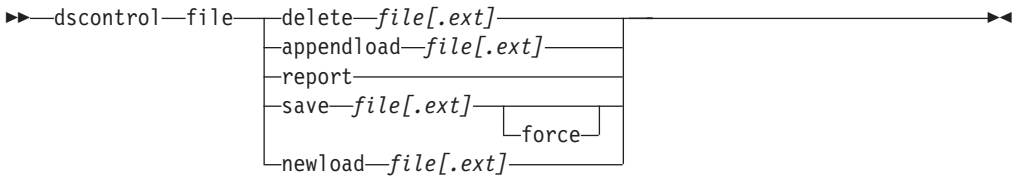
- To start the executor:

```
dscontrol executor start
```

- To stop the executor:

```
dscontrol executor stop
```

dscontrol file — manage configuration files



delete
Delete the file.

file[.ext]
A configuration file consisting of dscontrol commands.

The file extension (*.ext*) can be anything you like and can be omitted.

appendload
To update the current configuration, the appendload command runs the executable commands from your script file.

report
Report on the available file or files.

save
Save the current configuration for Load Balancer to the file.

Note: Files are saved into and loaded from the following directories, where ***component*** is either dispatcher or cbr:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/configurations/*component***
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\configurations*component***

force
To save your file to an existing file of the same name, use **force** to delete the existing file before saving the new file. If you do not use the force option, the existing file is not overwritten.

newload
Loads and runs a new configuration file into the Load Balancer. The new configuration file replaces the current configuration.

Examples

- To delete a file:

```
dscontrol file delete file3

File (file3) was deleted.
```

- To load a new configuration file to replace the current configuration:

```
dscontrol file newload file1.sv

File (file1.sv) was loaded into the Dispatcher.
```

- To append a configuration file to the current configuration and load:

```
dscontrol file appendload file2.sv

File (file2.sv) was appended to the current configuration and loaded.
```

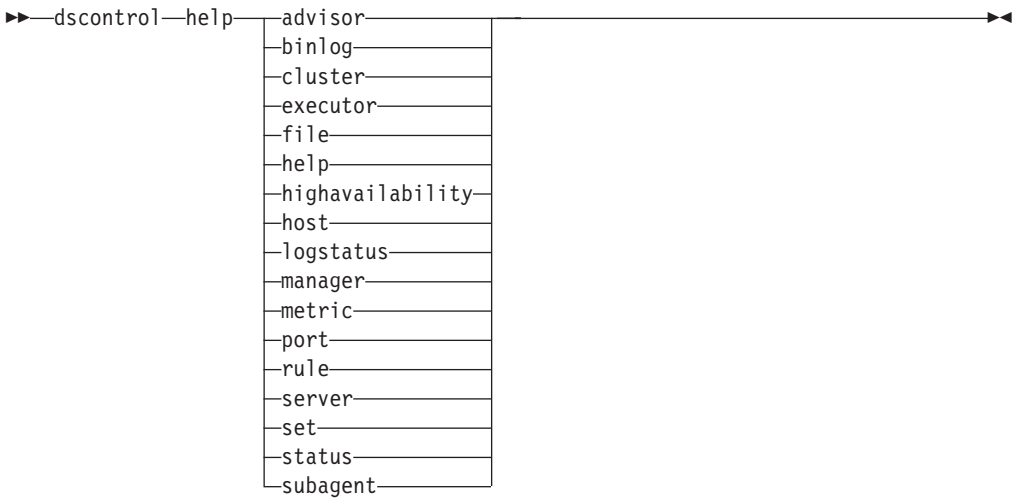
- To view a report of your files (that is, those files that you saved earlier):

```
dscontrol file report

FILE REPORT:
file1.save
file2.sv
file3
```

- To save your configuration into a file named file3:

```
dscontrol file save file3

The configuration was saved into file (file3).
```

dscontrol help — display or print help for this command



Examples

- To get help on the dscontrol command:

  ```
  dscontrol help
  ```

  This command produces output similar to:

  ```
  HELP COMMAND ARGUMENTS:
  --------------------------------
  Usage:  help <help option>
  Example: help cluster

  help             - print complete help text
  advisor          - help on advisor command
  cluster          - help on cluster command
  executor         - help on executor command
  file             - help on file command
  host             - help on host command
  binlog           - help on binary log command
  manager          - help on manager command
  metric           - help on metric command
  port             - help on port command
  rule             - help on rule command
  server           - help on server command
  set              - help on set command
  status           - help on status command
  logstatus        - help on server log status
  subagent         - help on subagent command
  highavailability - help on high availability command
  ```

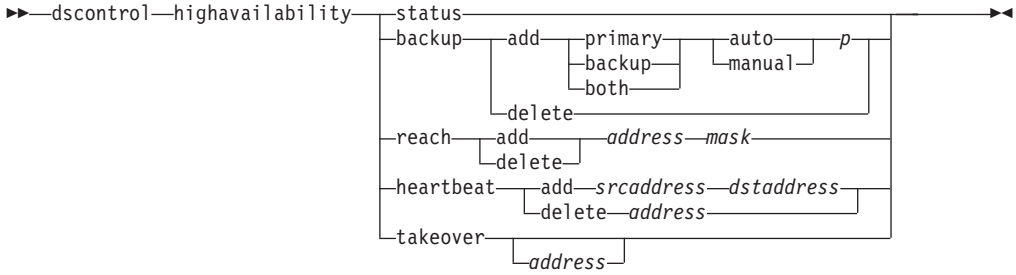
  Notice that parameters within <> are variables.

- Sometimes the help will show choices for the variables using | to separate the options:

  ```
  fintimeout <cluster address>|all <time>
   -Change FIN timeout
    (Use 'all' to change all clusters)
  ```

dscontrol highavailability — control high availability

Note: The dscontrol high availability syntax diagram only applies to the Dispatcher component.



status
Return a report on high availability. Machines are identified as having one of three status conditions or states:

Active A given machine (either a primary, backup, or both) is routing packets.

Standby
A given machine (either a primary, backup, or both) is not routing packets; it is monitoring the state of an **active** Dispatcher.

Idle A given machine is routing packets, and is not trying to establish contact with its partner Dispatcher.

In addition, the **status** keyword returns information about various substates:

Synchronized
A given machine has established contact with another Dispatcher.

Other substates
This machine is trying to establish contact with its partner Dispatcher but has not yet succeeded.

backup
Specify information for either the primary or backup machine.

add
Defines and runs the high availability functions for this machine.

primary
Identifies the Dispatcher machine that has a *primary* role.

backup
Identifies the Dispatcher machine that has a *backup* role.

both
Identifies the Dispatcher machine that has *both* a primary and backup role. This is the mutual high availability feature in which primary and backup roles are associated on a per cluster set basis. See "Mutual high availability" on page 46 for more information.

auto
Specifies an *automatic* recovery strategy, in which the primary machine will resume routing packets as soon as it comes back into service.

manual
Specifies a *manual* recovery strategy, in which the primary machine does not resume routing packets until the administrator issues a **takeover** command.

p[ort]
An unused TCP port on both machines, to be used by Dispatcher for its heartbeat messages. The *port* must be the same for both the primary and backup machines.

delete
Removes this machine from high availability, so that it will no longer be used as a backup or primary machine.

reach
Add or delete target address for the primary and backup Dispatchers, the reach advisor sends out *pings* from both the backup and the primary Dispatchers to determine how reachable their targets are.

Note: When configuring the reach target, you must also start the reach advisor. The reach advisor starts automatically by the manager function.

add
Adds a target address for the reach advisor.

delete
Removes a target address from the reach advisor.

address
IP address (IP address format or symbolic) of the target node.

mask
A subnet mask.

heartbeat
Defines a communication session between the primary and backup Dispatcher machines.

add
Tell the source Dispatcher the address of its partner (destination address).

srcaddress
Source address. The address (IP or symbolic) of this Dispatcher machine.

dstaddress
Destination address. The address (IP or symbolic) of the other Dispatcher machine.

Note: The srcaddress and dstaddress must be the NFAs of the machines for at least one heartbeat pair.

delete
Removes the address pair from the heartbeat information. You can specify either the destination or source address of the heartbeat pair.

address
The address (IP or symbolic) of either the destination or source.

takeover
Simple high availability configuration (role of the Dispatcher machines are either *primary* or *backup*):

- Takeover instructs a standby Dispatcher to become active and to begin routing packets. This will force the currently active Dispatcher to become

standby. The takeover command must be issued on the standby machine and works only when the strategy is **manual**. The substate must be *synchronized*.

Mutual high availability configuration (role of each Dispatcher machine is *both*):

- The Dispatcher machine with the mutual high availability feature contains two clusters which match its partner's. One of the clusters is considered the primary cluster (the partner's backup cluster), and the other is the backup cluster (the partner's primary cluster). Takeover instructs the Dispatcher machine to begin routing packets for the other machine's cluster(s). The takeover command can only be issued when the cluster(s) of the Dispatcher machine are in *standby* state and the substate is *synchronized*. This will force the partner's currently active cluster(s) to change to standby state. The takeover command works only when the strategy is **manual**. See "Mutual high availability" on page 46 for more information.

Notes:

1. Note that the *roles* of the machines (*primary*, *backup*, *both*) do not change. Only their relative *status* (*active* or *standby*) changes.
2. There are three possible takeover *scripts*: goActive, goStandby, and goInOp. See "Using scripts" on page 167.

address

The takeover address value is optional. It should only be used when the role of the machine is *both* primary and backup (mutual high availability configuration). The address specified is the NFA of the Dispatcher machine which normally routes this cluster's traffic. When there is a takeover of both clusters, specify the Dispatcher's own NFA address.

Examples

- To check the high availability status of a machine:

```
dscontrol highavailability status
```

Output:

```
High Availability Status:
-------------------------
Role .......................primary
Recovery Strategy ........... manual
State ...................... Active
Sub-state.............. Synchronized
Primary host........... 9.67.131.151
Port .........................12345
Preferred Target....... 9.67.134.223

Heartbeat Status:
-----------------
Count ......................... 1
Source/destination ............ 9.67.131.151/9.67.134.223

Reachability Status:
--------------------
Count ................ 1
Address .............. 9.67.131.1   reachable
```

- To add the backup information to the primary machine using the automatic recovery strategy and port 80:

```
dscontrol highavailability backup add primary auto 80
```

- To add an address that the Dispatcher must be able to reach:

```
dscontrol highavailability reach add 9.67.125.18
```

- To add heartbeat information for the primary and backup machines.

  ```
  Primary - highavailability heartbeat add 9.67.111.3 9.67.186.8
  Backup  - highavailability heartbeat add 9.67.186.8 9.67.111.3
  ```

- To tell the standby Dispatcher to become active, forcing the active machine to become standby:

  ```
  dscontrol highavailability takeover
  ```

dscontrol host — configure a remote machine

remote_host
The name of the remote Load Balancer machine being configured. When typing this command, make sure there is no space between **host:** and *remote_host*, for example:

```
dscontrol host:remote_host
```

After this command has been issued on the command prompt, enter any valid dscontrol command you want issued to the remote Load Balancer machine.

dscontrol logstatus — display server log settings

logstatus
Displays the server log settings (log file name, logging level, and log size).

Examples

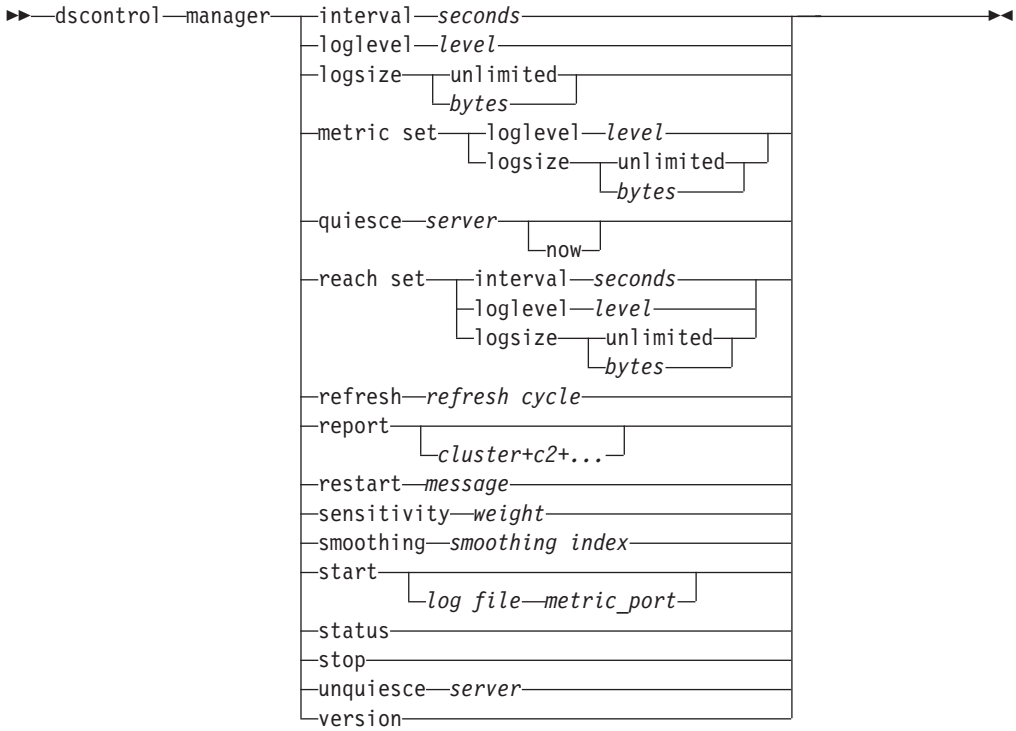
To display the logstatus:

```
dscontrol logstatus
```

This command produces output similar to:

```
Dispatcher Log Status:
------------------------------
Log filename ............... C:\PROGRA~1\IBM\edge\lb\servers\logs\dispatcher
\server.log
Log level .................. 1
Maximum log size (bytes) ... 1048576
```

dscontrol manager — control the manager



interval
Set how often the manager will update the weights of the servers to the executor, updating the criteria that the executor uses to route client requests.

seconds
A positive number representing in seconds how often the manager will update weights to the executor. The default is 2.

loglevel
Set the logging level for the manager log.

level
The number of the level (0 to 5). The higher the number, the more information that is written to the manager log. The default is 1. The following are the possible values: 0 is None, 1 is Minimal, 2 is Basic, 3 is Moderate, 4 is Advanced, 5 is Verbose.

logsize
Set the maximum size of the manager log. When you set a maximum size for the log file, the file will wrap; when the file reaches the specified size, the subsequent entries are written from the top of the file, overwriting the previous log entries. Log size cannot be set smaller than the current size of the log. Log entries are time stamped so you can tell the order in which they were written. The higher you set the log level, the more carefully you should choose the log size, because you can quickly run out of space when logging at the higher levels.

bytes
The maximum size in bytes for the manager log file. You can specify either a positive number greater than zero, or the word **unlimited**. The log file may not

reach the exact maximum size before overwriting because the log entries themselves vary in size. The default value is 1 MB.

metric set
Sets the **loglevel** and **logsize** for the metric monitor log. The loglevel is the metric monitor logging level (0 - None,1 - Minimal,2 - Basic,3 - Moderate, 4 - Advanced, or 5 - Verbose). The default loglevel is 1. The logsize is the maximum number of bytes to be logged in the metric monitor log file. You can specify either a positive number greater than zero, or unlimited. The default logsize is 1 MB.

quiesce
Specify no more connections to be sent to a server except subsequent new connections from the client to the quiesced server if the connection is designated as sticky and stickytime has not expired. The manager sets the weight for that server to 0 in every port to which it is defined. Use this command if you want to do some quick maintenance on a server and then unquiesce it. If you delete a quiesced server from the configuration and then add it back, it will not retain its status prior to being quiesced. For more information, see "Quiesce server connection handling" on page 180.

server
The IP address of the server as either a symbolic name or in dotted decimal format.

Or, if you used server partitioning, use the logical server's unique name. See "Server Partitioning: logical servers configured to one physical server (IP address)" on page 43 for more information.

now
Only use quiesce "now" if you have stickytime set and you want new connections sent to another server (other than the quiesced server) before stickytime expires. For more information, see "Quiesce server connection handling" on page 180.

reach set
Sets the interval, loglevel, and logsize for the reach advisor.

refresh
Set the number of intervals before querying the executor for a refresh of information about new and active connections.

refresh cycle
A positive number representing the number of intervals. The default is 2.

report
Display a statistics snapshot report.

cluster
The address of the cluster you want displayed in the report. The address can be either a symbolic name or in IP address format. The default is a manager report display for all the clusters.

Note: Additional clusters are separated by a plus sign (+).

restart
Restart all servers (that are not down) to normalized weights (1/2 of maximum weight).

message
A message that you want written to the manager log file.

sensitivity
Set minimum sensitivity to which weights update. This setting defines when the manager should change its weighting for the server based on external information.

weight
A number from 1 to 100 to be used as the weight percentage. The default of 5 creates a minimum sensitivity of 5%.

smoothing
Set an index that smooths the variations in weight when load balancing. A higher smoothing index will cause server weights to change less drastically as network conditions change. A lower index will cause server weights to change more drastically.

index
A positive floating point number. The default is 1.5.

start
Start the manager.

log file
File name to which the manager data is logged. Each record in the log is time stamped.

The default file is installed in the **logs** directory. See Appendix C, "Sample configuration files," on page 413. To change the directory where the log files are kept, see "Changing the log file paths" on page 219.

metric_port
Port that Metric Server will use to report system loads. If you specify a metric port, you must specify a log file name. The default metric port is 10004.

status
Display the current status of all the values in the manager that can be set globally and their defaults.

stop
Stop the manager.

unquiesce
Specify that the manager can begin to give a weight higher than 0 to a server that was previously quiesced, in every port to which it is defined.

server
The IP address of the server as either a symbolic name or in dotted decimal format.

version
Display the current version of the manager.

Examples

- To set the updating interval for the manager to every 5 seconds:
  ```
  dscontrol manager interval 5
  ```
- To set the level of logging to 0 for better performance:
  ```
  dscontrol manager loglevel 0
  ```
- To set the manager log size to 1,000,000 bytes:
  ```
  dscontrol manager logsize 1000000
  ```
- To specify that no more connections be sent to the server at 130.40.52.153:
  ```
  dscontrol manager quiesce 130.40.52.153
  ```

- To set the number of updating intervals before the weights are refreshed to 3:
  ```
  dscontrol manager refresh 3
  ```
- To get a statistics snapshot of the manager:
  ```
  dscontrol manager report
  ```

  This command produces output similar to:

```
------------------------------------------------------------------
|       SERVER                    |    IP ADDRESS   |   STATUS   |
------------------------------------------------------------------
|                   mach14.dmz.com |      10.6.21.14 |     ACTIVE |
|                   mach15.dmz.com |      10.6.21.15 |     ACTIVE |
------------------------------------------------------------------
----------------------------
|   MANAGER REPORT LEGEND   |
----------------------------
| ACTV | Active Connections |
| NEWC | New Connections    |
| SYS  | System Metric      |
| NOW  | Current Weight     |
| NEW  | New Weight         |
| WT   | Weight             |
| CONN | Connections        |
----------------------------
------------------------------------------------------------------
|     www.dmz.com     |        |        |        |        |       |
|     10.6.21.100     | WEIGHT |  ACTV  |  NEWC  |  PORT  |  SYS  |
|     PORT:    21     |NOW  NEW|   49%  |   50%  |    1%  |   0%  |
------------------------------------------------------------------
|     mach14.dmz.com | 10  10 |      0 |      0 |     -1 |      0 |
|     mach15.dmz.com | 10  10 |      0 |      0 |     -1 |      0 |
------------------------------------------------------------------
------------------------------------------------------------------
|     www.dmz.com     |        |        |        |        |       |
|     10.6.21.100     | WEIGHT |  ACTV  |  NEWC  |  PORT  |  SYS  |
|     PORT:    80     |NOW  NEW|   49%  |   50%  |    1%  |   0%  |
------------------------------------------------------------------
|     mach14.dmz.com | 10  10 |      0 |      0 |     23 |      0 |
|     mach15.dmz.com |  9  9  |      0 |      0 |     30 |      0 |
------------------------------------------------------------------
--------------------------------------------------
|    ADVISOR    |    CLUSTER:PORT     |  TIMEOUT  |
--------------------------------------------------
|          http |                  80 | unlimited |
|           ftp |                  21 | unlimited |
--------------------------------------------------
```

- To restart all the servers to normalized weights and write a message to the manager log file:
  ```
  dscontrol manager restart Restarting the manager to update code
  ```

  This command produces output similar to:
  ```
  320-14:04:54 Restarting the manager to update code
  ```
- To set the sensitivity to weight changes to 10:
  ```
  dscontrol manager sensitivity 10
  ```
- To set the smoothing index to 2.0:
  ```
  dscontrol manager smoothing 2.0
  ```
- To start the manager and specify the log file named ndmgr.log (paths cannot be set)
  ```
  dscontrol manager start ndmgr.log
  ```

- To display the current status of the values associated with the manager:

  ```
  dscontrol manager status
  ```

  This command produces output similar to the following example.

  ```
  Manager status:
  ===============
  Metric port................................. 10004
  Manager log filename........................ manager.log
  Manager log level........................... 1
  Maximum manager log size (bytes)............ unlimited
  Sensitivity level........................... 0.05
  Smoothing index............................. 1.5
  Update interval (seconds)................... 2
  Weights refresh cycle....................... 2
  Reach log level............................. 1
  Maximum reach log size (bytes).............. unlimited
  Reach update interval (seconds)............. 7
  Metric monitor log file name................ MetricMonitor.log
  Metric monitor log level.................... 1
  Maximum metric monitor log size............. 1048576
  ```

- To stop the manager:

  ```
  dscontrol manager stop
  ```

- To specify that no more new connections be sent to a server at 130.40.52.153. (Note: Only quiesce the server “now" if you have stickytime set and you want new connections sent to another server before stickytime expires.):

  ```
  dscontrol manager quiesce 130.40.52.153 now
  ```

- To specify that no more new connections be sent to a server at 130.40.52.153. (Note: If you have stickytime set, subsequent new connections from the client are sent to this server until stickytime expires.):

  ```
  dscontrol manager quiesce 130.40.52.153
  ```

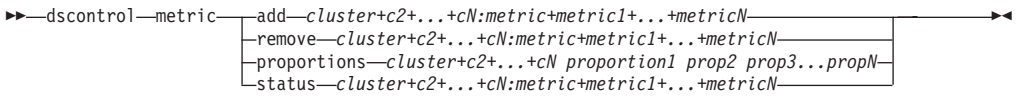
- To specify that the manager can begin to give a weight higher than 0 to a server at 130.40.52.153 that was previously quiesced:

  ```
  dscontrol manager unquiesce 130.40.52.153
  ```

- To display the current version number of the manager:

  ```
  dscontrol manager version
  ```

dscontrol metric — configure system metrics



add
Add the specified metric.

cluster
The address to which clients connect. The address can be either the host name of the machine, or the IP address notation format. Additional clusters are separated by a plus sign (+).

metric
The system metric name. This must be the name of an executable or script file in the metric server's script directory.

remove
Remove the specified metric.

proportions
Set the proportions for all the metrics associated with this object.

status
Display the current values of this metric.

Examples

- To add a system metric:

  ```
  dscontrol metric add site1:metric1
  ```

- To set proportions for a sitename with two system metrics:

  ```
  dscontrol metric proportions site1 0 100
  ```

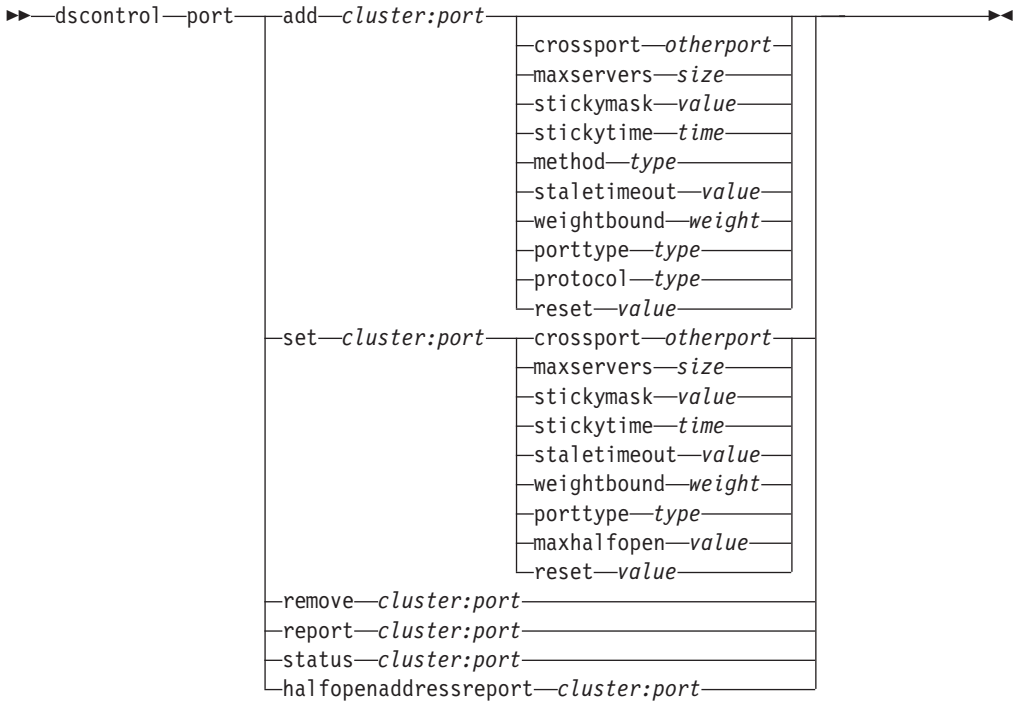
- To display the current status of values associated with the specified metric:

  ```
  dscontrol metric status site1:metric1
  ```

  This command produces output similar to the following:

  ```
  Metric Status:
  ------------
  Cluster ....................... 10.10.10.20
  Metric name ................... metric1
  Metric proportion ............. 50
      Server .................... plm3
      Metric data ............... -1
  ```

dscontrol port — configure ports



add
Add a port to a cluster. You must add a port to a cluster before you can add any servers to that port. If there are no ports for a cluster, all client requests are processed locally. You can add more than one port at one time using this command.

cluster
The address of the cluster as either a symbolic name or in IP address format. You can use a colon (:) to act as a wild card. For instance, the following command, `dscontrol port add :80`, will result in adding port 80 to all clusters.

Note: Additional clusters are separated by a plus sign (+).

port
The number of the port. A port number value of 0 (zero) can be used to specify a wildcard port.

Note: Additional ports are separated by a plus sign (+).

crossport
Crossport allows you to expand the sticky/affinity feature across multiple ports so that client requests received on different ports can still be sent to the same server for subsequent requests. For crossport value, specify the *otherport* number for which you want to share the cross port affinity feature. In order to use this feature, the ports must:

- share the same cluster address
- share the same servers
- have the same (nonzero) stickytime value
- have the same stickymask value

To remove the crossport feature, set the crossport value back to its own port number. For more information on cross port affinity feature, see "Cross port affinity" on page 179.

Note: Crossport only applies to the Dispatcher component's MAC and NAT/NATP forwarding methods.

otherport
The value of crossport. The default value is the same as its own *port* number.

maxservers
The maximum number of servers. The default value of maxservers is 32.

size
The value of maxservers.

stickymask
The affinity address mask feature groups incoming client requests based on common subnet addresses. When a client request first makes a connection to the port, all subsequent requests from clients with the same subnet address (designated by that part of the IP address which is being masked) are directed to the same server. In order to enable stickymask, port stickytime must be a nonzero value. See "Affinity address mask (stickymask)" on page 179 for more information.

Note: The stickymask keyword only applies to the Dispatcher component.

value
The stickymask value is the number of high-order bits of the 32-bit IP address you want to mask. Possible values are: 8, 16, 24, and 32. The default value is 32, which disables the affinity address mask feature.

stickytime
The interval between the closing of one connection and the opening of a new connection during which a client will be sent back to the same server used during the first connection. After the sticky time, the client may be sent to a server different from the first.

For the Dispatcher component:

- For Dispatcher's cbr forwarding method
  - You can only set stickytime (to a nonzero value) on an SSL (not HTTP) port because setting stickytime enables SSL ID affinity.
  - If you set the port stickytime, then the affinity type on the rule must be none (default). Rule-based affinity (passive cookie, URI) cannot co-exist when stickytime is set on the port.
- For Dispatcher's mac and nat forwarding methods
  - If you set the port stickytime (to a nonzero value), then you cannot set an affinity type on the rule. Rule-based affinity cannot co-exist when stickytime is set on the port.
  - Setting a port stickytime value enables IP address affinity.

For the CBR component: If you set the port stickytime to a nonzero value, then the affinity type on the rule must be none (default). Rule-based affinity (passive cookie, URI, active cookie) cannot co-exist when stickytime is set on the port.

time
The port sticky time in number of seconds. Zero signifies that the port is not sticky.

method
The forwarding method. Possible forwarding methods are: mac forwarding, nat forwarding, or content-based routing (cbr) forwarding. You may *not* add a forwarding method of nat or cbr unless you first specify a nonzero IP address in the clientgateway parameter of the dscontrol executor command. See “Dispatcher's NAT/NAPT (nat forwarding method)” on page 39 and “Dispatcher's content-based routing (cbr forwarding method)” on page 41 for more information.

Notes:

1. Method only applies to the Dispatcher component.
2. If the backend server is on the same subnet as the return address, and if you are using the cbr forwarding method or the nat forwarding method, you must define the router address to be the backend server address.
3. If you add a mac forwarding method, then you are required to specify the "protocol" parameter as either HTTP or SSL.

type
The forwarding method type. Possible values are: mac, nat, or cbr. The default is mac forwarding.

staletimeout
The number of seconds during which there can be no activity on a connection before that connection is removed. For the Dispatcher component, the default value is 900 for port 21 (FTP) and 259,200 for port 23 (Telnet). For all other Dispatcher ports and for all CBR ports, the default is 300. Staletimeout can also be set at the executor or cluster level. See “Using stale timeout value” on page 220 for more information.

value
The value of **staletimeout** in number of seconds.

weightbound
Set the maximum weight for servers on this port. This affects how much difference there can be between the number of requests the executor will give each server. The default value is 20.

weight
A number from 1–100 representing the maximum weight bound.

porttype
The port type.

Note: Porttype only applies to Dispatcher.

type
Possible values are **tcp**, **udp**, and **both**. The default value is both (tcp/udp).

protocol
The protocol type. For the Dispatcher component, this is a required parameter when specifying a "cbr" method on the port. If you select a port protocol type **SSL**, you should also specify a nonzero stickytime to enable SSL ID affinity. If you select **HTTP** protocol, you can establish server affinity using "content" rules. See “Dispatcher's content-based routing (cbr forwarding method)” on page 41 for more information.

Note: Protocol only applies to Dispatcher's cbr forwarding method.

type
Possible values are **HTTP** or **SSL**.

maxhalfopen
The threshold for maximum half-open connections. Use this parameter to detect possible denial of service attacks that result in a large number of half-opened TCP connections on servers.

A positive value indicates that a check is made to determine if the current half-open connections exceeds the threshold. If the current value is above the threshold, a call to an alert script is made. See "Denial of service attack detection" on page 193 for more information.

Note: maxhalfopen only applies to Dispatcher.

value
The value of maxhalfopen. The default is zero (no checking will be made).

reset
Reset allows you to specify whether Load Balancer will send TCP resets to down servers on the port. A TCP reset causes the connection to be immediately closed. See "Sending TCP reset to a down server (Dispatcher component only)" on page 144 for more information.

Note: Reset only applies to the Dispatcher component. The `clientgateway` on the `dscontrol executor` command must be set to a router address in order to use the reset keyword.

value
Possible values for reset are yes and no. The default is no (no TCP resets are made to down servers). When reset is yes, TCP resets are sent to down servers.

set
Set the fields of a port.

remove
Remove this port.

report
Report on this port.

status
Show status of servers on this port. If you want to see the status on all ports, do not specify a *port* with this command. Do not forget the colon, however.

numSeconds
The amount of time in seconds before resetting half-open connections.

halfopenaddressreport
Generates entries in the log (halfOpen.log) for all the client addresses (up to approximately 8000 address pairs) that have accessed servers that have any half open connnections. Also, statistical data is reported back to the command line, such as: total, largest, and average number of half-open connections, and the average half-open connection time (in seconds). See "Denial of service attack detection" on page 193 for more information.

Examples

- To add port 80 and 23 to a cluster address 130.40.52.153:
  ```
  dscontrol port add 130.40.52.153:80+23
  ```
- To add a wildcard port to a cluster address of 130.40.52.153:
  ```
  dscontrol port set 130.40.52.153:0
  ```
- To set the maximum weight of 10 to port 80 at a cluster address of 130.40.52.153:

```
dscontrol port set 130.40.52.153:80  weightbound 10
```

- To set the stickytime value to 60 seconds for port 80 and port 23 at a cluster address of 130.40.52.153:

  ```
  dscontrol port set 130.40.52.153:80+23  stickytime 60
  ```

- To set the cross port affinity of port 80 to port 23 at a cluster address of 130.40.52.153:

  ```
  dscontrol port set 130.40.52.153:80  crossport 23
  ```

- To remove port 23 from a cluster address of 130.40.52.153:

  ```
  dscontrol port remove 130.40.52.153:23
  ```

- To get the status of port 80 at a cluster address of 9.67.131.153:

  ```
  dscontrol port status 9.67.131.153:80
  ```

  This command produces output similar to:

  ```
  Port Status:
  ------------
  Port number ..................... 80
  Cluster ......................... 9.67.131.153
  Stale timeout ................... 300
  Weight bound .................... 20
  Maximum number of servers ...... 32
  Sticky time ..................... 0
  Port type ....................... tcp/udp
  Cross Port Affinity ............. 80
  Sticky mask bits ................ 32
  Max Half Open Connections ...... 0
  Send TCP Resets ................. no
  ```

- To get the report of port 80 at a cluster address of 9.62.130.157:

  ```
  dscontrol port report 9.62.130.157:80
  ```

  This command produces output similar to:

  ```
  Port Report:
  ------------
  Cluster address ................. 9.62.130.157
  Port number ..................... 80
  Number of servers ............... 5
  Maximum server weight .......... 10
  Total active connections ....... 55
  Connections per second ......... 12
  KBytes per second ............... 298
  Number half open ................ 0
  TCP Resets sent ................. 0
  Forwarding method ............... MAC Based Forwarding
  ```

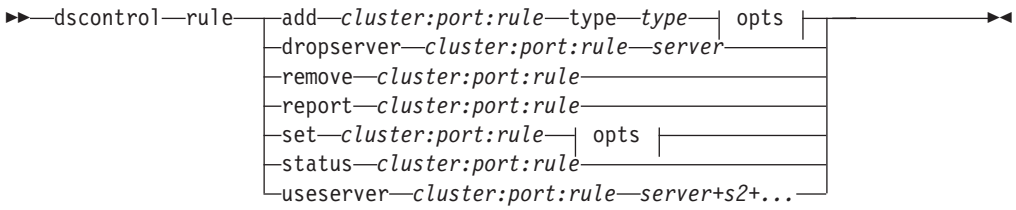
- To get the half open address report for port 80 at a cluster address of 9.67.127.121:

  ```
  dscontrol port halfopenaddressreport 9.67.127.121:80
  ```

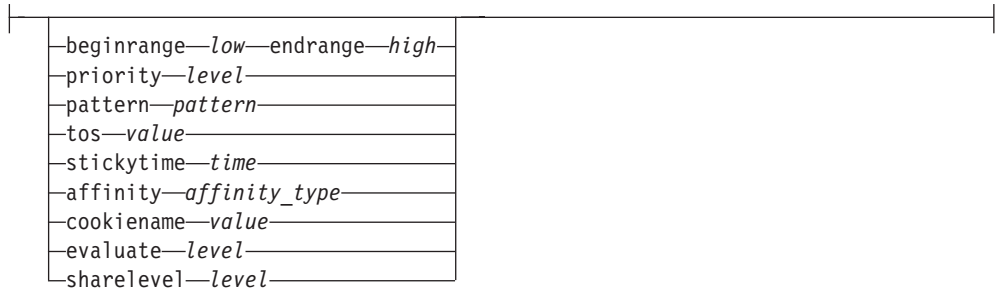
  This command produces output similar to:

  ```
  Half open connection report successfully created:
  ------------
  Half Open Address Report for cluster:port = 9.67.127.121:80
  Total addresses with half open connections reported ... 0
  Total number of half open connections reported ........ 0
  Largest number of half open connections reported ...... 0
  Average number of half open connections reported ...... 0
  Average half open connection time (seconds) reported .. 0
  Total half open connections received .................. 0
  ```

dscontrol rule — configure rules



opts:



add
Add this rule to a port.

cluster
The address of the cluster as either a symbolic name or in IP address format. You can use a colon (:) to act as a wild card. For instance, the following command, `dscontrol rule add :80:RuleA type` *type*, will result in adding RuleA to port 80 for all clusters.

Note: Additional clusters are separated by a plus sign (+).

port
The number of the port. You can use a colon (:) to act as a wild card. For instance, the following command, `dscontrol rule add clusterA::RuleA type` *type*, will result in adding RuleA to all ports for ClusterA.

Note: Additional ports are separated by a plus sign (+).

rule
The name you choose for the rule. This name can contain any alphanumeric character, underscore, hyphen, or period. It can be from 1 to 20 characters and cannot contain any blanks.

Note: Additional rules are separated by a plus sign (+).

type
The type of rule.

type
Your choices for *type* are:

ip The rule is based on the client IP address.

time The rule is based on the time of day.

connection
The rule is based on the number of connections per second for the port. This rule will work only if the manager is running.

active The rule is based on the number of active connections total for the port. This rule will work only if the manager is running.

port The rule is based on the client port.

Note: Port applies to the Dispatcher component.

service
This rule is based on the type of service (TOS) byte field in the IP header.

Note: Service only applies to the Dispatcher component.

reservedbandwidth
This rule is based on the bandwidth (kilobytes per second) being delivered by a set of servers. For more information, see "Using rules based on reserved bandwidth and shared bandwidth" on page 173 and "Reserved bandwidth rule" on page 173.

Note: Reservedbandwidth only applies to the Dispatcher component.

sharedbandwidth
This rule is based on the amount of bandwidth (kilobytes per second) that is shared at the executor or cluster level. For more information, see "Using rules based on reserved bandwidth and shared bandwidth" on page 173 and "Shared bandwidth rule" on page 174.

Note: Sharedbandwidth only applies to the Dispatcher component.

true This rule is always true. Think of it as an else statement in programming logic.

content
This rule describes a regular expression that will be compared to the client requested URLs. This is valid for Dispatcher and CBR.

beginrange
The lower value in the range used to determine whether or not the rule is true.

low
Depends on the type of rule. The kind of value and its default are listed here by the type of rule:

ip The address of the client as either a symbolic name or in IP address format. The default is 0.0.0.0.

time An integer. The default is 0, representing midnight.

connection
An integer. The default is 0.

active An integer. The default is 0.

port An integer. The default is 0.

reservedbandwidth
An integer (kilobytes per second). The default is 0.

endrange
The higher value in the range used to determine whether or not the rule is true.

high
Depends on the type of rule. The kind of value and its default are listed here by the type of rule:

ip The address of the client as either a symbolic name or in IP address format. The default is 255.255.255.254.

time An integer. The default is 24, representing midnight.

Note: When defining the beginrange and endrange of time intervals, note that each value must be an integer representing only the hour portion of the time; portions of an hour are not specified. For this reason, to specify a single hour—say, the hour between 3:00 and 4:00 am— you would specify a beginrange of **3** and an endrange also of **3**. This will signify all the minutes beginning with 3:00 and ending with 3:59. Specifying a beginrange of **3** and an endrange of **4** would cover the two-hour period from 3:00 through 4:59.

connections
An integer. The default is 2 to the 32nd power minus 1.

active An integer. The default is 2 to the 32nd power minus 1.

port An integer. The default is 65535.

reservedbandwidth
An integer (kilobytes per second). The default is 2 to the 32nd power minus 1.

priority
The order in which the rules are reviewed.

level
An integer. If you do not specify the priority of the first rule you add, Dispatcher will set it by default to 1. When a subsequent rule is added, by default its priority is calculated to be 10 + the current lowest priority of any existing rule. For example, assume you have an existing rule whose priority is 30. You add a new rule and set its priority at 25 (which, remember, is a *higher* priority than 30). Then you add a third rule without setting a priority. The priority of the third rule is calculated to be 40 (30 + 10).

pattern
Specifies the pattern to be used for a content type rule.

pattern
The pattern to be used. For more information on valid values, see Appendix B, "Content rule (pattern) syntax," on page 409.

tos
Specifies the "type of service" (TOS) value used for the **service** type rule.

Note: TOS only applies to the Dispatcher component.

value
The 8 character string to be used for the tos value, where valid characters are:

0 (binary zero), 1 (binary one), and x (do not care). For example: 0xx1010x. For more information, see “Using rules based on type of service (TOS)” on page 172.

stickytime
Specifies the stickytime to be used for a rule. When setting the affinity parameter to "activecookie" on the rule command, stickytime should be set to a nonzero value to enable this affinity type. Stickytime on the rule does not apply to "passivecookie" or "uri" affinity rule types.

See “Active cookie affinity” on page 181 for more information.

Note: Rule stickytime only applies to the CBR component.

time
Time in seconds.

affinity
Specifies the affinity type to be used for a rule: active cookie, passive cookie, URI, or none.

An affinity type of "activecookie" enables load-balancing Web traffic with affinity to the same server based upon cookies generated by Load Balancer.

An affinity type of "passivecookie" enables load-balancing Web traffic with affinity to the same server based upon self-identifying cookies generated by the servers. You must use the cookiename parameter in conjunction with passive cookie affinity.

An affinity type of "URI" enables load-balancing Web traffic to caching-proxy servers in a manner which effectively increases the size of the cache.

See “Active cookie affinity” on page 181, “Passive cookie affinity” on page 183, and “URI affinity” on page 184 for more information.

Note: Affinity applies to rules configured with the Dispatcher component's cbr forwarding method and to the CBR component.

affinity_type
Possible values for affinity type are: none (default), activecookie, passivecookie, or uri.

cookiename
An arbitrary name set by the administrator that acts as an identifier to Load Balancer. It is the name that Load Balancer should look for in the client HTTP header request. The cookie name, along with the cookie value, acts as an identifier to Load Balancer allowing Load Balancer to send subsequent requests of a Web site to the same server machine. Cookie name is only applicable with "passive cookie" affinity.

See “Passive cookie affinity” on page 183 for more information.

Note: Cookie name applies to rules configured with the Dispatcher component's cbr forwarding method and to the CBR component.

value
The cookie name value.

evaluate
This option is available only in the Dispatcher component. Specifies whether to evaluate the rule's condition across all servers within the port or across servers within the rule. This option is only valid for rules that make their decisions

based upon the characteristics of the servers, such as: connection, active, and reservedbandwidth rules. For more information, see “Server evaluation option for rules” on page 177.

For the connection type rule, you can also specify an evaluate option — upserversonrule. By specifying upserversonrule, you can ensure that the remaining servers within the rule will not be overloaded if some of the servers in the server-set are down.

level
Possible values are port, rule, or upserversonrule. The default is port. upserversonrule is only available for the connection type rule.

sharelevel
This parameter is only for the shared bandwidth rule. Specifies whether to share bandwidth at the cluster level or executor level. Sharing bandwidth at the cluster level allows a port (or ports) to share a maximum amount of bandwidth across several ports within the same cluster. Sharing bandwidth at the executor level allows a cluster (or clusters) within the entire Dispatcher configuration to share a maximum amount of bandwidth. For more information see “Shared bandwidth rule” on page 174.

level
Possible values are executor or cluster.

dropserver
Remove a server from a rule set.

server
The IP address of the TCP server machine as either a symbolic name or in IP address format.

Or, if you used server partitioning, use the logical server's unique name. See “Server Partitioning: logical servers configured to one physical server (IP address)” on page 43 for more information.

Note: Additional servers are separated by a plus sign (+).

remove
Remove one or more rules, separated from one another by plus signs.

report
Display the internal values of one or more rules.

set
Set values for this rule.

status
Display the settable values of one or more rules.

useserver
Insert servers into a rule set.

Examples

- To add a rule that will always be true, do not specify the beginning range or end range:

  ```
  dscontrol rule add 9.37.67.100:80:trule type true priority 100
  ```

- To create a rule forbidding access to a range of IP addresses, in this case those beginning with “9:”

  ```
  dscontrol rule add 9.37.131.153:80:ni type ip b 9.0.0.0 e 9.255.255.255
  ```

- To create a rule that will specify the use of a given server from the hour of 11:00 a.m. through the hour of 3:00 p.m.:

  ```
  dscontrol rule add cluster1:80:timerule type time beginrange 11 endrange 14
  dscontrol rule useserver cluster1:80:timerule server05
  ```

- To create a rule based on the content of the TOS byte field in the IP header:

  ```
  dscontrol rule add 9.67.131.153:80:tosrule type service tos 0xx1001x
  ```

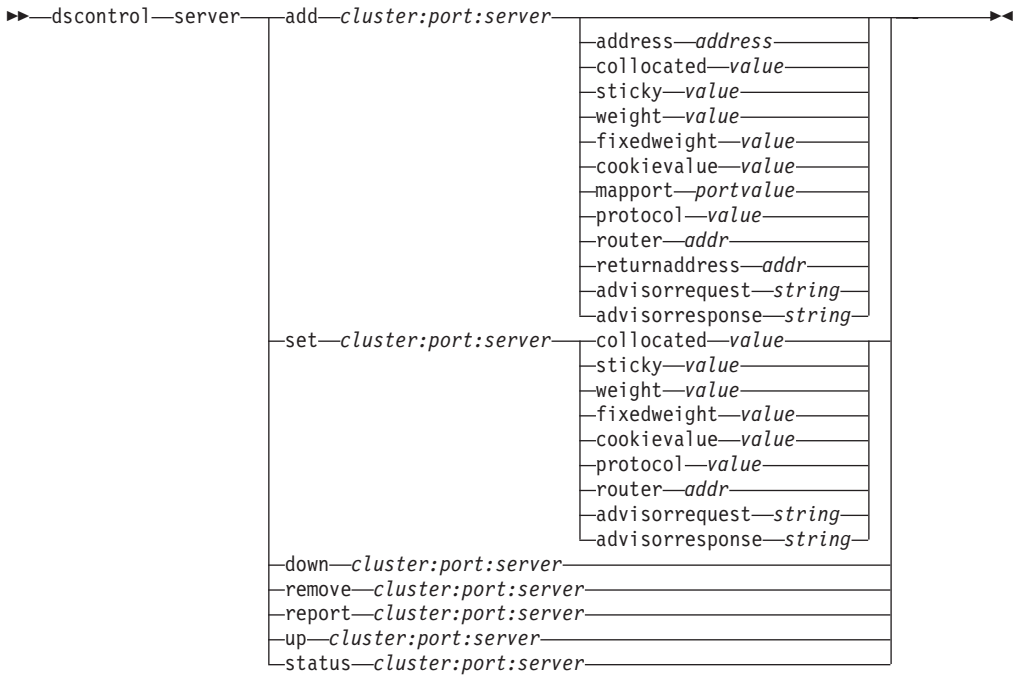
- To create a rule based on reserved bandwidth that will allocate a set of servers (evaluated within the rule) to deliver data up to a rate of 100 kilobytes per second:

  ```
  dscontrol rule add 9.67.131.153:80:rbwrule type reservedbandwidth
  beginrange 0 endrange 100 evaluate rule
  ```

- To create a rule based on shared bandwidth that will recruit unused bandwidth at the cluster level. (Note: You must first specify the maximum amount of bandwidth (kilobytes per second) that can be shared at the cluster level using the dscontrol cluster command):

  ```
  dscontrol cluster set 9.67.131.153 sharedbandwidth 200

  dscontrol rule add 9.67.131.153:80:shbwrule type sharedbandwidth
  sharelevel cluster
  ```

dscontrol server — configure servers



add
Add this server.

cluster
The address of the cluster as either a symbolic name or in IP address format. You can use a colon (:) to act as a wild card. For instance, the following command, `dscontrol server add :80:ServerA`, will result in adding ServerA to port 80 on all clusters.

Note: Additional clusters are separated by a plus sign (+).

port
The number of the port. You can use a colon (:) to act as a wild card. For instance, the following command, `dscontrol server add ::ServerA`, will result in adding ServerA to all clusters on all ports.

Note: Additional ports are separated by a plus sign (+).

server
The **server** is the unique IP address of the TCP server machine as either a symbolic name or in IP address format.

Or, if you use a unique name that does not resolve to an IP address, you must provide the server **address** parameter on the **dscontrol server add** command. See "Server Partitioning: logical servers configured to one physical server (IP address)" on page 43 for more information.

Note: Additional servers are separated by a plus sign (+).

address
The unique IP address of the TCP server machine as either a host name or in IP address format. If the server is unresolvable, you must provide the address

of the physical server machine. See “Server Partitioning: logical servers configured to one physical server (IP address)” on page 43 for more information.

address
Value of the address of the server.

collocated
Collocated allows you to specify if the Dispatcher is installed on one of the server machines it is load balancing.

Note: Be aware of the following:

- Collocated parameter is valid when using the Dispatcher's mac, nat, or cbr forwarding methods. Site Selector and CBR can be collocated on all platforms but do not require this keyword. For more information, see “Using collocated servers” on page 162.
- **Windows** systems: Collocation is not available.

value
Value of collocated: yes or no. Default is no.

sticky
Allows a server to override the stickytime setting on its port. With a default value of “yes," the server retains the normal affinity as defined at the port. With a value of “no," the client will *not* return to that server the next time it issues a request on that port regardless of the stickytime setting of the port. This is useful in certain situations when you are using rules. For more information, see “port affinity override” on page 176.

value
Value of sticky: yes or no. Default is yes.

weight
A number from 0–100 (but not to exceed the specified port's weightbound value) representing the weight for this server. Setting the weight to zero will prevent any new requests from being sent to the server, but will not end any currently active connections to that server. The default is one-half the specified port’s maximum weightbound value. If the manager is running, this setting will be quickly overwritten.

value
Value of the server weight.

fixedweight
The fixedweight option allows you to specify whether you want the manager to modify the server weight or not. If you set the fixedweight value to yes, when the manager runs it will not be allowed to modify the server weight. For more information, see “Manager fixed weights” on page 144.

value
Value of fixedweight: yes or no. Default is no.

cookievalue
Cookievalue is an arbitrary value that represents the server side of the cookie name/ cookie value pair. The cookie value, along with the cookie name, acts as an identifier allowing Load Balancer to send subsequent client requests to the same server. See “Passive cookie affinity” on page 183 for more information.

Note: Cookievalue is valid for Dispatcher (using cbr forwarding method) and CBR.

value
Value is any arbitrary value. Default is no cookie value.

mapport
Map the client request's destination port number (which is for Dispatcher) to the server's port number that Dispatcher uses to load balance the client's request. Allows Load Balancer to receive a client's request on one port and to transmit it to a different port on the server machine. With mapport you can load balance a client's requests to a server that may have multiple server daemons running.

Note: Mapport applies to Dispatcher (using nat or cbr forwarding methods) and to CBR. For Dispatcher, see "Dispatcher's NAT/NAPT (nat forwarding method)" on page 39 and "Dispatcher's content-based routing (cbr forwarding method)" on page 41. For CBR, see "Load balancing client-to-proxy in SSL and proxy-to-server in HTTP" on page 73.

protocol
The valid values for protocol are HTTP and HTTPS. The default is HTTP.

Note: Protocol only applies to the CBR component.

portvalue
Value of the map port number. The default is the client request's destination port number.

router
If you are setting up a wide area network, the address of the router to the remote server. Default is 0, indicating a local server. Note that when a server's router address is set to something other than zero (indicating a remote server), it cannot be reset to 0 to make the server local again. Instead, the server must be removed, then added again without a router address being a specified. Similarly, a server defined as local (router address = 0) cannot be made remote by changing the router address. The server must be removed and added again. See "Configure wide area Dispatcher support" on page 184 for more information.

Note: Router only applies to Dispatcher. If you are using nat or cbr forwarding methods, when you add a server to the configuration you must specify the router address.

addr
Value of the address of the router.

returnaddress
A unique IP address or hostname. It is an address configured on the Dispatcher machine that Dispatcher uses as its source address when load balancing the client's request to the server. This ensures that the server will return the packet to the Dispatcher machine in order to process the content of the request, rather than sending the packet directly to the client. (Dispatcher will then forward the IP packet on to the client.) You must specify the return address value when the server is added. Return address cannot be changed unless you remove the server and add it again. The return address cannot be the same as the cluster, server, or NFA address.

Note: Returnaddress only applies to Dispatcher. When you use nat or cbr forwarding methods, you must define a return address for communication between Load Balancer and the backend servers. The

number of connections that Load Balancer can keep active with the backend server is limited by the number of return addresses that are defined. Load Balancer uses ports that are based upon the return address only; not the return address and server combination. When all the available ports are in use, additional connections fail. In a busy environment, use multiple return addresses to prevent a shortage of available ports.

addr
Value of the return address.

advisorrequest
The HTTP or HTTPS advisor uses the advisor request string to query the health of the servers. It will only be valid for servers which are advised upon by the HTTP or HTTPS advisor. You must start the HTTP or HTTPS advisor in order for this value to be enabled. See "Configuring the HTTP or HTTPS advisor using the request and response (URL) option" on page 152 for more information.

Note: The advisorrequest applies to Dispatcher and CBR components.

string
Value of the string used by the HTTP or HTTPS advisor. The default is `HEAD / HTTP/1.0`.

Note: If a blank is contained within the string —

- When issuing the command from the **dscontrol>>** shell prompt, you must place quotes around the string. For example: **server set** *cluster:port:server* **advisorrequest** `"head / http/1.0"`
- When issuing the **dscontrol** command from the operating system prompt, you must precede the text with **"\"** and follow the text with **\""**. For example: **dscontrol server set** *cluster:port:server* **advisorrequest** `"\"head / http/1.0\""`

advisorresponse
The advisor response string that the HTTP or HTTPS advisor scans for in the HTTP response. It will only be valid for servers that are advised upon by the HTTP or HTTPS advisor. You must start the HTTP or HTTPS advisor in order for this value to be enabled. See "Configuring the HTTP or HTTPS advisor using the request and response (URL) option" on page 152 for more information.

Note: The advisorresponse applies to Dispatcher and CBR components.

string
Value of the string used by the HTTP or HTTPS advisor. The default is null.

Note: If a blank is contained within the string —

- When issuing the command from the **dscontrol>>** shell prompt, you must place quotes around the string.
- When issuing the **dscontrol** command from the operating system prompt, you must precede the text with **"\"** and follow the text with **\""**.

down
Mark this server down. This command breaks all active connections to that server and prevents any other connections or packets from being sent to that server.

When the server down command is used to bring a server offline, if the stickytime value is nonzero for that server, existing clients continue to be served by that server until stickytime expires. The server is not taken down until after the stickytime value expires.

remove
Remove this server.

report
Report on this server. The report contains the following information per server: current number of connections per second (CPS), kilobytes transferred in a one second interval (KBPS), total number of connections (Total), number of connections that are in the active state (Active), number of connections that are in the FIN state (FINed), and number of completed connections (Comp).

set
Set values for this server.

status
Show status of the servers.

up Mark this server up. Dispatcher will now send new connections to that server.

Examples

- To add the server at 27.65.89.42 to port 80 on a cluster address 130.40.52.153:

  ```
  dscontrol server add 130.40.52.153:80:27.65.89.42
  ```

- To set the server at 27.65.89.42 as nonsticky (port affinity override feature):

  ```
  dscontrol server set 130.40.52.153:80:27.65.89.42 sticky no
  ```

- To mark the server at 27.65.89.42 as down:

  ```
  dscontrol server down 130.40.52.153:80:27.65.89.42
  ```

- To remove the server at 27.65.89.42 on all ports on all clusters:

  ```
  dscontrol server remove ::27.65.89.42
  ```

- To set the server at 27.65.89.42 as collocated (server resides in the same machine as the Load Balancer):

  ```
  dscontrol server set 130.40.52.153:80:27.65.89.42 collocated yes
  ```

- To set the weight to 10 for server 27.65.89.42 at port 80 on cluster address 130.40.52.153:

  ```
  dscontrol server set 130.40.52.153:80:27.65.89.42 weight 10
  ```

- To mark the server at 27.65.89.42 as up:

  ```
  dscontrol server up 130.40.52.153:80:27.65.89.42
  ```

- To add a remote server:

  ```
  dscontrol server add 130.40.52.153:80:130.60.70.1 router 130.140.150.0
  ```

- To allow the HTTP advisor to query an HTTP URL request `HEAD / HTTP/1.0` for server 27.65.89.42 on HTTP port 80:

  ```
  dscontrol server set 130.40.52.153:80:27.65.89.42
  advisorrequest "\"HEAD / HTTP/1.0\""
  ```

- To show the status for server 9.67.143.154 on port 80:

  ```
  dscontrol server status 9.67.131.167:80:9.67.143.154
  ```

  This command produces output similar to:

  ```
  Server Status:
  --------------
  Server ......................... 9.67.143.154
  Port number .................... 80
  Cluster ........................ 9.67.131.167
  ```

```
Cluster address ................ 9.67.131.167
Quiesced ....................... N
Server up ...................... Y
Weight ......................... 10
Fixed weight ................... N
Sticky for rule ................ Y
Remote server .................. N
Network Router address ......... 0.0.0.0
Collocated ..................... N
Advisor request................. HEAD / HTTP/1.0
Advisor response................
Cookie value ................... n/a
Clone ID ....................... n/a
```

dscontrol set — configure server log

loglevel
The level at which the dsserver logs its activities.

level
The default value of **loglevel** is 0. The range is 0–5. The following are the possible values: 0 is None, 1 is Minimal, 2 is Basic, 3 is Moderate, 4 is Advanced, 5 is Verbose.

logsize
The maximum number of bytes to be logged in the log file.

size
The default value of logsize is 1 MB.

dscontrol status — display whether the manager and advisors are running

Examples

- To see what is running:

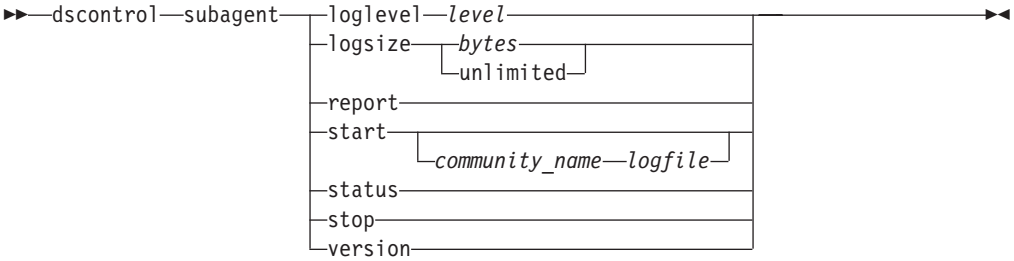
  ```
  dscontrol status
  ```

  This command produces output similar to:

  ```
  Executor has been started.
  Manager has been started.

  --------------------------------------
  | ADVISOR |  CLUSTER:PORT  |  TIMEOUT  |
  --------------------------------------
  | reach   |      0         | unlimited |
  |  http   |     80         | unlimited |
  |  ftp    |     21         | unlimited |
  --------------------------------------
  ```

dscontrol subagent — configure SNMP subagent

Note: dscontrol subagent command syntax diagrams applies to the Dispatcher component.



loglevel
The level at which the subagent logs its activities to a file.

level
The number of the level (0 to 5). The higher the number, the more information that is written to the manager log. The default is 1. The following are the possible values: 0 is None, 1 is Minimal, 2 is Basic, 3 is Moderate, 4 is Advanced, 5 is Verbose.

logsize
Set the maximum size of the bytes to be logged in the subagent log. The default is 1 MB. When you set a maximum size for the log file, the file will wrap; when the file reaches the specified size, the subsequent entries are written from the top of the file, overwriting the previous log entries. Log size cannot be set smaller than the current size of the log. Log entries are time—stamped so you can tell the order in which they were written. The higher you set the log level, the more carefully you should choose the log size, because you can quickly run out of space when logging at the higher levels.

bytes
The maximum size in bytes for the subagent log file. You can specify either a positive number greater than zero, or the word **unlimited**. The log file may not reach the exact maximum size before overwriting because the log entries themselves vary in size. The default value is unlimited.

report
Display a statistics snapshot report.

start
Start the subagent.

community_name
The name of the SNMP value of community name that you can use as a security password. The default is public.

For **Windows platform**: The community name for the operating system is used.

log file
File name to which the SNMP subagent data is logged. Each record in the log is time stamped. The default is subagent.log. The default file is installed in the **logs** directory. See Appendix C, "Sample configuration files," on page 413. To change the directory where the log files are kept, see "Changing the log file paths" on page 219.

status
Display the current status of all the values in the SNMP subagent that can be set globally and their defaults.

version
Display the current version of the subagent.

Examples

- To start the subagent with a community name of bigguy:

```
dscontrol subagent start bigguy bigguy.log
```

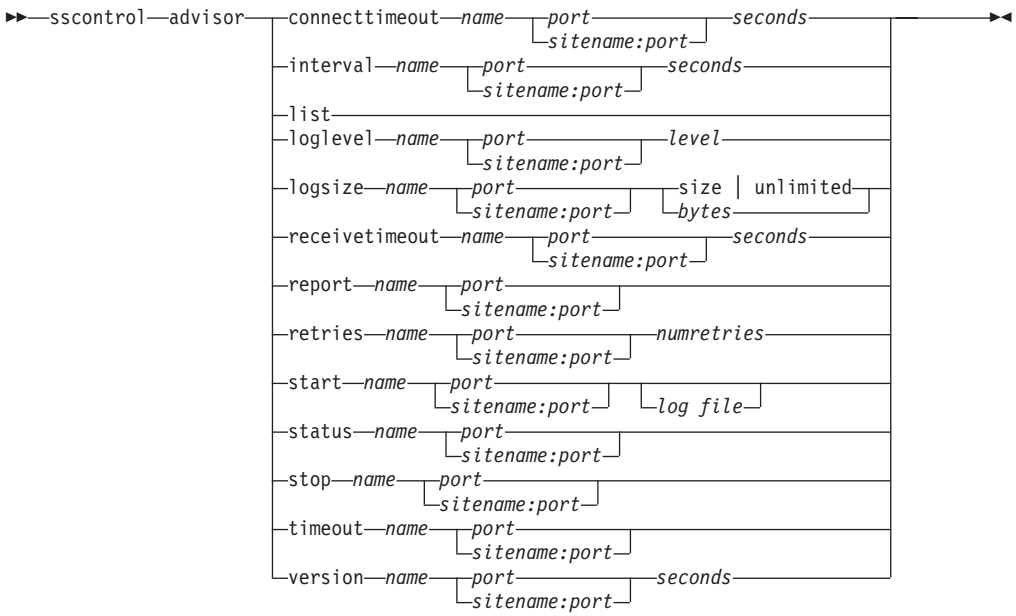
Chapter 27. Command reference for Site Selector

This chapter describes how to use the following Site Selector **sscontrol** commands:

You can enter a minimized version of the sscontrol command parameters. You only need to enter the unique letters of the parameters. For example, to get help on the file save command, you can enter **sscontrol he f** instead of **sscontrol help file**.

Note: The command parameter values must be entered in English characters. The only exceptions are host names (used in cluster and server commands) and file names (used in file commands).

sscontrol advisor — control the advisor



connecttimeout
Set how long an advisor waits before reporting that a connect to a server fails. For more information, see "Advisor connect timeout and receive timeout for servers" on page 149.

name
The name of the advisor. Possible values include **http**, **https**, **ftp**, **sip**, **ssl**, **smtp**, **imap**, **pop3**, **ldap**, **ldaps**, **nntp**, **telnet**, **connect**, **ping**, **WLM**, and **WTE**. Names of customized advisors are of the format xxxx, where ADV_*xxxx* is the name of the class that implements the custom advisor.

port
The number of the port that the advisor is monitoring.

seconds
A positive integer representing the time in seconds that the advisor waits before reporting that a connect to a server has failed. The default is 3 times the value specified for the advisor interval.

interval
Set how often the advisor queries the servers for information.

seconds
A positive integer representing the number of seconds between status requests to the servers. The default is 7.

list
Show list of advisors currently providing information to the manager.

loglevel
Set the logging level for an advisor log.

level
The number of the level (0 to 5). The default is 1. The higher the number, the more information that is written to the advisor log. The possible values are:

- 0 is None

- 1 is Minimal
- 2 is Basic
- 3 is Moderate
- 4 is Advanced
- 5 is Verbose

.

logsize
Set the maximum size of an advisor log. When you set a maximum size for the log file, the file wraps; when the file reaches the specified size, the subsequent entries overwrite the previous log entries. Log size cannot be set smaller than the current size of the log. Log entries are time-stamped so you can tell the order in which they were written. The higher you set the log level, the more carefully you must choose the log size, because you can quickly run out of space when logging at the higher levels.

size | unlimited
The maximum size in bytes for the advisor log file. You can specify either a positive number greater than zero, or **unlimited**. The log file may not reach the exact maximum size before being overwritten because the log entries vary in size. The default value is 1 MB.

receivetimeout
Set how long an advisor waits before reporting that a receive from a server fails. For more information, see "Advisor connect timeout and receive timeout for servers" on page 149.

seconds
A positive integer representing the time in seconds that the advisor waits before reporting that a receive from a server has failed. The default is 3 times the value specified for the advisor interval.

report
Display a report on the state of the advisor.

retries
The number of retries that an advisor can make before marking a server down.

numretries
An integer greater than or equal to zero. This value should be no larger than 3. If retries keyword is not configured, the number of retries defaults to zero.

start
Start the advisor. There are advisors for each protocol. The default ports are:

Advisor Name	Protocol	Port
Connect	n/a	user-defined
db2	private	50000
ftp	FTP	21
http	HTTP	80
https	SSL	443
imap	IMAP	143
ldap	LDAP	389
ldaps	SSL	443
nntp	NNTP	119

Advisor Name	Protocol	Port
PING	PING	N/A
pop3	POP3	110
sip	SIP	5060
smtp	SMTP	25
ssl	SSL	443
telnet	Telnet	23

name
The advisor name.

sitename:port
The sitename value is optional on the advisor commands; however, the port value is required. If the sitename value is not specified, the advisor starts running on all available sitenames configured. If you specify a sitename, the advisor starts running for only the sitename you specify. Additional sitenames are separated by a plus sign (+).

log file
File name to which the management data is logged. Each record in the log is time-stamped.

The default file is *advisorname_port*.log, for example, **http_80.log.** To change the directory where the log files are stored, see "Changing the log file paths" on page 219.

You can start only one advisor for each sitename.

status
Display the current status and defaults of all the global values in an advisor.

stop
Stop the advisor.

timeout
Set the number of seconds that the manager considers information from the advisor as valid. If the manager finds that the advisor information is older than this timeout period, the manager does not use that information in determining weights for the servers on the port the advisor is monitoring. An exception to this timeout is when the advisor has informed the manager that a specific server is down. The manager uses that information about the server, even after the advisor information times out.

seconds
A positive number representing the number of seconds or **unlimited**. The default value is unlimited.

version
Display the current version of the advisor.

Examples

- To set the time (30 seconds) an HTTP advisor (for port 80) waits before reporting that a connect to a server fails:

```
sscontrol advisor connecttimeout http 80 30
```

- To set the interval for the FTP advisor (for port 21) to 6 seconds:

```
sscontrol advisor interval ftp 21 6
```

- To display the list of advisors currently providing information to the manager:

  ```
  sscontrol advisor list
  ```

  This command produces output similar to:

  ```
  --------------------------------------
  | ADVISOR | SITENAME:PORT |  TIMEOUT  |
  --------------------------------------
  |  http   |           80 | unlimited |
  |  ftp    |           21 | unlimited |
  --------------------------------------
  ```

- To change the log level of the http advisor log for the sitename of mysite to 0 for better performance:

  ```
  sscontrol advisor loglevel http mysite:80 0
  ```

- To change the ftp advisor log size for the sitename of mysite to 5000 bytes:

  ```
  sscontrol advisor logsize ftp mysite:21 5000
  ```

- To set the time (60 seconds) an HTTP advisor (for port 80) waits before reporting that a receive from a server fails:

  ```
  sscontrol advisor receivetimeout http 80 60
  ```

- To display a report on the state of the ftp advisor (for port 21):

  ```
  sscontrol advisor report ftp 21
  ```

  This command produces output similar to:

  ```
  Advisor Report:
  ---------------
  Advisor name ............. http
  Port number .............. 80

  sitename ................. mySite
  Server address ........... 9.67.129.230
  Load ..................... 8
  ```

- To start the advisor with the ftpadv.log file:

  ```
  sscontrol advisor start ftp 21 ftpadv.log
  ```

- To display the current status of values associated with the http advisor:

  ```
  sscontrol advisor status http 80
  ```

  This command produces output similar to the following:

  ```
  Advisor Status:
  ---------------
  Interval (seconds) ............ 7
  Timeout (seconds) ............. Unlimited
  Connect timeout (seconds).......21
  Receive timeout (seconds).......21
  Advisor log filename .......... Http_80.log
  Log level ..................... 1
  Maximum log size (bytes) ...... Unlimited
  Number of retries ............. 0
  ```

- To stop the http advisor at port 80:

  ```
  sscontrol advisor stop http 80
  ```

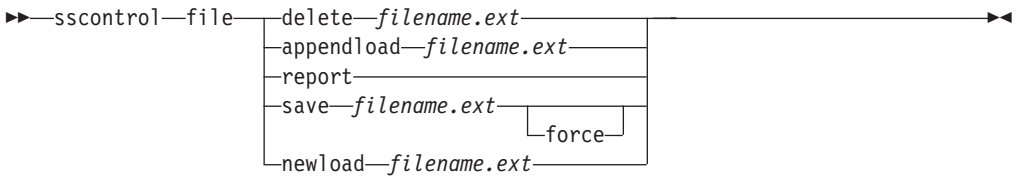
- To set the timeout value for advisor information to 5 seconds:

  ```
  sscontrol advisor timeout ftp 21 5
  ```

- To find out the current version number of the ssl advisor:

  ```
  sscontrol advisor version ssl 443
  ```

sscontrol file — manage configuration files



delete
Delete the file.

file.ext
A configuration file.

The file extension (*.ext*) can be anything you like and is optional.

appendload
Append a configuration file to the current configuration and load into the Site Selector.

report
Report on the available file or files.

save
Save the current configuration for Site Selector to the file.

Note: Files are saved into and loaded from the following directories:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/configurations/ss**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\configurations***component**

force
To save your file to an existing file of the same name, use **force** to delete the existing file before saving the new file. If you do not use the force option, the existing file is not overwritten.

newload
Load a new configuration file into Site Selector. The new configuration file will replace the current configuration.

Examples

- To delete a file:

```
sscontrol file delete file3

File (file3) was deleted.
```

- To load a new configuration file to replace the current configuration:

```
sscontrol file newload file1.sv

File (file1.sv) was loaded into the Dispatcher.
```

- To append a configuration file to the current configuration and load:

```
sscontrol file appendload file2.sv

File (file2.sv) was appended to the current configuration and loaded.
```

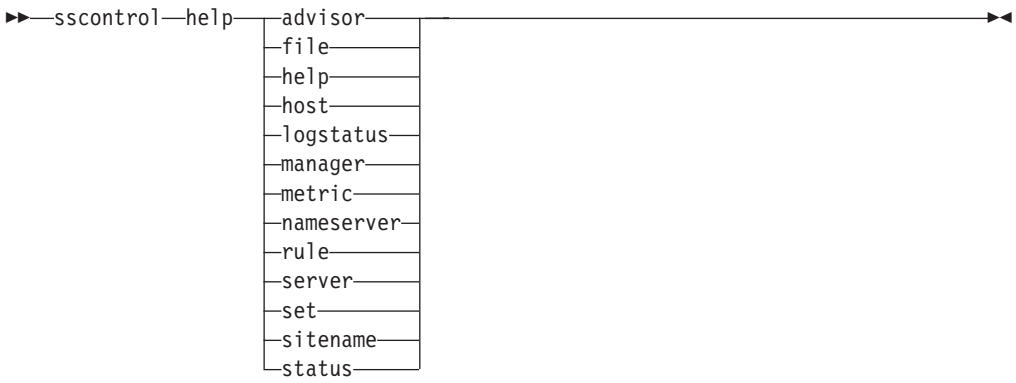
- To view a report of your files (that is, those files that you saved earlier):

```
sscontrol file report

FILE REPORT:
file1.save
file2.sv
file3
```

- To save your configuration into a file named file3:

```
sscontrol file save file3

The configuration was saved into file (file3).
```

sscontrol help — display or print help for this command



Examples

- To get help on the sscontrol command:

  ```
  sscontrol help
  ```

  This command produces output similar to:

  ```
  HELP COMMAND ARGUMENTS:
  ---------------------------------
  Usage:  help <help option>
  Example: help name

  help             - print complete help text
  advisor          - help on advisor command
  file             - help on file command
  host             - help on host command
  manager          - help on manager command
  metric           - help on metric command
  sitename         - help on sitename command
  nameserver       - help on nameserver command
  rule             - help on rule command
  server           - help on server command
  set              - help on set command
  status           - help on status command
  logstatus        - help on logstatus command
  ```

  Parameters within < > are variables.

- Sometimes the help shows choices for the variables using | to separate the options:

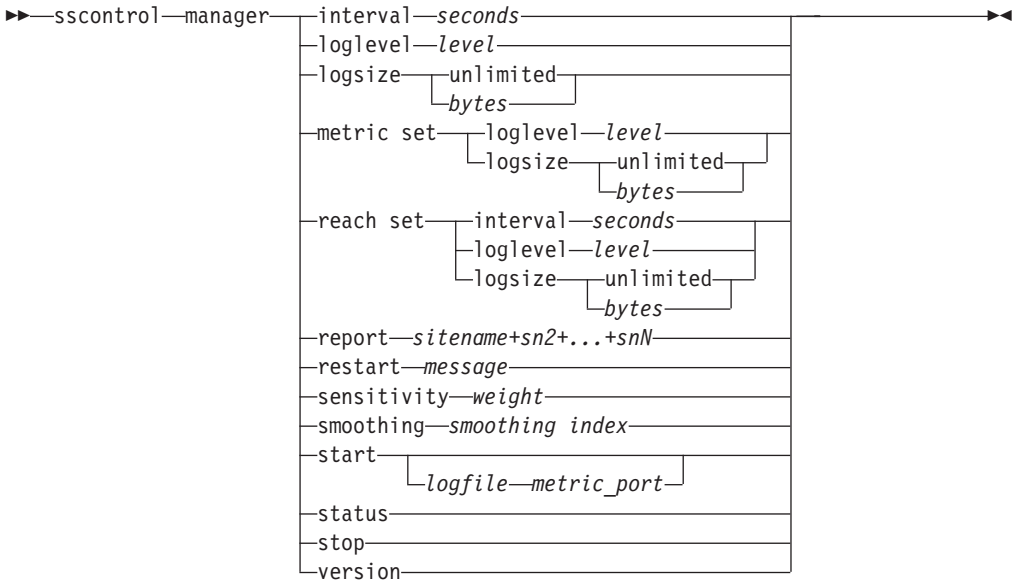
  ```
  logsize <number of bytes | unlimited>
   -Set the maximum number of bytes to be logged in the log file
  ```

sscontrol logstatus — display server log settings

logstatus

Displays the server log settings (log file name, logging level, and log size).

sscontrol manager — control the manager



interval
Set how often the manager updates the weights of the servers.

seconds
A positive number in seconds that represents how often the manager updates weights. The default is 2.

loglevel
Set the logging level for the manager log.

level
The number of the level (0 to 5). The higher the number, the more information that is written to the manager log. The default is 1. The possible values are:

- 0 is None
- 1 is Minimal
- 2 is Basic
- 3 is Moderate
- 4 is Advanced
- 5 is Verbose

logsize
Set the maximum size of the manager log. When you set a maximum size for the log file, the file wraps; when the file reaches the specified size, the subsequent entries are written from the top of the file, overwriting the previous log entries. Log size cannot be set smaller than the current size of the log. Log entries are time-stamped so you can tell the order in which they were written. The higher you set the log level, the more carefully you must choose the log size, because you can quickly run out of space when logging at the higher levels.

bytes
The maximum size in bytes for the manager log file. You can specify either a

positive number greater than zero, or **unlimited**. The log file may not reach the exact maximum size before being overwritten because the log entries vary in size. The default value is 1 MB.

metric set
Sets the **loglevel** and **logsize** for the metric monitor log. The loglevel is the metric monitor logging level (0 - None,1 - Minimal,2 - Basic,3 - Moderate, 4 - Advanced, or 5 - Verbose). The default loglevel is 1. The logsize is the maximum number of bytes to be logged in the metric monitor log file. You can specify either positive number greater than zero, or unlimited. The default logsize is 1.

reach set
Sets the interval, loglevel, and logsize for the reach advisor.

report
Display a statistics snapshot report.

sitename
The sitename you want displayed in the report. This is an unresolvable hostname that the client will request. The sitename must be a fully qualified domain name.

Note: Additional sitenames are separated by a plus sign (+).

restart
Restart all servers (that are not down) to normalized weights (1/2 of maximum weight).

message
A message that you want written to the manager log file.

sensitivity
Set minimum sensitivity to which weights update. This setting defines when the manager should change its weighting for the server based on external information.

weight
A number from 0 to 100 used as the weight percentage. The default of 5 creates a minimum sensitivity of 5%.

smoothing
Set an index that smooths the variations in weight when load balancing. A higher smoothing index causes server weights to change less drastically as network conditions change. A lower index causes server weights to change more drastically.

index
A positive floating point number. The default is 1.5.

start
Start the manager.

log file
File name to which the manager data is logged. Each record in the log is time-stamped.

The default file is installed in the **logs** directory. See Appendix C, “Sample configuration files,” on page 413. To change the directory where the log files are kept, see “Changing the log file paths” on page 219.

metric_port
Port that Metric Server uses to report system loads. If you specify a metric port, you must specify a log file name. The default metric port is 10004.

status
Display the current status and defaults of all the global values in the manager.

stop
Stop the manager.

version
Display the current version of the manager.

Examples

- To set the updating interval for the manager to every 5 seconds:
  ```
  sscontrol manager interval 5
  ```
- To set the level of logging to 0 for better performance:
  ```
  sscontrol manager loglevel 0
  ```
- To set the manager log size to 1,000,000 bytes:
  ```
  sscontrol manager logsize 1000000
  ```
- To get a statistics snapshot of the manager:
  ```
  sscontrol manager report
  ```

  This command produces output similar to:

```
---------------------------------
|     SERVER        |   STATUS   |
---------------------------------
|       9.67.129.221|      ACTIVE|
|       9.67.129.213|      ACTIVE|
|       9.67.134.223|      ACTIVE|
---------------------------------
-------------------------
| MANAGER REPORT LEGEND  |
-------------------------
| CPU  | CPU Load        |
| MEM  | Memory Load     |
| SYS  | System Metric   |
| NOW  | Current Weight  |
| NEW  | New Weight      |
| WT   | Weight          |
-------------------------
-------------------------------------------------------------------------
|        mySite |  WEIGHT  | CPU  49% | MEM  50% | PORT  1% | SYS   0% |
-------------------------------------------------------------------------
|               |NOW   NEW | WT  LOAD | WT  LOAD | WT  LOAD | WT  LOAD |
-------------------------------------------------------------------------
|   9.37.56.180 | 10    10 |-99     -1|-99     -1|-99     -1|  0      0|
-------------------------------------------------------------------------
|        TOTALS:| 10    10 |        -1|        -1|        -1|         0|
-------------------------------------------------------------------------

----------------------------------------
| ADVISOR |  SITENAME:PORT  |  TIMEOUT  |
----------------------------------------
|   http  |             80  | unlimited |
----------------------------------------
```

- To restart all the servers to normalized weights and write a message to the manager log file:
  ```
  sscontrol manager restart Restarting the manager to update code
  ```

This command produces output similar to:

```
320-14:04:54 Restarting the manager to update code
```

- To set the sensitivity to weight changes to 10:

  ```
  sscontrol manager sensitivity 10
  ```

- To set the smoothing index to 2.0:

  ```
  sscontrol manager smoothing 2.0
  ```

- To start the manager and specify the log file named ndmgr.log (paths cannot be set)

  ```
  sscontrol manager start ndmgr.log
  ```

- To display the current status of the values associated with the manager:

  ```
  sscontrol manager status
  ```

  This command produces output similar to the following example.

  ```
  Manager status:
  =============
  Metric port................................. 10004
  Manager log filename........................ manager.log
  Manager log level........................... 1
  Maximum manager log size (bytes)............ unlimited
  Sensitivity level........................... 5
  Smoothing index............................. 1.5
  Update interval (seconds)................... 2
  Weights refresh cycle....................... 2
  Reach log level............................. 1
  Maximum reach log size (bytes).............. unlimited
  Reach update interval (seconds)............. 7
  ```

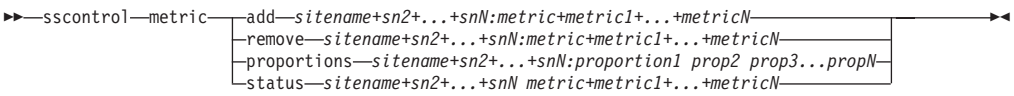
- To stop the manager:

  ```
  sscontrol manager stop
  ```

- To display the current version number of the manager:

  ```
  sscontrol manager version
  ```

sscontrol metric — configure system metrics



add
Add the specified metric.

sitename
The configured sitename. Additional sitenames are separated by a plus sign (+).

metric
The system metric name. This must be the name of an executable or script file in the metric server's script directory.

remove
Remove the specified metric.

proportions
Proportions determines the significance of each metric as compared to the others when they are combined into a single system load for a server.

status
Display the current server values for this metric.

Examples

- To add a system metric:

  ```
  sscontrol metric add site1:metric1
  ```

- To set proportions for a sitename with two system metrics:

  ```
  sscontrol metric proportions site1 0 100
  ```

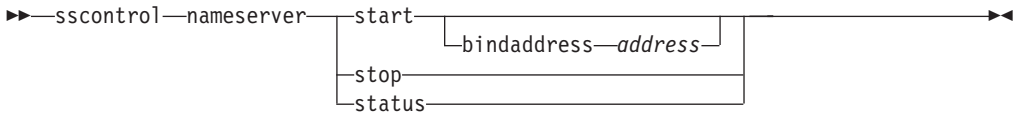
- To display the current status of values associated with the specified metric:

  ```
  sscontrol metric status site1:metric1
  ```

  This command produces output similar to the following:

  ```
  Metric Status:
  ------------
  sitename ..................... site1
  Metric name .................. metric1
  Metric proportion ............. 50
      Server ......... 9.37.56.100
      Metric data .... -1
  ```

sscontrol nameserver — control the NameServer



start
Starts the name server.

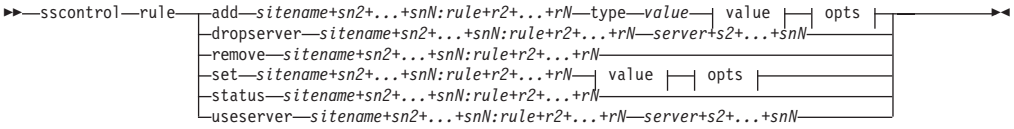
bindaddress
Starts the nameserver bound to the specified address. The nameserver responds only to a request destined for this address.

address
An address (IP or symbolic) configured on the Site Selector machine.

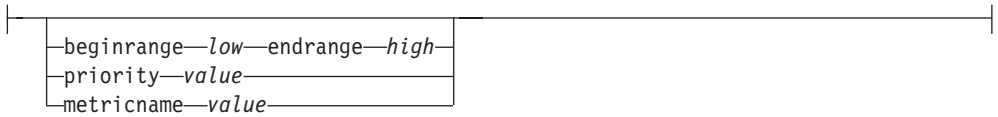
stop
Stops the name server.

status
Displays the status of the name server.

sscontrol rule — configure rules



opts:



add
Add this rule to a sitename.

sitename
An unresolvable hostname that the client will request. The sitename must be a fully qualified domain name. Additional sitenames are separated by a plus sign (+).

rule
The name you choose for the rule. This name can contain any alphanumeric character, underscore, hyphen, or period. It can be from 1 to 20 characters and cannot contain any blanks.

Note: Additional rules are separated by a plus sign (+).

type
The type of rule.

type
Your choices for *type* are:

ip The rule is based on the client IP address.

metricall
The rule is based on the current metric value for all the servers in the server set.

metricavg
The rule is based on the average of the current metric values for all the servers in the server set.

time The rule is based on the time of day.

true This rule is always true. Think of it as an else statement in programming logic.

beginrange
The lower value in the range used to determine whether or not the rule is true.

low
Depends on the type of rule. The kind of value and its default are listed here by the type of rule:

ip The address of the client as either a symbolic name or in IP address format. The default is 0.0.0.0.

time An integer. The default is 0, representing midnight.

metricall
An integer. The default is 100.

metricavg
An integer. The default is 100.

endrange
The higher value in the range used to determine whether or not the rule is true.

high
Depends on the type of rule. The kind of value and its default are listed here by the type of rule:

ip The address of the client as either a symbolic name or in IP address format. The default is 255.255.255.254.

time An integer. The default is 24, representing midnight.

Note: When defining the beginrange and endrange of time intervals, note that each value must be an integer representing only the hour portion of the time; portions of an hour are not specified. For this reason, to specify a single hour—say, the hour between 3:00 and 4:00 am— you would specify a beginrange of **3** and an endrange also of **3**. This will signify all the minutes beginning with 3:00 and ending with 3:59. Specifying a beginrange of **3** and an endrange of **4** would cover the two-hour period from 3:00 through 4:59.

metricall
An integer. The default is 2 to the 32nd power minus 1.

metricavg
An integer. The default is 2 to the 32nd power minus 1.

priority
The order in which the rules are reviewed.

level
An integer. If you do not specify the priority of the first rule you add, Site Selector sets it by default to 1. When a subsequent rule is added, by default its priority is calculated to be 10 + the current lowest priority of any existing rule. For example, assume you have an existing rule whose priority is 30. You add a new rule and set its priority at 25 (which is a *higher* priority than 30). Then you add a third rule without setting a priority. The priority of the third rule is calculated to be 40 (30 + 10).

metricname
Name of the metric measured for a rule.

dropserver
Remove a server from a rule set.

server
The IP address of the TCP server machine as either a symbolic name or in IP address format.

Note: Additional sitenames are separated by a plus sign (+).

remove
Remove one or more rules, separated from one another by plus signs.

set
Set values for this rule.

status
Display all the values of one or more rules.

useserver
Insert server into a rule set.

Examples

- To add a rule that will always be true, do not specify the beginning range or end range:

  ```
  sscontrol rule add sitename:rulename type true priority 100
  ```

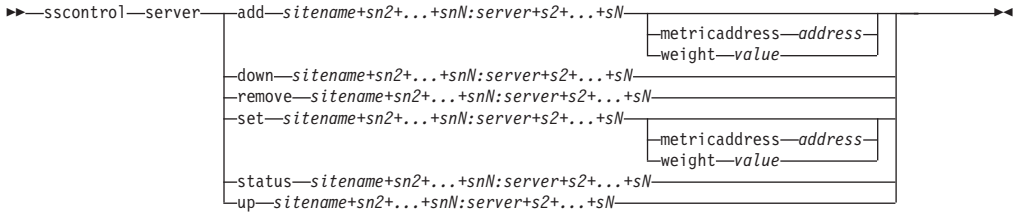
- To create a rule forbidding access to a range of IP addresses, in this case those beginning with "9" :

  ```
  sscontrol rule add sitename:rulename type ip b 9.0.0.0 e 9.255.255.255
  ```

- To create a rule that will specify the use of a given server from the hour of 11:00 a.m. through the hour of 3:00 p.m.:

  ```
  sscontrol rule add sitename:rulename type time beginrange 11 endrange 14
  sscontrol rule useserver sitename:rulename server05
  ```

sscontrol server — configure servers



add
Add this server.

sitename
An unresolvable hostname that the client requests. The sitename must be a fully qualified domain name. Additional sitenames are separated by a plus sign (+).

server
The IP address of the TCP server machine as either a symbolic name or in IP address format.

Note: Additional servers are separated by a plus sign (+).

metricaddress
The address of the metric server.

address
The address of the server as either a symbolic name or in IP address format.

weight
A number from 0–100 (not to exceed the specified sitename's maximum weightbound value) representing the weight for this server. Setting the weight to zero will prevent any new requests from being sent to the server. The default is one-half the specified sitename's maximum weightbound value. If the manager is running, this setting will be quickly overwritten.

value
The server weight value.

down
Mark this server down. This command prevents any other request from being resolved to that server.

remove
Remove this server.

set
Set values for this server.

status
Show status of the servers.

up Mark this server up. Site Selector will now resolve new requests to that server.

Examples

- To add the server at 27.65.89.42 to a sitename of site1:

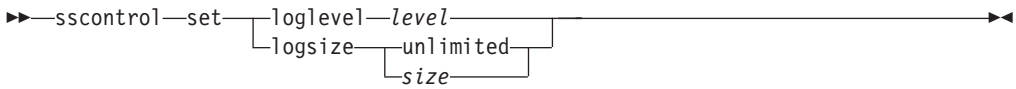
  ```
  sscontrol server add site1:27.65.89.42
  ```

- To mark the server at 27.65.89.42 as down:

```
sscontrol server down site1:27.65.89.42
```

- To remove the server at 27.65.89.42 for all sitenames:
  ```
  sscontrol server remove :27.65.89.42
  ```
- To mark the server at 27.65.89.42 as up:
  ```
  sscontrol server up site1:27.65.89.42
  ```

sscontrol set — configure server log



loglevel
The level at which the ssserver logs its activities.

level
The default value of **loglevel** is 0. The possible values are:

- 0 is None
- 1 is Minimal
- 2 is Basic
- 3 is Moderate
- 4 is Advanced
- 5 is Verbose

logsize
The maximum number of bytes to be logged in the log file.

size
The default value of logsize is 1 MB.

sscontrol sitename — configure a sitename

add
Add a new sitename.

sitename
An irresolvable host name, requested by the client. Additional sitenames are separated by a plus sign (+).

cachelife
The amount of time a proximity response is valid and saved in the cache. The default is 1800. See "Using the Network Proximity feature" on page 93 for more information.

value
A positive number representing the number of seconds a proximity response is valid and saved in the cache.

networkproximity
Determines each server's network proximity to the requesting client. Use this proximity response in the load balancing decision. Set the proximity on or off. See "Using the Network Proximity feature" on page 93 for more information.

value
The choices are yes or no. The default is no, which means network proximity is turned off.

proportions
Set the proportion of importance for cpu, memory, port (information from any advisors), and system metrics for the Metric Server that are used by the manager to set server weights. Each of these values is expressed as a percentage of the total and the total is always 100.

cpu The percentage of CPU in use on each load balanced server machine (input from Metric Server agent).

memory
The percentage of memory in use (input from Metric Server agent) on each load balanced server

port The input from advisors listening on the port.

system The input from the Metric Server.

proximitypercentage
Sets the importance of the proximity response versus the health of the server (manager weight). See "Using the Network Proximity feature" on page 93 for more information.

value
The default is 50.

stickytime
The interval during which a client will receive the same server ID previously returned for the first request. The default value of stickytime is 0, which signifies that the sitename is not sticky.

time
A positive, non-zero number representing the number of seconds during which the client receives the same server ID previously returned for the first request.

ttl
Sets the time to live. This indicates how long another nameserver will cache the resolved response. The default value is 5.

value
A positive number representing the number of seconds the nameserver will cache the resolved response.

waitforallresponses
Sets whether to wait for all proximity responses from the servers before responding to the client request. See "Using the Network Proximity feature" on page 93 for more information.

value
The choices are yes or no. The default is yes.

weightbound
A number representing the maximum weight that can be set for servers on this sitename. The weightbound value set for the sitename may be overridden for individual servers using **server weight**. The default value of sitename weightbound is 20.

weight
The value of weightbound.

set
Set the properties of the sitename.

remove
Remove this sitename.

status
Show current status of a specific sitename.

Examples

- To add a sitename:

```
sscontrol sitename add 130.40.52.153
```

- To turn on network proximity:

```
sscontrol sitename set mySite networkproximity yes
```

- To set a cache life of 1900000 seconds:

```
sscontrol sitename set mySite cachelife 1900000
```

- To set a proximity percent of 45:

```
sscontrol sitename set mySite proximitypercentage 45
```

- To set a sitename to not wait for all responses before responding:

  ```
  sscontrol sitename set mySite waitforallresponses no
  ```

- To set the time to live to 7 seconds:

  ```
  sscontrol sitename set mySite ttl 7
  ```

- To set the proportions of importance for CpuLoad, MemLoad, Port, and System Metric, respectively:

  ```
  sscontrol sitename set mySite proportions 50 48 1 1
  ```

- To remove a sitename:

  ```
  sscontrol sitename remove 130.40.52.153
  ```

- To show the status for sitename mySite:

  ```
  sscontrol sitename status mySite
  ```

  This command produces output similar to:

  ```
  SiteName Status:
  ---------------
  SiteName ........................... mySite
  WeightBound ........................ 20
  TTL ................................ 5
  StickyTime ......................... 0
  Number of Servers .................. 1
  Proportion given to CpuLoad ........ 49
  Proportion given to MemLoad ........ 50
  Proportion given to Port ........... 1
  Proportion given to System metric .. 0
  Advisor running on port ............ 80
  Using Proximity .................... N
  ```

sscontrol status — display whether the manager and advisors are running

Examples

- To see what is running, type:

  ```
  sscontrol status
  ```

  This command produces output similar to:

  ```
         NameServer has been started.
            Manager has been started.
  ----------------------------------------
  | ADVISOR |  SITENAME:PORT  |  TIMEOUT  |
  ----------------------------------------
  |    http |              80 | unlimited |
  ----------------------------------------
  ```

Chapter 28. Command reference for Cisco CSS Controller

This chapter describes how to use the following **ccocontrol** commands for Cisco CSS Controller:

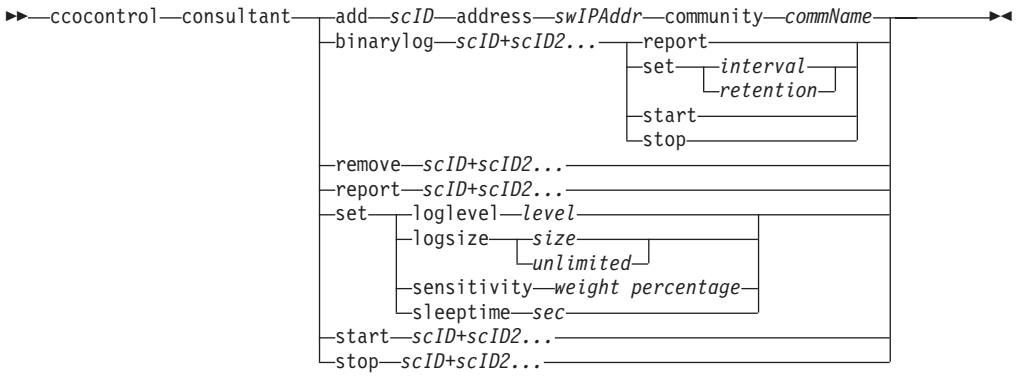
You can use an abbreviated version of the ccocontrol command parameters by typing the unique letters of the parameters. For example, to get help on the file save command, you can type **ccocontrol he f** instead of **ccocontrol help file**.

To get the ccocontrol command prompt: type **ccocontrol**.

To end the command line interface: type **exit** or **quit**.

Note: You must use English characters for all command parameter values. The only exceptions are host names (used in server commands) and file names (used in file commands).

ccocontrol consultant — configure and control a consultant



add
Adds a switch consultant.

scID **(switchConsultantID)**
A user-defined string that refers to the consultant.

address
The IP address of the Cisco CSS Switch to which the consultant provides weights.

swIPAddr **(switchIPAddress)**
The IP address of the switch.

community
The name used in SNMP to get and set communications with the Cisco CSS Switch.

commName
The read/write community name of the Cisco CSS Switch.

binarylog
Controls binary logging for a consultant.

report
Reports on the characteristics of binary logging.

set
Sets how often, in seconds, information is written to the binary logs. The binary logging feature allows service information to be stored in binary log files for each service defined in the configuration. The information is written to the logs only when the specified log interval seconds elapse after the last record was written to the log. The default binary logging interval is 60.

interval
Sets the number of seconds between entries in the binary log.

retention
Sets the number of hours that the binary log files are kept.

start
Starts binary logging.

stop
Stops binary logging.

remove
Removes a switch consultant.

report
Reports on the characteristics of switch consultants.

set
Sets the characteristics of switch consultants.

loglevel
Sets the level at which the switch consultant logs activities. The default value is 1.

level
The number of the level from 0 to 5. The default is 1. The possible values are:

0 = None

1 = Minimal

2 = Basic

3 = Moderate

4 = Advanced

5 = Verbose

logsize
Sets the maximum number of bytes logged in the log file. The default value is 1048576. When you set a maximum size for the log file, the file wraps; when the file reaches the specified size, the subsequent entries are written from the top of the file, overwriting the previous log entries. Log size cannot be set smaller than the current size of the log. Log entries are time-stamped so you can tell the order in which they were written. The higher you set the log level, the more carefully you must choose the log size, because you can quickly run out of space when logging at the higher levels.

size
The maximum number of bytes logged in the consultant log. You can specify either a positive number greater than zero, or the word **unlimited**. The log file might not reach the exact maximum size before overwriting because the log entries vary in size.

sensitivity
Indicates the amount of change that must take place between the old and new weights for the weight to change. The difference between the new and old weight must be greater than the sensitivity percentage for the weight to change. The valid range is 0-through-100; the default is 5.

weight percentage
An integer from 0-to-100, representing the sensitivity value.

sleeptime
Sets the number of seconds to sleep between weight-setting cycles. The default is 7.

sec
An integer, representing the sleeptime in seconds. Valid range is from 0 to 2,147,460.

start
Starts collecting metrics and setting weights.

stop
Stops collecting metrics and setting weights.

Examples

- To add a switch consultant with a switch identifier of sc1, an IP address of 9.37.50.17, and a community name of comm1:

  ```
  ccocontrol consultant add sc1 address 9.37.50.17 community comm2
  ```

- To start binary logging:

  ```
  ccocontrol consultant binarylog sc1 start
  ```

- To see a report on the characteristics of switch consultant sc1:

  ```
  ccocontrol consultant report sc1
  ```

  This command produces output similar to:

  ```
  Consultant sc1 connected to switch at 9.37.50.1:cn1
          Consultant has been started
          Sleep time  = 7
          Sensitivity = 5
          Log level   = 5
          Log size    = 1,048,576
          ownerContent(s):
            ownerContent oc1
  ```

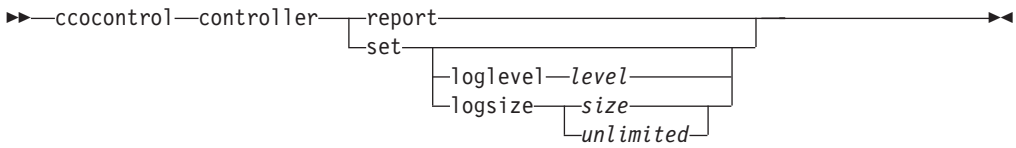
- To set the sleeptime between weight-setting cycles for the sc1 switch ID to 10 seconds:

  ```
  ccocontrol consultant set sc1 sleeptime 10
  ```

- To start collecting metrics and setting weights for the consultant ID of sc1:

  ```
  ccocontrol consultant start sc1
  ```

ccocontrol controller — manage the controller



report
Display characteristics of the controller. Version information displays as part of this report.

set
Set characteristics of the controller.

loglevel
Sets the level at which the controller logs activities. The default value is 1.

level
The number of the level from 0 to 5. The default is 1. The possible values are:

0 = None

1 = Minimal

2 = Basic

3 = Moderate

4 = Advanced

5 = Verbose

logsize
Sets the maximum number of bytes logged in the log file. The default value is 1048576. When you set a maximum size for the log file, the file wraps; when the file reaches the specified size, the subsequent entries are written from the top of the file, overwriting the previous log entries. Log size cannot be set smaller than the current size of the log. Log entries are time-stamped so you can tell the order in which they were written. The higher you set the log level, the more carefully you must choose the log size, because you can quickly run out of space when logging at the higher levels.

size | *unlimited*
The maximum number of bytes logged in the consultant log. You can specify either a positive number greater than zero, or the word **unlimited**. The log file might not reach the exact maximum size before overwriting because the log entries vary in size.

Examples

- To display a report on the controller:

  ```
  ccocontrol controller report
  ```

  This command produces output similar to:

  ```
  Controller Report:
  ----------------------
  Version . . . . . . . . . Version: 05.00.00.00 - 03/21/2002-09:49:57-EST
  Logging level . . . . . . 1
  Log size. . . . . . . . . 1048576
  Configuration File. . . . config1.xml
          Consultants:
                  Consultant consult1 -Started
  ```

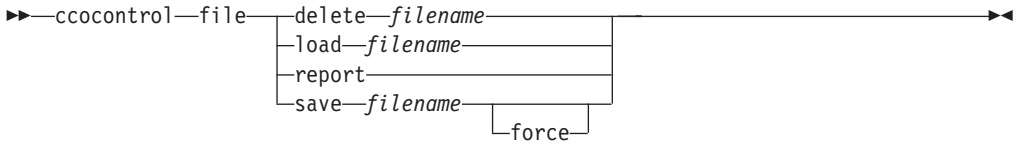
- To set the level of logging to zero for better performance:

```
ccocontrol set loglevel 0
```

- To set the controller log size to 1,000,000 bytes:

```
ccocontrol controller set logsize 1000000
```

ccocontrol file — manage configuration files



delete
Deletes the specified configuration file.

filename
A configuration file. The file extension must be .xml. If this extension is not specified, it will be assumed.

load
Loads the configuration stored in the specified file.

Note: Loading a file appends the configuration stored in that file to the running configuration. If you want to load a *new* configuration, you must stop and restart the server before you load the file.

report
Lists the configuration files.

save
Saves the current configuration to the specified file.

Note: Files are saved into and loaded from the following directories:

- AIX, HP-UX, Linux, and Solaris systems: **/opt/ibm/edge/lb/servers/configurations/cco**
- Windows systems: **<***install_root***>ibm\edge\lb\servers\configurations\cco**

force
Saves to an existing file.

Examples

- To delete a file named file1:

  ```
  ccocontrol file delete file1
  ```

- To append the configuration in the file to the current configuration:

  ```
  ccocontrol file load config2
  ```

- To see a report of files that you have previously saved:

  ```
  ccocontrol file report
  ```

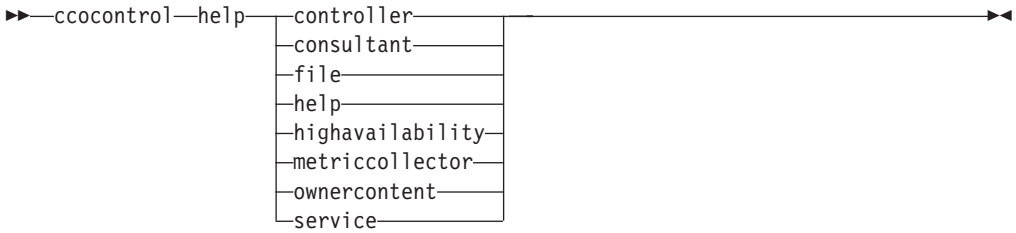
  This command produces output similar to:

  ```
  FILE REPORT:
  ------------
  file1.xml
  file2.xml
  file3.xml
  ```

- To save your configuration file in a file named config2.xml:

  ```
  ccocontrol file save config2
  ```

ccocontrol help — display or print help for this command



Examples

- To get help on the ccocontrol command, type:

  ```
  ccocontrol help
  ```

  This command produces output similar to:

  ```
  The following commands are available:
     controller       - operate on the controller
     consultant       - operate on switch consultants
     file             - operate on configuration files
     help             - operate on help
     highavailability - operate on high availability
     metriccollector  - operate on metric collectors
     ownerContent     - operate on ownerContents
     service          - operate on services
  ```

- The following symbols are used in the online help syntax:

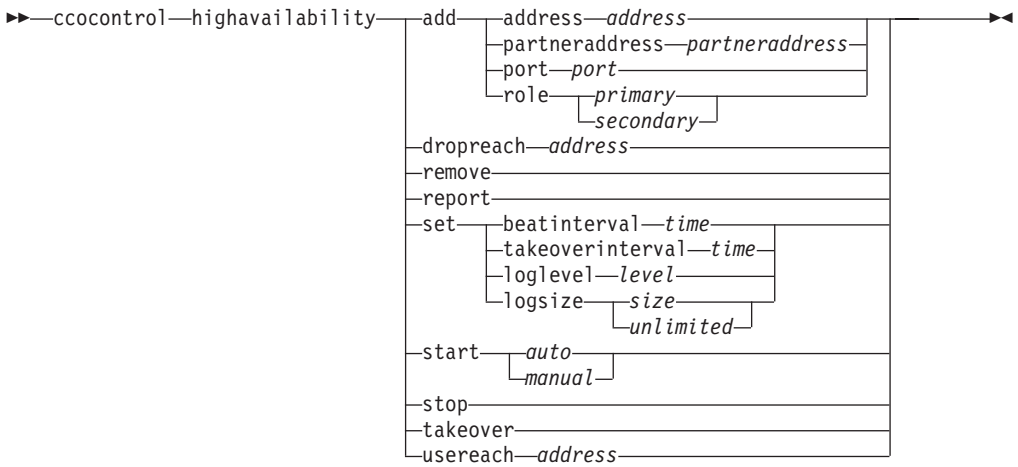
  < > Braces enclose parameters or a sequence of characters.

  [ ] Brackets enclose optional items.

  | A vertical bar separates alternatives within brackets and braces.

  : A colon is a separator between names; for example, **consultant1:ownercontent1**.

ccocontrol highavailability — control high availability



add
Configures a high-availability node, partner, and reach targets.

address
The address from which to receive heartbeats.

address
The IP address of the high-availability node.

partneraddress
The address to which to send heartbeats. This is the IP address or host name configured on the partner node. This address is used to communicate with the partner high-availability machine.

address
The IP address of the partner.

port
The port used to communicate with the partner. The default is 12345.

port
The port number.

role
The high-availability role.

primary | secondary
The primary or secondary role.

dropreach
Remove this reach target from high availability criteria.

address
The IP address of the reach target.

remove
Remove the node, partner and reach target from high availability configuration. High availability must be stopped before using this command.

report
Displays high availability information.

set
Sets the characteristics of high availability.

beatinterval
Sets how often, in milliseconds, heartbeats are sent to the partner. The default is 500.

time
A positive integer representing the beat interval time, in milliseconds.

takeoverinterval
Sets the amount of time, in milliseconds, that must lapse (during which no heartbeats are received) before a takeover occurs. The default is 2000.

time
A positive integer representing the takeover interval time, in milliseconds.

loglevel
Sets the level at which activities are logged. The default value is 1.

level
The number of the level from 0 to 5. The default is 1. The possible values are:

0 = None

1 = Minimal

2 = Basic

3 = Moderate

4 = Advanced

5 = Verbose

logsize
Sets the maximum number of bytes logged in the high availability log file. The default value is 1048576. When you set a maximum size for the log file, the file wraps; when the file reaches the specified size, the subsequent entries are written from the top of the file, overwriting the previous log entries. Log size cannot be set smaller than the current size of the log. Log entries are time-stamped so you can tell the order in which they were written. The higher you set the log level, the more carefully you must choose the log size, because you can quickly run out of space when logging at the higher levels.

size | unlimited
The maximum number of bytes logged in the high availability log. You can specify either a positive number greater than zero, or the word **unlimited**. The log file might not reach the exact maximum size before overwriting because the log entries vary in size.

start
Starts using high availability. A high availability node, partner, and reach target must be configured before using this command.

auto | manual
Determines whether to start high availability with a recovery strategy of automatic or manual.

stop
Stops using high availability.

takeover
Takes control from the active high availability node.

usereach
The reach target address that will start using high availability. Add a reach target that can be pinged so that the high availability partners can determine how reachable their targets are.

address
The IP address of the reach target.

Examples

- To add a high availability node with an IP address of 9.37.50.17 with a primary role on port 12345, and a partner address of 9.37.50.14:

```
ccocontrol highavailability add
  address 9.37.50.17 role primary port 12345 partneraddress 9.37.50.14
```

- To add a reach target address of 9.37.50.9:

```
ccocontrol highavailability usereach 9.37.50.9
```

- To remove the reach target address of 9.37.50.9:

```
ccocontrol highavailability dropreach 9.37.50.9
```

- To start high availability with a recovery strategy of manual:

```
ccocontrol highavailability start manual
```

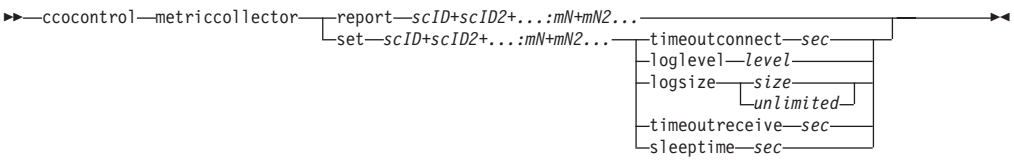
- To get a statistical snapshot of high availability:

```
ccocontrol highavailability report
```

This command produces output similar to:

```
High Availability Status:
-------------------------
Node . . . . . . . . . . . primary
Node Address . . . . . . . 9.37.50.17
Port . . . . . . . . . . . 12345
Partner Address. . . . . . 9.37.50.14
Recovery Strategy. . . . . manual
Heartbeat Interval . . . . 500
Takeover Interval. . . . . 2000
State. . . . . . . . . . . idle
Sub-state. . . . . . . . . unsynchronized

Reachability Status : Node/Partner
-----------------------------------
No reach targets configured
```

ccocontrol metriccollector — configure metric collector



report
Displays the characteristics of a metric collector.

scID **(switch consultant ID)**
A user-defined string that refers to the consultant.

mN **(metric name)**
Name that identifies the provided or custom metric.

set
Sets the characteristics of a metric collector.

timeoutconnect
Set how long a metric collector waits before reporting that a connection fails.

sec
A positive integer representing the amount of time in seconds that the metric collector waits before reporting that a connection to a service has failed.

loglevel
Sets the level at which the specified consultant logs activities. The default is 1.

level
The number of the level. The default is 1. The higher the number, the more information that is written to the consultant log. The possible values are:

0 = None

1 = Minimal

2 = Basic

3 = Moderate

4 = Advanced

5 = Verbose

logsize
Sets the maximum number of bytes logged in the log file. The default value is 1048576. When you set a maximum size for the log file, the file wraps; when the file reaches the specified size, the subsequent entries are written from the top of the file, overwriting the previous log entries. Log size cannot be set smaller than the current size of the log. Log entries are time-stamped so you can tell the order in which they were written. The higher you set the log level, the more carefully you must choose the log size, because you can quickly run out of space when logging at the higher levels.

size | unlimited
The maximum number of bytes logged in the consultant log. You can specify either a positive number greater than zero, or the word **unlimited**. The log file might not reach the exact maximum size before overwriting because the log entries vary in size.

timeoutreceive
Sets how long the consultant waits before reporting that a receive from service fails.

sec
A positive integer representing the amount of time in seconds that the consultant waits before reporting that a receive from a service has failed.

sleeptime
Sets the amount of time in seconds that the metric collector sleeps between metric-collecting cycles.

A positive integer representing the number of seconds of sleep time.

Examples

- To see a report on the characteristics of a metric collector:

  ```
  ccocontrol metriccollector report sc1:http
  ```

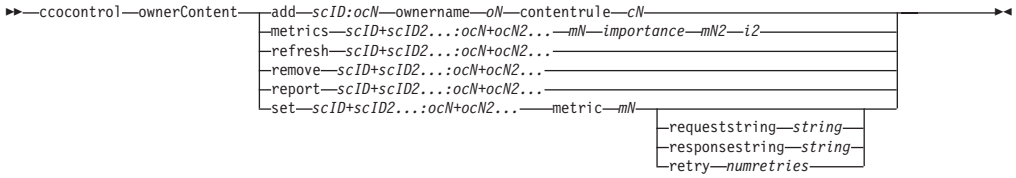
  This command produces output similar to:

  ```
  MetricCollector sc1:http
    collected metric(s).... http
    loglevel.............. 5
    logSize............... 1048576
    sleepTimeSeconds....... 7
    timeoutConnectSeconds.. 21
    timeoutReceiveSeconds.. 21
  ```

- To set a timeoutconnect of 15 seconds and a logsize of unlimited for the sc1 switch consultant and the http metric:

  ```
  ccocontrol metriccollector set sc1:http timeoutconnect 15 logsize unlimited
  ```

ccocontrol ownercontent — control the owner name and content rule



add
Adds an ownercontent to the specified consultant.

scID **(switch consultant ID)**
A user-defined string that represents the consultant.

OCName **(ownercontent name)**
A user-defined string that represents the owner name and the content rule on the switch.

ownername
The name configured on the switch that identifies the owner configuration.

oN **(ownername)**
A unique text string with no spaces. The ownername must be the same as specified on the Cisco switch.

contentrule
The name configured on the switch that identifies the owner's content rule configuration.

cN **(contentname)**
A unique text string with no spaces. The contentname must be the same as specified on the Cisco switch.

metrics
Specifies the set of metrics used in calculating weights and the importance of each metric. The importance is expressed as a percentage of the total. The sum of importance values must total 100. The metrics can be any combination of the connection data metric, application advisor metrics, and metric server metrics. The defaults are active connection (activeconn) and connection rate (connrate) metrics with 50/50 importance.

mN **(metricname)**
Name that identifies the metric collector that will collect measurements to determine the weight of the server.

Following is a list of valid metric names and their associated ports.

Advisor Name	Protocol	Port
connect	ICMP	12345
db2	private	50000
dns	DNS	53
ftp	FTP	21
http	HTTP	80
https	SSL	443
cachingproxy	HTTP (via Caching Proxy)	80
imap	IMAP	143

Advisor Name	Protocol	Port
ldap	LDAP	389
ldaps	SSL	443
nntp	NNTP	119
ping	PING	0
pop3	POP3	110
sip	SIP	5060
smtp	SMTP	25
ssl	SSL	443
telnet	Telnet	23
WLM	private	10,007
activeconn	n/a	n/a
connrate	n/a	n/a
cpuload	n/a	n/a
memload	n/a	n/a

importance
A number from 0-to-100 that represents the importance of this metric in calculating server weights.

refresh
Refreshes the configured services with the configuration from the Cisco CSS Switch.

remove
Removes an ownercontent

report
Reports characteristics of ownercontents.

set
Sets characteristics of ownercontents.

metric
Sets the characteristics of a metric.

mN The name of the desired metric.

requeststring
Sets a request string for the specified metric. This represents the request sent by a metric collector to gather metric information.

string
The request string sent by the metric collector to the server.

responsestring
Sets a response string for the specified metric. The specified response string is used by the metric collector to compare the responses it receives from servers and subsequently determine server availability.

string
The response string to which the metric collector compares received server responses.

retry
Retry sets the number of retries that can be made before marking a server down.

numretries
An integer greater than or equal to zero. This value should be no larger than 3. If the retry keyword is not configured, the number of retries defaults to zero.

Examples

- To add an ownerContent named oc1 (with an owner name of owner1 and a content name of content1) to the switch consultant ID of sc1:

```
ccocontrol ownerContent add sc1:oc1 ownername owner1 contentrule content1
```

- To specify a proportion of 50 each to the activeconn and http metrics:

```
ccocontrol ownerContent metrics sc1:oc1 activeconn 50 http 50
```

- To see a report of characteristics of ownercontents:

```
ccocontrol ownerContent report sc1:oc1
```

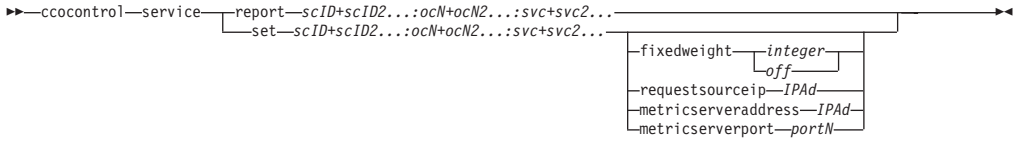
This command produces output similar to:

```
ownerContent sc1:oc1
        Weightbound = 10
        Metric activeconn has proportion 25
          ResponseString... n/a
          RequestString.... n/a
        Metric http has proportion 50
          ResponseString... n/a
          RequestString.... n/a
        Metric connrate has proportion 25
          ResponseString... n/a
          RequestString.... n/a
        Contains Service t3
        Contains Service t2
        Contains Service t1
```

- To set an http request string:

```
ccocontrol ownerContent set sc1:oc1 metric http requeststring getCookie
```

ccocontrol service — configure a service



report
Display characteristics of services.

scID **(switch consultant ID)**
A user-defined string that represents the consultant.

OCName **(ownercontent name)**
A user-defined string that represents the owner name and the content rule on the switch.

svc **(service)**
A user-defined string on the switch that represents the service.

set
Set characteristics of services

fixedweight
Sets a fixed weight for this service. The default is off.

integer | *off*
A positive integer in the range of 0-to-10, representing the fixed weight for this service, or the word **off** to specify no fixed weight.

requestsourceip
Sets the address from which to contact service for application requests.

IPAd **(IP address)**
The IP address from which to contact service, as a symbolic name or in IP address format.

metricserveraddress
Sets the address at which to contact service for metric server requests.

IPAd **(IP address)**
The IP address of the metric server, as a symbolic name or in IP address format.

metricserverport
Sets the port to use for contacting the metric server.

portN **(port number)**
The port number used to contact the metric server.

Examples

- To display a report on service t1 for the sc1 consultant:

  ```
  ccocontrol service report sc1:oc1:t1
  ```

  This command produces output similar to:

  ```
  Service sc1:oc1:ta has weight 10
    Fixed weight is off
    Request Source Ip..... 9.27.24.156
    Application port...... 80
    MetricServer address.. 1.0.0.1
  ```

```
  MetricServer port..... 10004
    Metric activeconn has value -99
    Metric http has value -99
    Metric connrate has value -99
```

- To set a metric server address for service t2:

```
ccocontrol service set sc1:oc1:t2 metricserveraddress 9.37.50.17
```

Chapter 29. Command reference for Nortel Alteon Controller

This chapter describes how to use the following **nalcontrol** commands for Nortel Alteon Controller:

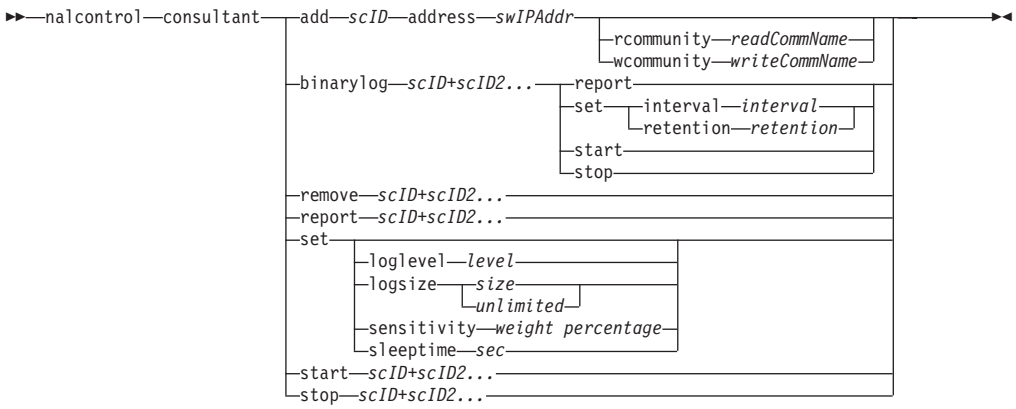
You can use an abbreviated version of the nalcontrol command parameters by typing the unique letters of the parameters. For example, to get help on the file save command, you can type **nalcontrol he f** instead of **nalcontrol help file**.

To get the nalcontrol command prompt: type **nalcontrol**.

To end the command line interface: type **exit** or **quit**.

Note: You must use English characters for all command parameter values. The only exceptions are host names (used in server commands) and file names (used in file commands).

nalcontrol consultant — configure and control a consultant



add
Adds a switch consultant.

scID
A user-defined string that refers to the consultant.

address
The IP address of the Nortel Alteon Web Switch to which the consultant provides weights.

swIPAddr
The IP address of the switch.

rcommunity
The read community name used in the SNMP get communications with the Nortel Alteon Web Switch. The default is public.

readCommName
The string that represents the read community name, as it is configured on the Nortel Alteon Web Switch. The default is public.

wcommunity
The write community name used in the SNMP set communications

writeCommName
The string that represents the write community name, as it is configured on the Nortel Alteon Web Switch. The default is private.

binarylog
Controls binary logging for a consultant.

report
Reports on the characteristics of binary logging.

set
Sets how often, in seconds, information is written to the binary logs. The binary logging feature allows service information to be stored in binary log files for each service defined in the configuration. The information is written to the logs only when the specified log interval seconds elapse after the last record was written to the log. The default binary logging interval is 60.

interval
Sets the number of seconds between entries in the binary log.

retention
Sets the number of hours that the binary log files are kept.

start
Starts binary logging.

stop
Stops binary logging.

remove
Removes a switch consultant.

report
Reports on the characteristics of switch consultants.

set
Sets the characteristics of switch consultants.

loglevel
Sets the level at which the switch consultant logs activities. The default value is 1.

level
The number of the level from 0 to 5. The default is 1. The possible values are:

0 = None

1 = Minimal

2 = Basic

3 = Moderate

4 = Advanced

5 = Verbose

logsize
Sets the maximum number of bytes logged in the log file. The default value is 1048576. When you set a maximum size for the log file, the file wraps; when the file reaches the specified size, the subsequent entries are written from the top of the file, overwriting the previous log entries. Log size cannot be set smaller than the current size of the log. Log entries are time-stamped so you can tell the order in which they were written. The higher you set the log level, the more carefully you must choose the log size, because you can quickly run out of space when logging at the higher levels.

size
The maximum number of bytes logged in the consultant log. You can specify either a positive number greater than zero, or the word **unlimited**. The log file might not reach the exact maximum size before overwriting because the log entries vary in size.

sensitivity
Indicates the amount of change that must take place between the old and new weights for the weight to change. The difference between the new and old weight must be greater than the sensitivity percentage for the weight to change. The valid range is 0-through-100; the default is 5.

weight percentage
An integer from 0-to-100, representing the sensitivity value.

sleeptime
Sets the number of seconds to sleep between weight-setting cycles. The default is 7.

seconds
An integer, representing the sleeptime in seconds. Valid range is from 0 to 2,147,460.

start
Starts collecting metrics and setting weights.

stop
Stops collecting metrics and setting weights.

Examples

- To add a switch consultant with a switch identifier of sc1, an IP address of 9.37.50.17:

```
nalcontrol consultant add sc1 address 9.37.50.17
```

- To start binary logging:

```
nalcontrol consultant binarylog sc1 start
```

- To see a report on the characteristics of switch consultant sc1:

```
nalcontrol consultant report sc1
```

This command produces output similar to:

```
Consultant ID:  sc1  Switch IP addr:  9.37.50.1
Read Community:  public
Write Community:  private
Consultant has been started
        Sleep time  = 7
        Sensitivity = 5
        Log level   = 5
        log size    = 1,048,576
        Service(s):
                 Service svc1
```

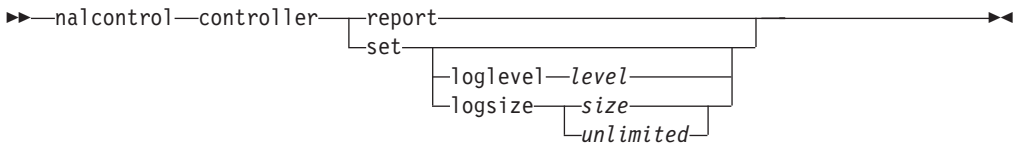
- To set the sleeptime between weight-setting cycles for the sc1 switch ID to 10 seconds:

```
nalcontrol consultant set sc1 sleeptime 10
```

- To start collecting metrics and setting weights for the consultant ID of sc1:

```
nalcontrol consultant start sc1
```

nalcontrol controller — manage the controller



report
Display characteristics of the controller. Version information displays as part of this report.

set
Set characteristics of the controller.

loglevel
Sets the level at which the controller logs activities. The default value is 1.

level
The number of the level from 0 to 5. The default is 1. The possible values are:

0 = None

1 = Minimal

2 = Basic

3 = Moderate

4 = Advanced

5 = Verbose

logsize
Sets the maximum number of bytes logged in the log file. The default value is 1048576. When you set a maximum size for the log file, the file wraps; when the file reaches the specified size, the subsequent entries are written from the top of the file, overwriting the previous log entries. Log size cannot be set smaller than the current size of the log. Log entries are time-stamped so you can tell the order in which they were written. The higher you set the log level, the more carefully you must choose the log size, because you can quickly run out of space when logging at the higher levels.

size | *unlimited*
The maximum number of bytes logged in the consultant log. You can specify either a positive number greater than zero, or the word **unlimited**. The log file might not reach the exact maximum size before overwriting because the log entries vary in size.

Examples

- To display a report on the controller:

  ```
  nalcontrol controller report
  ```

  This command produces output similar to:

  ```
  Controller Report:
  ----------------------
  Version . . . . . . . . . Version: 05.00.00.00 - 03/21/2002-09:49:57-EST
  Logging level . . . . . . 1
  Log size. . . . . . . . . 1048576
  Configuration File. . . . config1.xml
          Consultants:
                  Consultant consult1 -Started
  ```

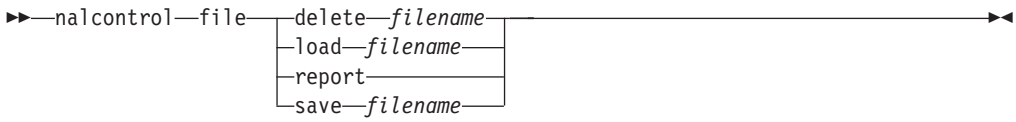
- To set the level of logging to zero for better performance:

```
nalcontrol set loglevel 0
```

- To set the controller log size to 1,000,000 bytes:

```
nalcontrol controller set logsize 1000000
```

nalcontrol file — manage configuration files



delete
Deletes the specified configuration file.

filename
A configuration file. The file extension must be .xml. If this extension is not specified, it will be assumed.

load
Loads the configuration stored in the specified file.

Note: Loading a file appends the configuration stored in that file to the running configuration. If you want to load a *new* configuration, you must stop and restart the server before you load the file.

report
Lists the configuration files.

save
Saves the current configuration to the specified file.

Note: Files are saved into and loaded from the following directories:

- AIX, HP-UX, Linux, and Solaris systems: **/opt/ibm/edge/lb/servers/configurations/nal**
- Windows systems: **<***install_root***>ibm\edge\lb\servers\configurations\nal**

Examples

- To delete a file named file1:

  ```
  nalcontrol file delete file1
  ```

- To load a new configuration file to replace the current configuration:

  ```
  nalcontrol file load config2
  ```

- To see a report of files that you have previously saved:

  ```
  nalcontrol file report
  ```

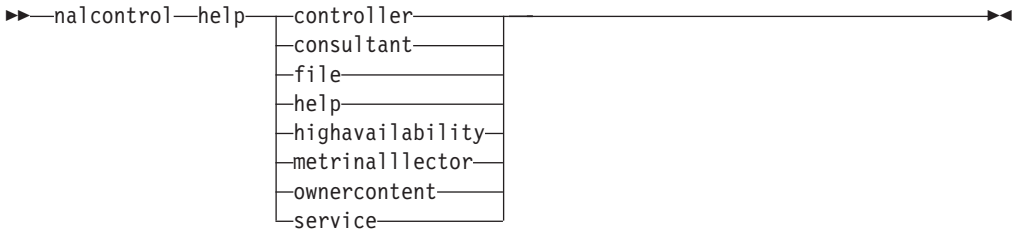
  This command produces output similar to:

  ```
  FILE R EPORT:
  ------------
  file1.xml
  file2.xml
  file3.xml
  ```

- To save your configuration file in a file named config2:

  ```
  nalcontrol file save config2
  ```

nalcontrol help — display or print help for this command



Examples

- To get help on the nalcontrol command, type:

  ```
  nalcontrol help
  ```

  This command produces output similar to:

  ```
  The following commands are available:
     controller       - operate on the controller
     consultant       - operate on switch consultants
     file             - operate on configuration files
     help             - operate on help
     highavailability - operate on high availability
     metriccollector  - operate on metric collectors
     server           - operate on servers
     service          - operate on services
  ```

- The following symbols are used in the online help syntax:

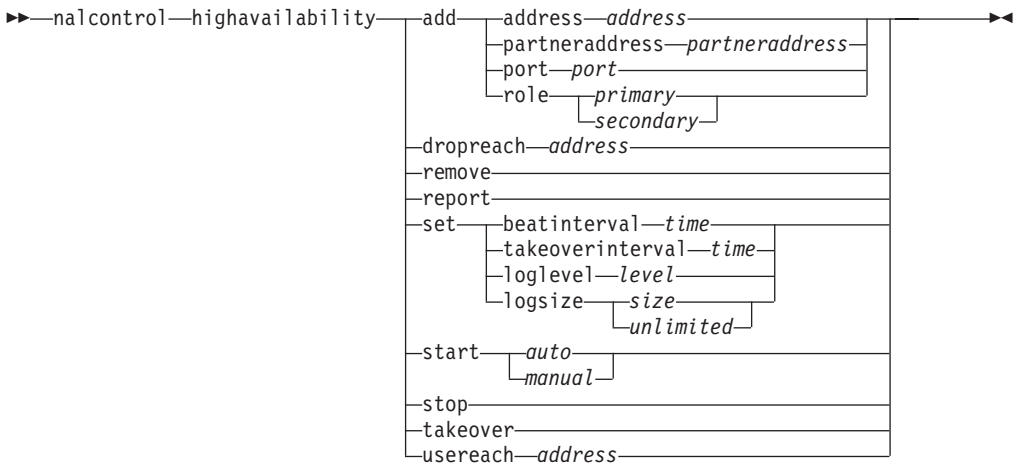
  < > Braces enclose parameters or a sequence of characters.

  [ ] Brackets enclose optional items.

  | A vertical bar separates alternatives within brackets and braces.

  : A colon is a separator between names; for example, **consultant1:service1**.

nalcontrol highavailability — control high availability



add
Configures a high-availability node, partner, and reach targets.

address
The address from which to receive heartbeats.

address
The IP address of the high-availability node.

partneraddress
The address to which to send heartbeats. This is the IP address or host name configured on the partner node. This address is used to communicate with the partner high-availability machine.

address
The IP address of the partner.

port
The port used to communicate with the partner. The default is 12345.

port
The port number.

role
The high-availability role.

primary | secondary
The primary or secondary role.

dropreach
Remove this reach target from high availability criteria.

address
The IP address of the reach target.

remove
Remove the node, partner and reach target from high availability configuration. High availability must be stopped before using this command.

report
Displays high availability information.

set
Sets the characteristics of high availability.

beatinterval
Sets how often, in milliseconds, heartbeats are sent to the partner. The default is 500.

time
A positive integer representing the beat interval time, in milliseconds.

takeoverinterval
Sets the amount of time, in milliseconds, that must lapse (during which no heartbeats are received) before a takeover occurs. The default is 2000.

time
A positive integer representing the takeover interval time, in milliseconds.

loglevel
Sets the level at which activities are logged. The default value is 1.

level
The number of the level from 0 to 5. The default is 1. The possible values are:

0 = None

1 = Minimal

2 = Basic

3 = Moderate

4 = Advanced

5 = Verbose

logsize
Sets the maximum number of bytes logged in the high availability log file. The default value is 1048576. When you set a maximum size for the log file, the file wraps; when the file reaches the specified size, the subsequent entries are written from the top of the file, overwriting the previous log entries. Log size cannot be set smaller than the current size of the log. Log entries are time-stamped so you can tell the order in which they were written. The higher you set the log level, the more carefully you must choose the log size, because you can quickly run out of space when logging at the higher levels.

size | unlimited
The maximum number of bytes logged in the high availability log. You can specify either a positive number greater than zero, or the word **unlimited**. The log file might not reach the exact maximum size before overwriting because the log entries vary in size.

start
Starts using high availability. A high availability node, partner, and reach target must be configured before using this command.

auto | manual
Determines whether to start high availability with a recovery strategy of automatic or manual.

stop
Stops using high availability.

takeover
Takes control from the active high availability node.

usereach
The reach target address that will start using high availability. Add a reach target that can be pinged so that the high availability partners can determine how reachable their targets are.

address
The IP address of the reach target.

Examples

- To add a high availability node with an IP address of 9.37.50.17 with a primary role on port 12345, and a partner address of 9.37.50.14:

```
nalcontrol highavailability add
  address 9.37.50.17 role primary port 12345 partneraddress 9.37.50.14
```

- To add a reach target address of 9.37.50.9:

```
nalcontrol highavailability usereach 9.37.50.9
```

- To remove the reach target address of 9.37.50.9:

```
nalcontrol highavailability dropreach 9.37.50.9
```

- To start high availability with a recovery strategy of manual:

```
nalcontrol highavailability start manual
```

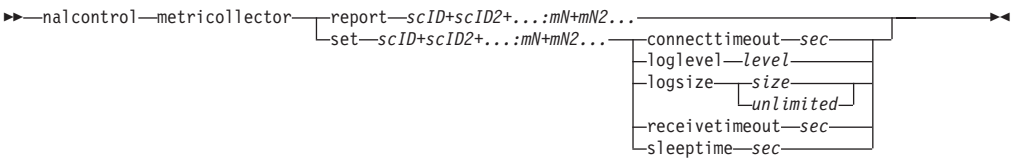
- To get a statistical snapshot of high availability:

```
nalcontrol highavailability report
```

This command produces output similar to:

```
High Availability Status:
-------------------------
Node . . . . . . . . . . . primary
Node Address . . . . . . . 9.37.50.17
Port . . . . . . . . . . . 12345
Partner Address. . . . . . 9.37.50.14
Recovery Strategy. . . . . manual
Heartbeat Interval . . . . 500
Takeover Interval. . . . . 2000
Started. . . . . . . . . . N
State. . . . . . . . . . . idle
Sub-state. . . . . . . . . unsynchronized

Reachability Status : Node/Partner
--------------------------------------
```

nalcontrol metriccollector — configure metric collector



report
Displays the characteristics of a metric collector.

scID **(switch consultant ID)**
A user-defined string that refers to the consultant.

mN **(metric name)**
Name that identifies the provided or custom metric.

set
Sets the characteristics of a metric collector.

connecttimeout
Set how long a metric collector waits before reporting that a connection fails.

sec
A positive integer representing the amount of time in seconds that the metric collector waits before reporting that a connection to a service has failed.

loglevel
Sets the level at which the specified consultant logs activities. The default is 1.

level
The number of the level. The default is 1. The higher the number, the more information that is written to the consultant log. The possible values are:

0 = None

1 = Minimal

2 = Basic

3 = Moderate

4 = Advanced

5 = Verbose

logsize
Sets the maximum number of bytes logged in the log file. The default value is 1048576. When you set a maximum size for the log file, the file wraps; when the file reaches the specified size, the subsequent entries are written from the top of the file, overwriting the previous log entries. Log size cannot be set smaller than the current size of the log. Log entries are time-stamped so you can tell the order in which they were written. The higher you set the log level, the more carefully you must choose the log size, because you can quickly run out of space when logging at the higher levels.

size | *unlimited*
The maximum number of bytes logged in the consultant log. You can specify either a positive number greater than zero, or the word **unlimited**. The log file might not reach the exact maximum size before overwriting because the log entries vary in size.

receivetimeout
Sets how long the consultant waits before reporting that a receive from service fails.

sec
A positive integer representing the amount of time in seconds that the consultant waits before reporting that a receive from a service has failed.

sleeptime
Sets the amount of time in seconds that the metric collector sleeps between metric-collecting cycles.

sec
A positive integer representing the number of seconds of sleep time.

Examples

- To see a report on the characteristics of a metric collector:

  ```
  nalcontrol metrinalllector report sc1:http
  ```

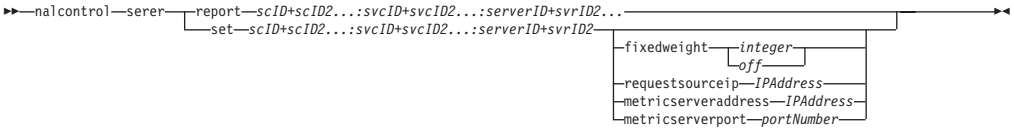
  This command produces output similar to:

  ```
  Metrinalllector sc1:http
    collected metric(s).... http
    loglevel.............. 5
    logSize............... 1048576
    sleepTimeSeconds....... 7
    timeoutConnectSeconds.. 21
    timeoutReceiveSeconds.. 21
  ```

- To set a connecttimeout of 15 seconds and a logsize of unlimited for the sc1 switch consultant and the http metric:

  ```
  nalcontrol metrinalllector set sc1:http connecttimeout 15 logsize unlimited
  ```

nalcontrol server — configure a server



report
Display characteristics of servers.

scID
A user-defined string that represents the consultant.

svcID
A user-defined string that represents the virtual service identifier and the virtual port number on the switch.

serverID
An integer that represents the server on the switch.

set
Set characteristics of servers

fixedweight
Sets a fixed weight for this server. The default is off. The maximum fixedweight is 48.

integer | off
A positive integer representing the fixed weight for this server, or the word **off** to specify no fixed weight.

requestsourceip
Sets the address from which to contact the server for application requests.

IPAddress
The IP address from which to contact the server, as a symbolic name or in IP address format.

metricserveraddress
Sets the address from which to contact the server for metric server requests.

IPAddress
The IP address of the metric server, as a symbolic name or in IP address format.

metricserverport
Sets the port to use for contacting the metric server.

portNumber
The port number used to contact the metric server.

Examples

- To display a report on server 1 for the sc1 consultant:

  ```
  nalcontrol server report sc1:svc1:1
  ```

  This command produces output similar to:

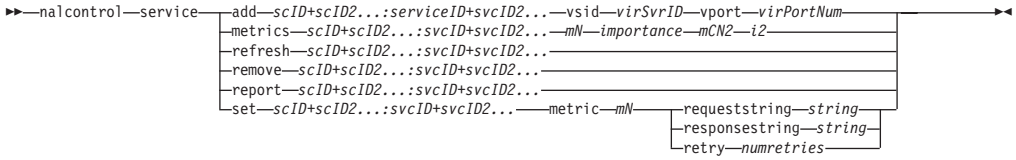
  ```
  Server sc1:svc1:1 has weight -99
          Fixed weight is off
          Request Source Ip...... 9.27.24.156
          Application port....... 99
          MetricServer address... 9.99.99.98
  ```

```
        MetricServer port...... 10004
                Metric activeconn has value -99
                Metric connrate has value -99
```

- To set a metric server address for service 2:

```
nalcontrol server set sc1:svc1:2 metricserveraddress 9.37.50.17
```

nalcontrol service — configure a service



add
Adds a service to the specified consultant.

scID **(switchConsultantID)**
A user-defined string that refers to the consultant.

svcID **(serviceID)**
A user-defined string that identifies the service.

vsid
The virtual service identifier keyword.

virSvrID **(virtualServerID)**
The number on the switch that represents the virtual server.

vport
The virtual port keyword.

virPortNum **(virtualPortNumber)**
The port number for the service that is currently configured on the switch.

metrics
Specifies the set of metrics used in calculating weights and the importance of each metric. The importance is expressed as a percentage of the total. The sum of importance values must total 100. The metrics can be any combination of the connection data metric, application advisor metrics, and metric server metrics. The defaults are active connection (activeconn) and connection rate (connrate) metrics with 50/50 importance.

mN **(metric name)**
Name that identifies the metric collector that will collect measurements to determine the weight of the server.

Following is a list of valid metric names and their associated ports.

Advisor Name	Protocol	Port
connect	ICMP	12345
db2	private	50000
dns	DNS	53
ftp	FTP	21
http	HTTP	80
https	SSL	443
cachingproxy	HTTP (via Caching Proxy)	80
imap	IMAP	143
ldap	LDAP	389
nntp	NNTP	119
ping	PING	0

Advisor Name	Protocol	Port
pop3	POP3	110
sip	SIP	5060
smtp	SMTP	25
ssl	SSL	443
telnet	Telnet	23
WLM	private	10,007
activeconn	n/a	n/a
connrate	n/a	n/a
cpuload	n/a	n/a
memload	n/a	n/a

importance
A number from 0-to-100 that represents the importance of this metric in calculating server weights.

refresh
Refreshes a service with information from the Nortel Alteon Web Switch.

remove
Removes a service.

report
Reports characteristics of a service.

set
Sets characteristics of a service.

metric
Sets the characteristics of a configured metric.

mN **(metric name)**
The name of the desired metric.

requeststring
Sets a request string for the specified metric. This represents the request sent by a metric collector to gather metric information.

string
The request string sent by the metric collector to the server.

responsestring
Sets a response string for the specified metric. The specified response string is used by the metric collector to compare the responses it receives from servers and subsequently determine server availability.

string
The response string to which the metric collector compares received server responses.

retry
Retry sets the number of retries that can be made before marking a server down.

numretries
An integer greater than or equal to zero. This value should be no larger than 3. If retries keyword is not configured, the number of retries defaults to zero.

Examples

- To add a service named svc1 (with an virtual server ID of 1 and a virtual port of 80) to the switch consultant ID of sc1:

  ```
  nalcontrol service add sc1:svc1 vsid 1 vport 80
  ```

- To specify a proportion of 50 each to the activeconn and http metrics:

  ```
  nalcontrol service metrics sc1:svc1 activeconn 50 http 50
  ```

- To see a report of characteristics of ownercontents:

  ```
  nalcontrol service report sc1:svc1
  ```

  This command produces output similar to:x

  ```
  Service sc1:svc1
          Weightbound = 48
          Metric activeconn has proportion 50
          Metric connrate has rpoportion 50
          Contains Server 4
          Contains Server 3
          Contains Server 2
          Contains Server 1
  ```

- To set an http request string:

  ```
  nalcontrol service set sc1:svc1 metric http requeststring getLastErrorCode
  ```

Appendix A. GUI: General instructions

In the Load Balancer graphical user interface (GUI), the left side of the panel displays a tree structure with Load Balancer at the top level, and Dispatcher, Content Based Routing (CBR), Site Selector, Cisco CSS Controller, and Nortel Alteon Controller as components.

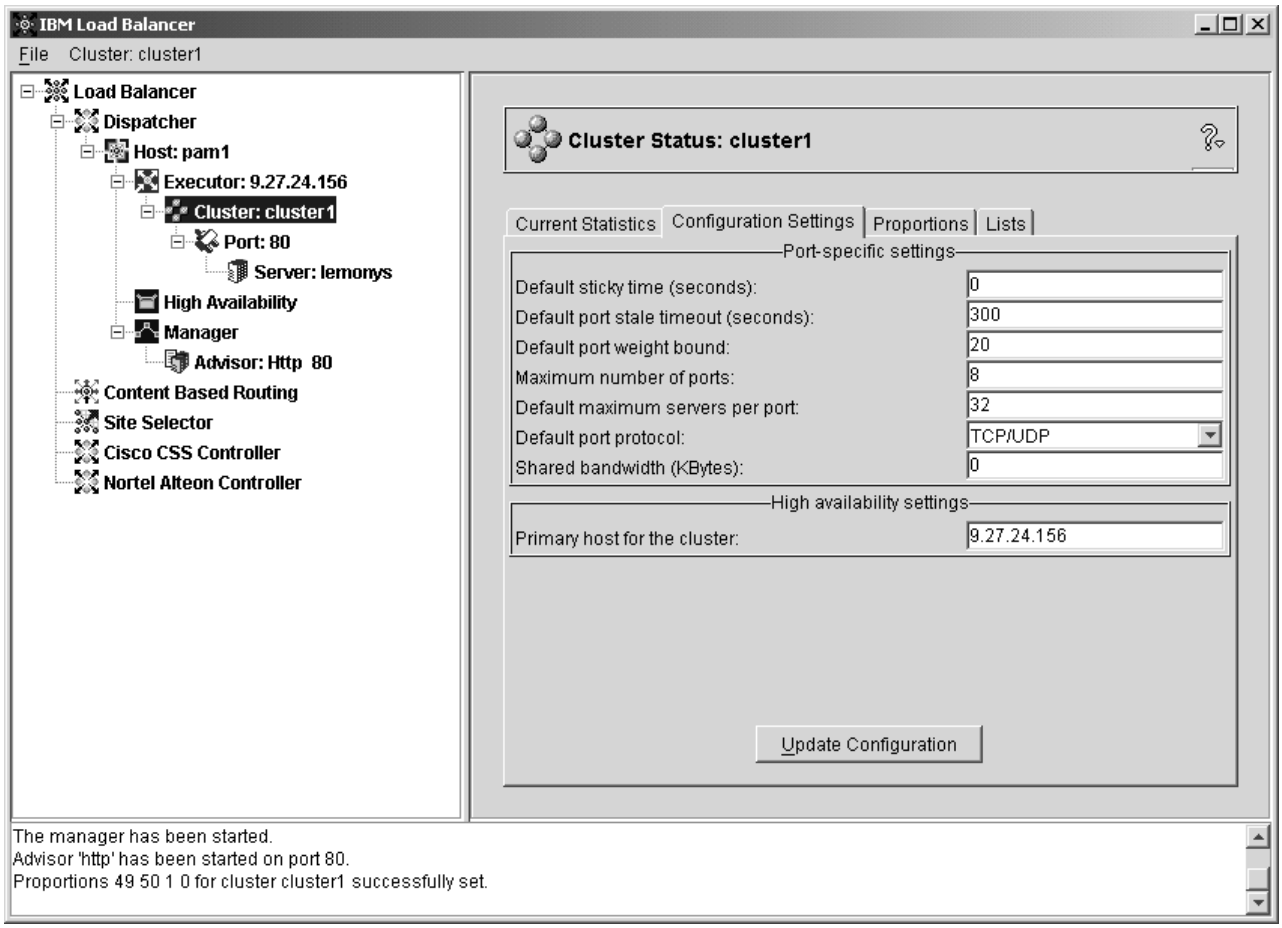


Figure 38. The graphical user interface (GUI) displaying the GUI tree structure expansion of the Dispatcher component

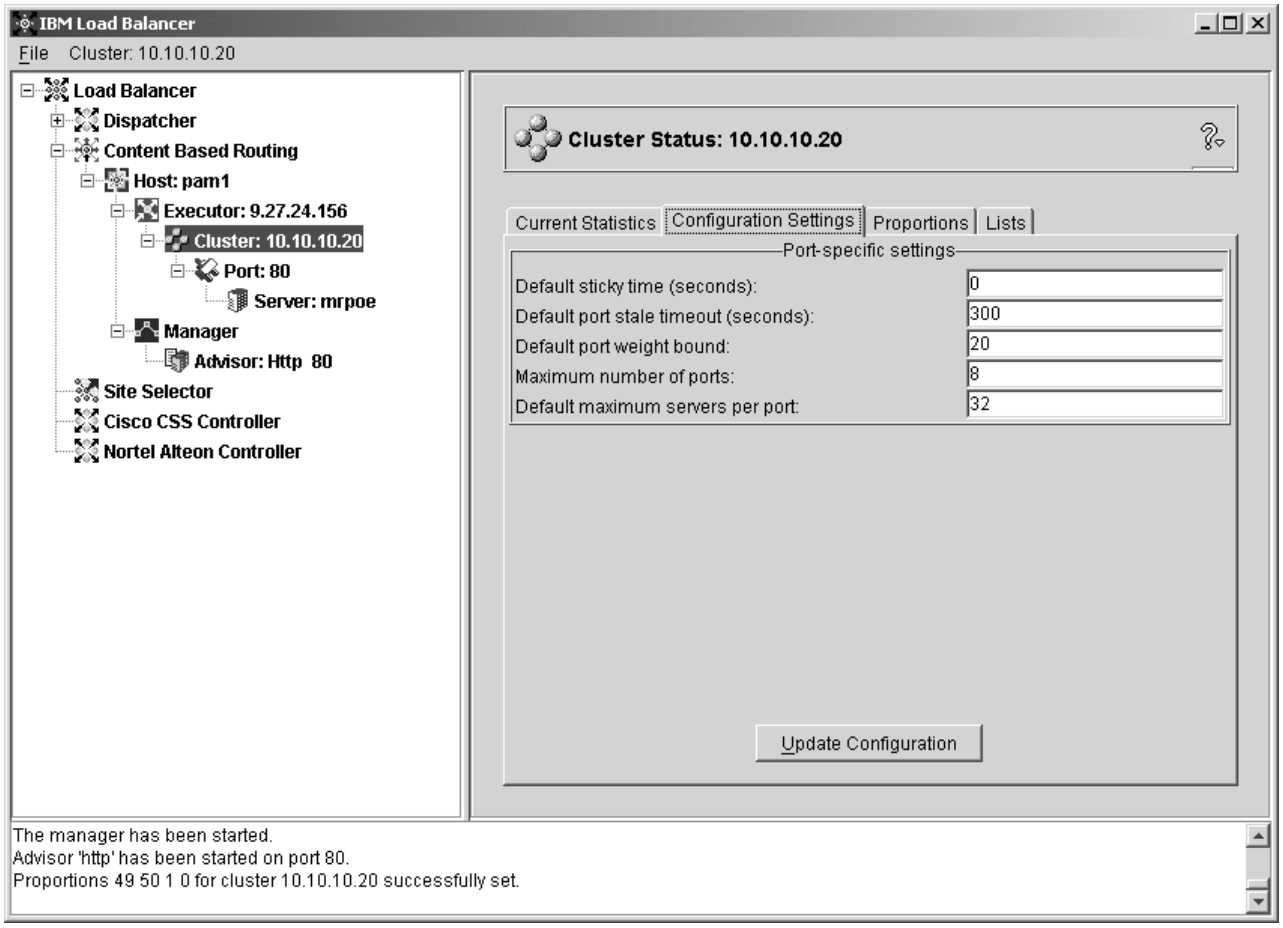


Figure 39. The graphical user interface (GUI) displaying the GUI tree structure expansion of the CBR component

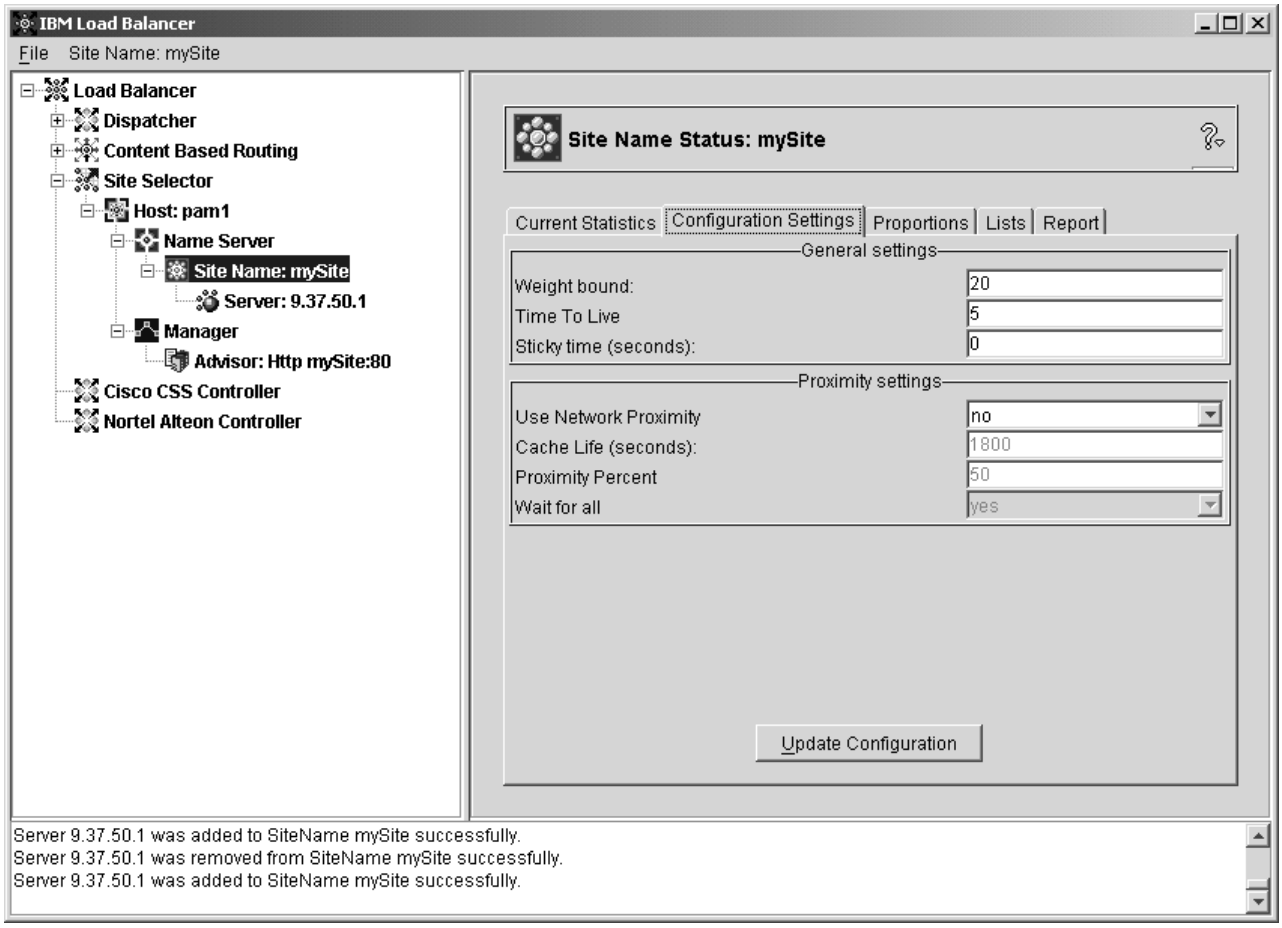


Figure 40. The graphical user interface (GUI) displaying the GUI tree structure expansion of the Site Selector component

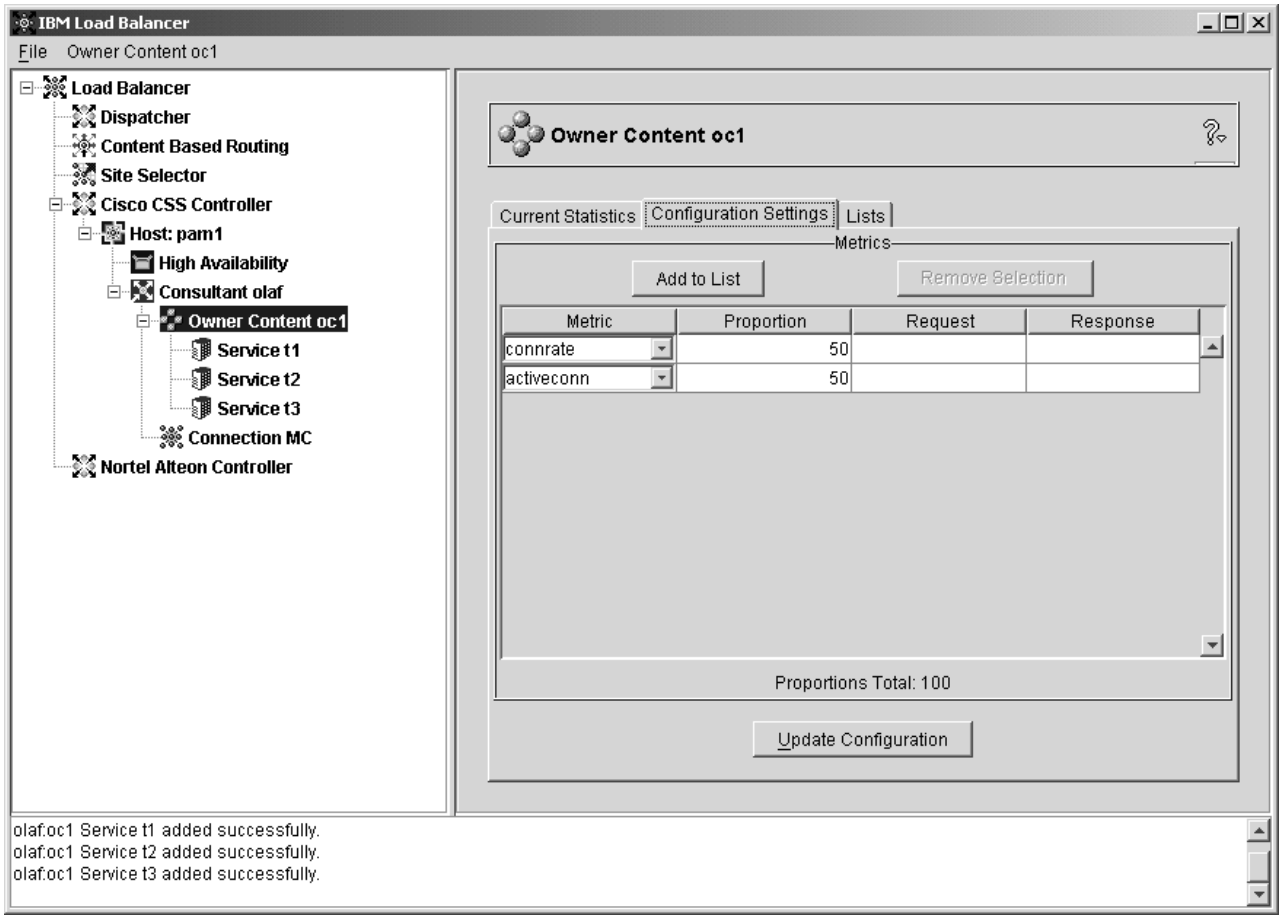


Figure 41. The graphical user interface (GUI) displaying the GUI tree structure expansion of the Cisco CSS Controller component

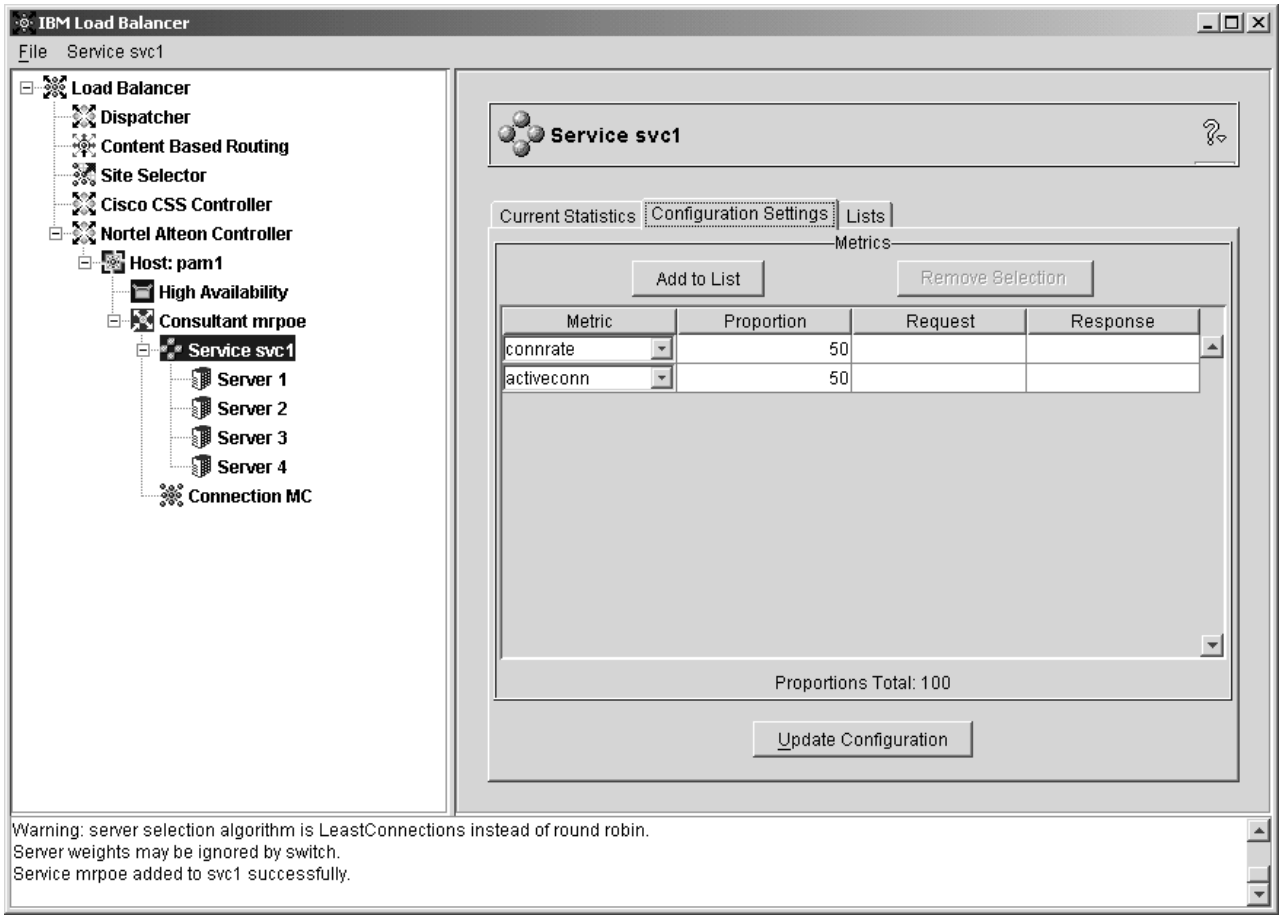


Figure 42. The graphical user interface (GUI) displaying the GUI tree structure expansion of the Nortel Alteon Controller component

All of the components can be configured from the GUI. You can select elements in the tree structure by clicking mouse button one (normally the left button) and then display pop-up menus by clicking mouse button two (normally the right button). The pop-up menus for the tree elements are also accessible from the menu bar located at the top of the panel.

Click the plus or minus signs to expand or compact the items in the tree structure.

To run a command from the GUI: highlight the Host node from the GUI tree and select **Send command....** from the Host pop-up menu. In the command entry field, type the command that you want to run, for example: **executor report**. The results and history of the commands run in the current session appear in the window provided.

The right side of the panel displays status indicator tabs for the element currently selected.

- The **Current Statistics** tab presents statistical information about the element. This tab does not appear for all elements in the tree structure.
- The **Refresh Statistics** button displays the latest statistical data. If a Refresh Statistics button does not appear, the statistics are dynamically refreshed and are always current.

- The **Configuration Settings** tab presents configuration parameters that can be set using the procedures outlined in the configuration chapters for each of the components. This tab does not appear for all elements in the tree structure.
- The **Update Configuration** button applies the latest changes to the configuration currently running.
- The **Proportions** tab presents proportion (or weight) parameters that can be set using the information from Chapter 21, "Advanced features for Dispatcher, CBR, and Site Selector," on page 161. This tab does not appear for all elements in the tree structure.
- The **Lists** tab presents additional details about the selected tree element. This tab does not appear for all elements in the tree structure.
- The **Remove** button deletes highlighted items from the lists.
- The **Report** tab presents manager report information about the element. This tab does not appear for all elements in the tree structure.
- The **Refresh Report** button displays the latest manager report data.

To access **Help**, click the question mark (**?**) in the upper right corner of the Load Balancer window.

- **Help: Field level —** describes each field, default values
- **Help: How do I —** lists tasks that can be done from the current screen
- **InfoCenter —** provides access to product information including: overview and highlight of new feature information, link to product Web site, index of online Help files, glossary of terms

Appendix B. Content rule (pattern) syntax

This appendix describes how to use the content rule (pattern) syntax for the CBR component and the Dispatcher component's cbr forwarding method, along with scenarios and examples of their usage.

Content rule (pattern) syntax:

Only applicable if you selected "content" for the rule type.

Enter the pattern syntax you want to use, with the following restrictions

- no spaces can be used within the pattern
- special characters, unless you precede the character with a backward slash (\):

  ***** wildcard (matches 0 to x of any character)

  (left parenthesis used for logic grouping

) right parenthesis used for logic grouping

  & logical AND

  | logical OR

  ! logical NOT

Reserved keywords

Reserved keywords are always followed by an equal sign “=”.

Method
HTTP method in the request, for example GET, POST, and so forth

URI path of the URL request (case sensitive)

Version
specific version of request, either HTTP/1.0 or HTTP/1.1

Host value from the host: header (not case sensitive)

Note: Optional in HTTP/1.0 protocols

<key> any valid HTTP header name that Dispatcher can search for. Examples of HTTP headers are User-Agent, Connection, Referer, and so forth

A browser targeting `http://www.company.com/path/webpage.htm` might result in values such as:

```
Method=GET
URI=/path/webpage.htm
Version=HTTP/1.1
Host=www.company.com
Connection=Keep-Alive
Referer=http://www.company.com/path/parentwebpage.htm
```

Note: The operating system's shell may interpret special characters, such as "&", and convert them to alternate text before **cbrcontrol** evaluates them. If you are entering the command from the **dscontrol**, **cbrcontrol**, or a configuration file, use double quotation marks (**" "**) around the special characters.

For example, the following command is valid only when using the **cbrcontrol>>** prompt or from a configuration file:

```
rule add 10.1.203.4:80:cbr_prod_rule_ek type content
  pattern "uri=/nipoek/*"
```

When using special characters, for this same command to work at the operating system's prompt you must place double quotation marks around the entire command:

```
cbrcontrol "rule add 10.1.203.4:80:cbr_prod_rule_ek type content
  pattern uri=/nipoek/*"
```

If the quotation marks are not used, some of the pattern might be truncated when the rule is saved in CBR.

The following is a collection of possible scenarios and examples for using pattern syntaxes

Scenario 1:

The setup for one cluster name involves one set of Web servers for standard HTML content, another set of Web servers with WebSphere Application Server for servlet requests, another set of Lotus® Notes® servers for NSF files, and so forth. Access to the client data is required to distinguish between those requested pages. It is also required to send them to the appropriate servers. The content pattern matching rules provide the separation needed to accomplish these tasks. A series of rules are configured so that the necessary separation of requests occurs automatically. For example, the following commands accomplish the three splits mentioned:

```
>>rule add cluster1:80:servlets type content pattern "uri=*/servlet/*" priority 1
>>rule uses cluster1:80:servlets server1+server2
>>rule add cluster1:80:notes type content pattern "uri=*.nsf*" priority 2
>>rule uses cluster1:80:notes server3+server4
>>rule add cluster1:80:regular type true priority 3
>>rule uses cluster1:80:regular server5+server6
```

If a request for an NSF file arrives at Load Balancer, the servlets rule is checked first, but does not match. The request is then checked by the notes rule and returns a match. The client is load-balanced between server3 and server4.

Scenario 2

Another common scenario is when the main Web site controls several distinct internal groups. For example, `www.company.com/software` involves a different set of servers and content from `www.company.com/hardware` division. Because the requests are all based off the root `www.company.com` cluster, content rules are required to find the URI differences and complete load balancing. The scenario's rule looks similar to the following:

```
>>rule add cluster1:80:div1 type content pattern "uri=/software/*" priority 1
>>rule uses cluster1:80:div1 server1+server2
>>rule add cluster1:80:div2 type content pattern "uri=/hardware/*" priority 2
>>rule uses cluster1:80:div2 server3+server4
```

Scenario 3

Certain combinations are sensitive to the order in which rules are searched. For example, in Scenario 2, clients were split based on a directory in their request path;

however, the target directory might appear at multiple levels of the path and mean different things on placement. For example, `www.company.com/pcs/fixes/software` is a different target from `www.company.com/mainframe/fixes/software`. The rules must be defined to account for this possibility and not catch too many scenarios at the same time. For example, the "uri=*/software/*" test is too broad a wildcard search in this case. Alternative rules could be structured in the following manner:

A combination search can narrow this down:

```
>>rule add cluster1:80:pcs type content pattern "(uri=/pcs/*)&(uri=*/software/*)"
>>rule uses cluster 1:80:pcs server1
```

In cases where there are no combinations to use, the order becomes important:

```
>>rule add cluster1:80:pc1 type content pattern "uri=/pcs/*"
>>rule uses cluster1:80:pc1 server2
```

The second rule catches when "pcs" appears in later directory spots instead of the first.

```
>>rule add cluster1:80:pc2 type content pattern "uri=/*/pcs/*"
>>rule uses cluster1:80:pc2 server3
```

In almost every case, you want to complete the rules with a default **always true** rule to catch anything that falls through the other rules. This can also be a "Sorry, the site is currently down, please try again later" server for scenarios where all other servers fail for this client.

```
>>rule add cluster1:80:sorry type true priority 100
>>rule uses cluster1:80:sorry server5
```

Appendix C. Sample configuration files

This appendix contains sample configuration files for the Dispatcher component of Load Balancer.

Sample Load Balancer configuration files

Sample files are located in the following directory:

- AIX, HP-UX, Linux, and Solaris operating systems: **/opt/ibm/edge/lb/servers/samples**
- Windows operating systems: **<***install_root***>ibm\edge\lb\servers\samples**

Dispatcher Configuration file — AIX, Linux, and Solaris systems

```
#!/bin/bash
#
# configuration.sample - Sample configuration file for the
Dispatcher component
#
#
# Ensure the root user is the one executing this script.
#
# iam=`whoami`

# if [ "$iam" != "root" ]if [ "$iam" != "root" ]
#  then
#  echo "You must login as root to run this script"
#   exit 2
# fi

#
# First start the server
#
# dsserver start
# sleep 5

#
# Then start the executor
#
# dscontrol executor start

#
#  The Dispatcher can be removed at any time using the
# "dscontrol executor stop" and "dsserver stop" commands to
# stop the executor and server respectively prior to removing
# the Dispatcher software.
#
# The next step in configuring the Dispatcher is to set the
# NFA (non-forwarding address) and the cluster address(es).
#
# The NFA is used to remotely access the Dispatcher machine
# for administration or  configuration purposes.  This
# address is required since the Dispatcher will forward  packets
# to the cluster address(es).
#
# The CLUSTER address is the hostname (or IP address) to
# which remote clients will connect.
#
# Anywhere in this file, you may use hostnames and IP
```

```
# addresses interchangeably.
#

# NFA=hostname.domain.name
# CLUSTER=www.yourcompany.com

# echo "Loading the non-forwarding address"
# dscontrol executor set nfa $NFA

#
#  The next step in configuring the Dispatcher is to create
# a cluster.  The Dispatcher will route requests sent to
# the cluster address to the corresponding server machines
# defined to that cluster.  You may configure and server
# multiple cluster address using Dispatcher.

# Use a similar configuration for CLUSTER2, CLUSTER3, etc.
#

# echo "Loading first CLUSTER address "
# dscontrol cluster add $CLUSTER

#
# Now we must define the ports this cluster will use.  Any
# requests received by the Dispatcher on a defined port will
# be forwared to the corresponding port of one of the server
# machines.
#

# echo "Creating ports for CLUSTER: $CLUSTER"

# dscontrol port add $CLUSTER:20+21+80

#
# The last step is to add each of the server machines to the
# ports in this cluster.
# Again, you can use either the hostname or the IP address
# of the server machines.
#

# SERVER1=server1name.domain.name
# SERVER2=server2name.domain.name
# SERVER3=server3name.domain.name

# echo "Adding server machines"
# dscontrol server add $CLUSTER:20+21+80:
# $SERVER1+$SERVER2+$SERVER3

#
#  We will now start the load balancing components of the
# Dispatcher.  The main load  balancing component is called
# the manager and the second load balancing components are the
# advisors.  If the manager and advisors are not running the
# Dispatcher sends requests in a round-robin format.  Once the
# manager is started, weighting decisions based on the number
# of new and active connections is employed and incoming
# requests are sent to the best server.  The advisors give the
# manager further insight into a servers ability to service
# requests as well as detecting whether a server is up.  If
# an advisor detects that a server is down it will be
# marked down (providing the manager proportions have been
# set to include advisor input) and no further requests will be
# routed to the server.

#  The last step in setting up the load balancing components
# is to set the manager proportions.  The manager updates the
# weight of each of the servers based on four policies:
```

```
#   1. The number of active connections on each server.
#   2. The number of new connections to each server.
#   3. Input from the advisors.
#   4. Input from the system level advisor.
# These proportions must add up to 100.  As an example, setting
# the manager proportions to
#    dscontrol manager proportions 48 48 0 0
# will give active and new connections 48% input into the
# weighting decision, the advisors will contribute 4% and
# the system input will not be considered.
#
# NOTE: By default the manager proportions are set to 50 50 0 0
#

# echo "Starting the manager..."
# dscontrol manager start

# echo "Starting the FTP advisor on port 21 ..."
# dscontrol advisor start ftp 21
# echo "Starting the HTTP advisor on port 80 ..."
# dscontrol advisor start http 80
# echo "Starting the Telnet advisor on port 23 ..."
# dscontrol advisor start telnet 23
# echo "Starting the SMTP advisor on port 25 ..."
# dscontrol advisor start smtp 25
# echo "Starting the POP3 advisor on port 110 ..."
# dscontrol advisor start pop3 110
# echo "Starting the NNTP advisor on port 119 ..."
# dscontrol advisor start nntp 119
# echo "Starting the SSL advisor on port 443 ..."
# dscontrol advisor start ssl 443
#

# echo "Setting the manager proportions..."
# dscontrol manager proportions 58 40 2 0

#
# The final step in setting up the Dispatcher machine is to
# alias the Network Interface Card (NIC).
#
# NOTE: Do NOT use this command in a high availability
# environment.  The go* scripts will configure the NIC and
# loopback as necessary.
# dscontrol executor configure $CLUSTER

#  If your cluster address is on a different NIC or subnet
from the NFA use the following format for the cluster configure
command.
#  dscontrol executor configure $CLUSTER tr0 0xfffff800
# where tr0 is your NIC (tr1 for the second token ring card, en0
# for the first ethernet card) and 0xfffff800 is a valid
# subnet mask for your site.
#

#
# The following commands are set to the default values.
# Use these commands as a guide to change from the defaults.
#  dscontrol manager loglevel    1
#  dscontrol manager logsize     1048576
#  dscontrol manager sensitivity 5
#  dscontrol manager interval    2
#  dscontrol manager refresh     2
#
#  dscontrol advisor interval ftp  21  5
#  dscontrol advisor loglevel ftp  21  1
#  dscontrol advisor logsize  ftp  21  1048576
#  dscontrol advisor timeout  ftp  21  unlimited
```

```
#  dscontrol advisor interval telnet 23 5
#  dscontrol advisor loglevel telnet 23 1
#  dscontrol advisor logsize  telnet 23 1048576
#  dscontrol advisor timeout  telnet 23 unlimited
#  dscontrol advisor interval smtp 25  5
#  dscontrol advisor loglevel smtp 25  1
#  dscontrol advisor logsize  smtp 25  1048576
#  dscontrol advisor timeout  smtp 25  unlimited
#  dscontrol advisor interval http 80  5
#  dscontrol advisor loglevel http 80  1
#  dscontrol advisor logsize  http 80  1048576
#  dscontrol advisor timeout  http 80  unlimited
#  dscontrol advisor interval pop3 110 5
#  dscontrol advisor loglevel pop3 110 1
#  dscontrol advisor logsize  pop3 110 1048576
#  dscontrol advisor timeout  pop3 110 unlimited
#  dscontrol advisor interval nntp 119 5
#  dscontrol advisor loglevel nntp 119 1
#  dscontrol advisor logsize  nntp 119 1048576
#  dscontrol advisor timeout  nntp 119 unlimited
#  dscontrol advisor interval ssl  443 5
#  dscontrol advisor loglevel ssl  443 1
#  dscontrol advisor logsize  ssl  443 1048576
#  dscontrol advisor timeout  ssl  443 unlimited
#
```

Dispatcher Configuration file — Windows systems

The following is a sample Load Balancer configuration file called **configuration.cmd.sample** for use with Window.

```
@echo off
rem configuration.cmd.sample - Sample configuration file for the
rem Dispatcher component.
rem

rem dsserver must be started by Services

rem

rem
rem Then start the executor
rem
rem call dscontrol executor start

rem

rem The next step in configuring the Dispatcher is to set the
rem NFA (non-forwarding address) and to set the cluster
rem address(es).
rem

rem The NFA is used to remotely access the Dispatcher
rem machine for administration configuration purposes.  This
rem address is required since the Dispatcher will forward
rem packets to the cluster address(es).

rem
rem The CLUSTER address is the hostname (or IP address) to which
rem remote clients will connect.
rem

rem Anywhere in this file, you may use hostnames and IP
rem addresses interchangeably.
rem  NFA=[non-forwarding address]
rem CLUSTER=[your clustername]
rem
```

```
rem set NFA=hostname.domain.name
rem set CLUSTER=www.yourcompany.com

rem echo "Loading the non-forwarding address"
rem call dscontrol executor set nfa %NFA%

rem
rem The following commands are set to the default values.
rem Use these commands to change the defaults

rem  call dscontrol executor set fintimeout 30
rem
rem The next step in configuring the Dispatcher is to create
rem a cluster.  The Dispatcher will route requests sent to
rem the cluster address to the corresponding server machines
rem defined to that cluster.  You may configure and server
rem multiple cluster addresses using Dispatcher.
rem Use a similar configuration for CLUSTER2, CLUSTER3, etc.
rem

rem echo "Loading first CLUSTER address "
rem call dscontrol cluster add %CLUSTER%

rem
rem Now we must define the ports this cluster will use. Any
rem requests received by the Dispatcher on a defined port
rem will be forwarded to the corresponding
rem port of one of the server machines.
rem

rem echo "Creating ports for CLUSTER: %CLUSTER%"
rem call dscontrol port add %CLUSTER%:20+21+80

rem
rem The last step is to add each of the server machines to
rem the ports in this cluster.  Again, you can use either the
rem hostname or the IP address of the server machines.
rem

rem set SERVER1=server1name.domain.name
rem set SERVER2=server2name.domain.name
rem set SERVER3=server3name.domain.name

rem echo "Adding server machines"
rem call dscontrol server add %CLUSTER%:20+21+80:
rem %SERVER1%+%SERVER2%+%SERVER3%

rem
rem  We will now start the load balancing components of the
rem Dispatcher.  The main load balancing component is called
rem the manager and the second load balancing components are the
rem advisors.  If the manager and advisors are not
rem running the Dispatcher sends requests in a round-robin
rem format.  Once the manager is started, weighting decisions
rem based on the number of new and active connections is
rem employed and incoming requests are sent to the best
rem server.  The advisors give the manager further insight
rem into a servers ability to service requests as well as
rem detecting whether a server is up.  If an advisor detects
rem that a server is down it will be marked down (providing the
rem manager proportions have been set to include advisor
rem input) and no further requests will be routed to the server.
rem The last step in setting up the load balancing
rem components is to set the manager proportions.  The
rem manager updates the weight of each of the servers based
rem on four policies:
```

```
rem   1.  The number of active connections on each server
rem   2.  The number of new connections for each server
rem   3.  Input from the advisors.
rem   4.  Input from the system level advisor.
rem
rem  These proportions must add up to 100.  As an example,
rem  setting the cluster proportions using
rem      dscontrol cluster set <cluster> proportions 48 48 4 0
rem  will give active and new connections 48% input into the
rem  weighting decision, the advisor will contribute 4% and
rem  the system input will not be considered.
rem
rem NOTE: By default the manager proportions are set to
rem 50 50 0 0

rem echo "Starting the manager..."
rem call dscontrol manager start

rem echo "Starting the FTP advisor on port 21 ..."
rem call dscontrol advisor start ftp 21
rem echo "Starting the HTTP advisor on port 80 ..."
rem call dscontrol advisor start http 80
rem echo "Starting the Telnet advisor on port 23 ..."
rem call dscontrol advisor start telnet 23
rem echo "Starting the SMTP advisor on port 25 ..."
rem call dscontrol advisor start smtp 25
rem echo "Starting the POP3 advisor on port 110 ..."
rem call dscontrol advisor start pop3 110
rem echo "Starting the NNTP advisor on port 119 ..."
rem call dscontrol advisor start nntp 119
rem echo "Starting the SSL advisor on port 443 ..."
rem call dscontrol advisor start ssl 443
rem

rem echo "Setting the cluster proportions..."
rem call dscontrol cluster set %CLUSTER% proportions 58 40 2 0

rem
rem The final step in setting up the Dispatcher machine is
rem  to alias the Network Interface Card (NIC).
rem
rem NOTE: Do NOT use this command in a high availability
rem environment.  The go* scripts will configure the NIC and
rem loopback as necessary.
rem
rem dscontrol executor configure %CLUSTER%

rem  If your cluster address is on a different NIC or subnet
rem  from the NFA use the following format for the cluster
rem  configure command.
rem  dscontrol executor configure %CLUSTER% tr0 0xfffff800
rem  where tr0 is your NIC (tr1 for the second token ring card,
rem  en0 for the first ethernet card) and 0xfffff800 is
rem  a valid subnet mask for your site.
rem

rem
rem The following commands are set to the default values.
rem Use these commands to guide to change from the defaults.
rem call dscontrol manager loglevel    1
rem call dscontrol manager logsize     1048576
rem call dscontrol manager sensitivity 5
rem call dscontrol manager interval    2
rem call dscontrol manager refresh     2
rem
rem call dscontrol advisor interval ftp  21  5
```

```
rem call dscontrol advisor loglevel ftp  21  1
rem call dscontrol advisor logsize  ftp  21  1048576
rem call dscontrol advisor timeout  ftp  21  unlimited
rem call dscontrol advisor interval telnet 23 5
rem call dscontrol advisor loglevel telnet 23 1
rem call dscontrol advisor logsize  telnet 23 1048576
rem call dscontrol advisor timeout  telnet 23 unlimited
rem call dscontrol advisor interval smtp 25  5
rem call dscontrol advisor loglevel smtp 25  1
rem call dscontrol advisor logsize  smtp 25  1048576
rem call dscontrol advisor timeout  smtp 25  unlimited
rem call dscontrol advisor interval http 80  5
rem call dscontrol advisor loglevel http 80  1
rem call dscontrol advisor logsize  http 80  1048576
rem call dscontrol advisor timeout  http 80  unlimited
rem call dscontrol advisor interval pop3 110 5
rem call dscontrol advisor loglevel pop3 110 1
rem call dscontrol advisor logsize  pop3 110 1048576
rem call dscontrol advisor timeout  pop3 110 unlimited
rem call dscontrol advisor interval nntp 119 5
rem call dscontrol advisor loglevel nntp 119 1
rem call dscontrol advisor logsize  nntp 119 1048576
rem call dscontrol advisor timeout  nntp 119 unlimited
rem call dscontrol advisor interval ssl  443 5
rem call dscontrol advisor loglevel ssl  443 1
rem call dscontrol advisor logsize  ssl  443 1048576
rem call dscontrol advisor timeout  ssl  443 unlimited
rem
```

Sample advisor

The following is a sample advisor file called **ADV_sample**.

```
/**
 * ADV_sample:  The Load Balancer HTTP advisor
 *
 *
 * This class defines a sample custom advisor for Load Balancer.  Like all
 * advisors, this custom advisor extends the function of the advisor base,
 * called ADV_Base.  It is the advisor base that actually performs most of
 * the advisor's functions, such as reporting loads back to the Load Balancer
 * for use in the Load Balancer's weight algorithm.  The advisor base also
 * performs socket connect and close operations and provides send and receive
 * methods for use by the advisor.  The advisor itself is used only for
 * sending and receiving data to and from the port on the server being
 * advised.  The TCP methods within the advisor base are timed to calculate
 * the load.  A flag within the constructor in the ADV_base overwrites the
 * existing load with the new load returned from the advisor if desired.
 *
 * Note: Based on a value set in the constructor,  the advisor base supplies
 * the load to the weight algorithm at specified intervals.  If the actual
 * advisor has not completed so that it can return a valid load, the advisor
 * base uses the previous load.
 *
 * NAMING
 *
 * The naming convention is as follows:
 *
 *  - The file must be located in the following Load Balancer directory:
 *
 *    lb/servers/lib/CustomAdvisors/ (lb\servers\lib\CustomAdvisors on Windows)
 *
 * - The Advisor name must be preceded with "ADV_".  The advisor can be
 *    started with only the name, however; for instance, the "ADV_sample"
 *    advisor can be started with "sample".
 *
 * - The advisor name must be in lowercase.
 *
```

```
 *  With these rules in mind, therefore, this sample is referred to as:
 *
 *             <base directory>/lib/CustomAdvisors/ADV_sample.class
 *
 *
 * Advisors, as with the rest of Load Balancer, must be compiled with the
 * prereq version of Java.  To ensure access to Load Balancer classes, make
 * sure that the ibmlb.jar file (located in the lib subdirectory of the base
 * directory) is included in the system's CLASSPATH.
 *
 * Methods provided by ADV_Base:
 *
 * - ADV_Base (Constructor):
 *
 *   - Parms
 *     - String sName = Name of the advisor
 *     - String sVersion = Version of the advisor
 *     - int iDefaultPort = Default port number to advise on
 *     - int iInterval = Interval on which to advise on the servers
 *     - String sDefaultName = Unused.  Must be passed in as "".
 *     - boolean replace = True - replace the load value being calculated
 *                                by the advisor base
 *                         False - add to the load value being calculated
 *                                 by the advisor base
 *   - Return
 *     - Constructors do not have return values.
 *
 * Because the advisor base is thread based, it has several other methods
 * available for use by an advisor.  These methods can be referenced using
 * the CALLER parameter passed in getLoad().
 *
 * These methods are as follows:
 *
 * - send - Send a packet of information on the established socket connection
 *          to the server on the specified port.
 *   - Parms
 *     - String sDataString - The data to be sent in the form of a string
 *   - Return
 *     - int RC - Whether the data was sucessfully sent or not: zero indicates
 *                data was sent; a negative integer indicates an error.
 *
 * - receive - Receive information from the socket connection.
 *   - Parms
 *     - StringBuffer sbDataBuffer - The data received during the receive call
 *   - Return
 *     - int RC - Whether the data was successfully received or not; zero
 *                indicates data was sent; a negative integer indicates
 *                an error.
 *
 * If the function provided by the advisor base is not sufficient,
 * you can create the appropriate function within the advisor and
 * the methods provided by the advisor base will then be ignored.
 *
 * An important question regarding the load returned is whether to apply
 * it to the load being generated within the advisor base,
 * or to replace it; there are valid instances of both situations.
 *
 * This sample is essentially the Load Balancer HTTP advisor. It functions
 * very simply: a send request--an http head request--is issued.  Once a
 * response is received, the getLoad method terminates, flagging the advisor
 * base to stop timing the request.  The method is then complete.  The
 * information returned is not parsed; the load is based on the time
 * required to perform the send and receive operations.
 */

package CustomAdvisors;
import com.ibm.internet.nd.advisors.*;
```

```
public class ADV_sample extends ADV_Base implements ADV_MethodInterface
{
  String COPYRIGHT =
            "(C) Copyright IBM Corporation 1997, All Rights Reserved.\n";

  static final String  ADV_NAME              = "Sample";
  static final int     ADV_DEF_ADV_ON_PORT   = 80;
  static final int     ADV_DEF_INTERVAL      = 7;

  // Note: Most server protocols require a carriage return ("\r") and line
  //       feed ("\n") at the end of messages.  If so, include them in
  //       your string here.
  static final String  ADV_SEND_REQUEST      =
    "HEAD / HTTP/1.0\r\nAccept: */*\r\nUser-Agent: " +
    "IBM_Load_Balancer_HTTP_Advisor\r\n\r\n";

  /**
   * Constructor.
   *
   * Parms:  None; but the constructor for ADV_Base has several parameters
   *         that must be passed to it.
   *
   */
  public ADV_sample()
  {
    super( ADV_NAME,
       "2.0.0.0-03.27.98",
           ADV_DEF_ADV_ON_PORT,
           ADV_DEF_INTERVAL,
           "",     // not used
           false);
    super.setAdvisor( this );
  }


  /**
   * ADV_AdvisorInitialize
   *
   * Any Advisor-specific initialization that must take place after the
   * advisor base is started. This method is called only once and is
   * typically not used.
   */
  public void ADV_AdvisorInitialize()
  {
    return;
  }


  /**
   * getLoad()
   *
   * This method is called by the advisor base to complete the advisor's
   * operation, based on details specific to the protocol.  In this sample
   * advisor, only a single send and receive are necessary; if more complex
   * logic is necessary, multiple sends and receives can be issued.  For
   * example, a response might be received and parsed.  Based on the
   * information learned thereby, another send and receive could be issued.
   *
   * Parameters:
   *
   * - iConnectTime - The current load as it refers to the length of time it
   *                  took to complete the connection to the server through
   *                  the specified port.
   *
   * - caller - A reference to the advisor base class where the Load
   *            Balancer-supplied methods are to perform simple TCP requests,
```

```
     *            mainly send and receive.
     *
     * Results:
     *
     * - The load - A value, expressed in milliseconds, that can either be added
     *   to the existing load, or that can replace the existing load, as
     *   determined by the constructor's "replace" flag.
     *
     *   The larger the load, the longer it took the server to respond;
     *   therefore, the lower the weight will become within the Load Balancer.
     *
     *   If the value is negative, an error is assumed.  An error from an
     *   advisor indicates that the server the advisor is trying to reach is not
     *   accessible and has been identified as being down.  Load Balancer will
     *   not attempt to load balance to a server that is down. Load Balancer will
     *   resume load balancing to the server when a positive value is received.
     *
     */
    public int getLoad(int iConnectTime, ADV_Thread caller)
    {
      int iRc;
      int iLoad = ADV_HOST_INACCESSIBLE;  // -1

      // Send tcp request
      iRc = caller.send(ADV_SEND_REQUEST);
      if (iRc >= 0)
      {
        // Perform a receive
        StringBuffer sbReceiveData = new StringBuffer("");
        iRc = caller.receive(sbReceiveData);

        /**
        *  In the normal advisor mode ("replace" flag is false), the load
        *  returned is either 0 or 1 indicating the server is up or down.
        *  If the receive is successful, a load of zero is returned
        *  indicating that the load built within the base advisor is to be used.
        *
        *  Otherwise ("replace" flag is true), return the desired load value.
        */

        if (iRc >= 0)
        {
          iLoad = 0;
        }
      }
      return iLoad;
    }

  } // End - ADV_sample
```

Appendix D. Sample of a 2-tier high availability configuration using Dispatcher, CBR, and Caching Proxy

This appendix describes how to set up a 2-tier, high availability configuration combining the capabilities of two Load Balancer components (the Dispatcher component and the CBR component) along with Caching Proxy.

Server machine set up

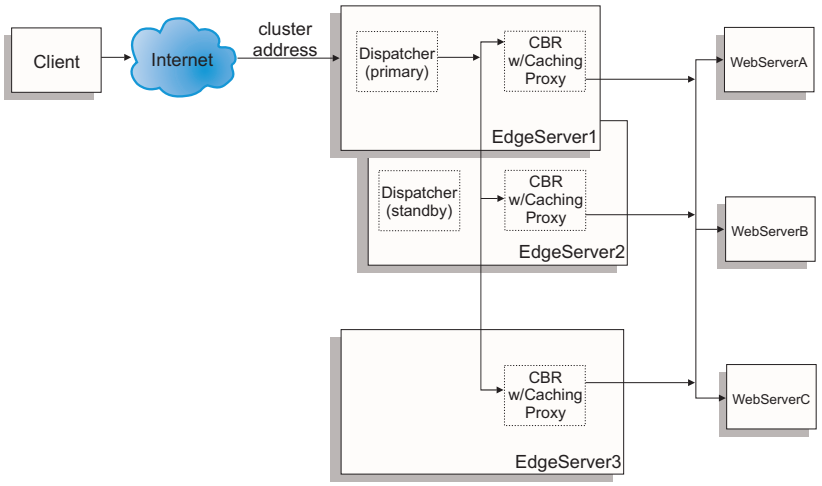


Figure 43. Example of a 2-tier, high availability configuration using Dispatcher, CBR, and Caching Proxy

The server machine set up for Figure 43 is the following:

- EdgeServer1: primary (high availability) Dispatcher machine collocated with CBR and Caching Proxy that load balances across Web servers
- EdgeServer2: standby (high availability) Dispatcher machine collocated with CBR and Caching Proxy
- EdgeServer3: CBR and Caching Proxy machine
- WebServerA, WebServerB, WebServerC: backend Web servers

Figure 43 shows a basic representation of multiple servers (EdgeServer1, EdgeServer2, EdgeServer3) load balancing across multiple backend Web servers. The CBR component uses Caching Proxy to forward requests based on the content of the URL to the backend Web servers. The Dispatcher component is used to load balance the CBR components across the EdgeServers. The high availability feature of the Dispatcher component is used to ensure that requests to the backend servers continue even if the primary high availability machine (EdgeServer1) fails at any time.

Basic Configuration Guidelines:

- Configure Caching Proxy to be the same on all the EdgeServers. To improve the overall accessibility to the Web pages on the backend servers, set up Caching Proxy to do memory caching. This will enable the EdgeServers to cache Web pages that are requested more frequently. For more information on setting up Caching Proxy, refer to the *Caching Proxy Administration Guide*.

- Define the cluster address and ports to be the same in both the CBR and Dispatcher components of Load Balancer.
- Configure the CBR component to be the same across all EdgeServers. Use Web Servers A, B, and C as your servers on the ports you wish to define for the cluster. For more information to configure CBR, see Chapter 10, "Configuring Content Based Routing," on page 75.
- Configure the Dispatcher component to be the same on EdgeServer1 and EdgeServer2. Define all the EdgeServers as your servers on the ports you want to be defined on the cluster to be load balanced by Dispatcher. For more information on how to configure Dispatcher, see Chapter 7, "Configuring Dispatcher," on page 47.
- Configure EdgeServer1 as the primary high availability machine and EdgeServer2 as the standby (backup) high availability machine. For more information, see "High availability" on page 164.

Note:

1. To avoid backend server addresses displayed in the URL on a client, you will need to set the ReversePass directive for each backend server address in the Caching Proxy configuration file.
2. To ensure that Web memory caching is being used effectively, set the "Caching" directive to "ON"' and increase the "CacheMemory" directive to the size required in the Caching Proxy configuration file.
3. Sample lines referred to in notes 1-2 (above):

```
Caching              ON
CacheMemory          128000 K
ReversePass /* http://websrvA.company.com/* http://www.company.com/*
```

4. Remember to alias the cluster address on the network interface card for EdgeServer1 and to alias the cluster address on the loopback device on the remaining EdgeServers.
5. If using the Linux platform for the EdgeServers, you may need to install a patch to the Linux kernel or use an alternative to aliasing the loopback device. For more information, see "Linux loopback aliasing alternatives when using Load Balancer's mac forwarding" on page 60.
6. For CBR, port affinity (stickytime) must not be used when using content rules, otherwise the content rules will not fire while processing requests to the backend Web servers.

Sample Configuration Files:

The following sample configuration files are similar to files that are created when setting up an Edge Components configuration as shown in Figure 43 on page 423. The sample configuration files represent the files for the Dispatcher and CBR components of Load Balancer. In the sample configuration, a single Ethernet adapter is used for each of the EdgeServer machines and all addresses are represented within a private subnet. The sample configuration files use the following IP addresses for the specified machines:

- EdgeServer1 (Primary high availability EdgeServer): 192.168.1.10
- EdgeServer2 (Backup high availability EdgeServer): 192.168.1.20
- EdgeServer3 (Web caching EdgeServer): 192.168.1.30
- Web site cluster address: 192.168.1.11
- WebServersA-C (Backend Web Servers): 192.168.1.71, 192.168.1.72, and 192.168.1.73

Sample Configuration file for Dispatcher component on Primary high availability EdgeServer:

```
dscontrol executor start

dscontrol cluster add 192.168.1.11 primaryhost 192.168.1.10

dscontrol port add 192.168.1.11:80

dscontrol server add 192.168.1.11:80:edgeserver1 address 192.168.1.10

dscontrol server add 192.168.1.11:80:edgeserver2 address 192.168.1.20

dscontrol server add 192.168.1.11:80:edgeserver3 address 192.168.1.30

dscontrol manager start manager.log 10004

dscontrol highavailability heartbeat add 192.168.1.10 192.168.1.20
dscontrol highavailability backup add primary auto 4567
```

Sample Configuration file for CBR component on the EdgeServers:

```
cbrcontrol set loglevel 1
cbrcontrol executor start

cbrcontrol cluster add 192.168.1.11

cbrcontrol port add 192.168.1.11:80

cbrcontrol server add 192.168.1.11:80:webserverA address 192.168.1.71

cbrcontrol server add 192.168.1.11:80:webserverB address 192.168.1.72

cbrcontrol server add 192.168.1.11:80:webserverC address 192.168.1.73

cbrcontrol rule add 192.168.1.11:80:webA_rule type content
  pattern (URI=*WSA*)|(URI=*wsA*) priority 21
cbrcontrol rule useserver 192.168.1.11:80:webA_rule webserverA

cbrcontrol rule add 192.168.1.11:80:webB_rule type content
  pattern (URI=/WS_B*) priority 22
cbrcontrol rule useserver 192.168.1.11:80:webB_rule webserverB

cbrcontrol rule add 192.168.1.11:80:webC_rule type content
  pattern URI=*webC* priority 23
cbrcontrol rule useserver 192.168.1.21:80:webC_rule webserverC
```

Appendix E. Notices

This information was developed for products and services offered in the U.S.A.

IBM may not offer the products, services, or features discussed in this document in other countries. Consult your local IBM representative for information on the products and services currently available in your area. Any reference to an IBM product, program, or service is not intended to state or imply that only that IBM product, program, or service may be used. Any functionally equivalent product, program or service that does not infringe any IBM intellectual property right may be used instead. However, it is the user's responsibility to evaluate and verify the operation of any no-IBM product, program or service.

IBM may have patents or pending patent applications covering subject matter described in this document. The furnishing of this document does not give you any license to these patents. You can send license inquiries, in writing, to:

IBM Director of Licensing
IBM Corporation
500 Columbus Avenue
Thornwood, NY 10594
U.S.A.

For license inquiries regarding double-byte (DBCS) information, contact the IBM Intellectual Property Department in your country or send inquiries, in writing, to:

IBM World Trade Asia Corporation
Licensing
2–31 Roppongi 3–chome, Minato-ku
Tokyo 106, Japan

The following paragraph does not apply to the United Kingdom or any country where such provisions are inconsistent with local law:

INTERNATIONAL BUSINESS MACHINES CORPORATION PROVIDES THIS DOCUMENT "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESS OR IMPLIED, INCLUDING, BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OR CONDITIONS OF NON-INFRINGEMENT, MERCHANTABILITY OR FITNESS FOR A PARTICULAR PURPOSE. Some states do not allow disclaimer of express or implied warranties in certain transactions, therefore, this statement may not apply to you.

This information could include technical inaccuracies or typographical errors. Changes are periodically made to the information herein; these changes will be incorporated in new editions or the document. IBM may make improvements and / or changes in the product(s) and / or the program(s) described in this publication at any time without notice.

Any references in this information to non-IBM Web sites are provided for convenience only and do not in any manner serve as an endorsement of those Web sites. The materials at those Web sites are not part of the materials for this IBM product and use of those Web sites is at your own risk.

IBM may use or distribute any of the information you supply in any way it believes appropriate without incurring any obligation to you.

Licensees of this program who wish to have information about it for the purpose of enabling: (i) the exchange of information between independently created programs and other programs (including this one) and (ii) the mutual use of the information which has been exchanged, should contact:

IBM Corporation
Attn.: G7IA./503.
P.O. Box 12195
3039 Cornwallis Rd.
Research Triangle Park, N.C. 27709-2195
U.S.A.

Such information may be available, subject to appropriate terms and conditions, including in some cases, payment of a fee.

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM International Program License Agreement or any equivalent agreement between us.

Any performance data contained herein was determined in a controlled environment. Therefore, the results obtained in other operating environments may vary significantly. Some measurements may have been made on development-level systems and there is not guarantee that these measurements will be the same on generally available systems. Furthermore, some measurements may have been estimated through extrapolation. Actual results may vary. Users of this document should verify the applicable data for their specific environment.

Information concerning non-IBM products was obtained from the suppliers of those products, their published announcements or other publicly available sources. IBM has not tested those products and cannot confirm the accuracy of performance, compatibility or any other claims related to non-IBM products. Questions on the capabilities of non-IBM products should be addressed to the suppliers of those products.

All statements regarding IBM's future direction or intent are subject to change or withdrawal without notice, and represent goals and objectives only.

This information contains examples of data and reports used in daily business operations. To illustrate them as completely as possible, the examples may include the names of individuals, companies, brands, and products. All of these names are fictitious and any similarity to the names and addresses used by an actual business enterprise is entirely coincidental.

If you are viewing this information softcopy, the photographs and color illustrations many not appear.

Trademarks

The following terms are registered trademarks or trademarks of IBM Corporation in the United States, other countries, or both.

AFS

AIX

DFS
IBM
iSeries®
NetView
OS/2
Redbooks®
RS/6000®
SecureWay
ViaVoice
WebSphere
zSeries®

Java and all Java-based trademarks are trademarks of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft, Windows, Windows NT, and the Windows logo are trademarks of Microsoft Corporation in the United States, other countries, or both.

Intel, Intel Inside (logos), MMX and Pentium are trademarks of Intel Corporation in the United States, other countries, or both.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Linux is a trademark of Linus Torvalds in the United States, other countries, or both.

Other company, product, or service names may be trademarks or service marks of others.

Glossary

A

ACK. A control bit (acknowledge) occupying no sequence space, which indicates that the acknowledgment field of this segment specifies the next sequence number the sender of this segment is expecting to receive, hence acknowledging receipt of all previous sequence numbers.

address. The unique code assigned to each device or workstation connected to a network. A standard IPv4 address is a 32-bit address field containing two parts. The first part is the network address, and the second part is the host number.

advisor. The advisors are a function of the Load Balancer. Advisors collect and analyze feedback from individual servers and inform the manager function.

agent. (1) In systems management, a user that, for a particular interaction, has assumed an agent role. (2) An entity that represents one or more managed objects by (a) emitting notifications regarding the objects and (b) handling requests from managers for management operations to modify or query the objects.

alias. An additional name assigned to a server. The alias makes the server independent of the name of its host machine. The alias must be defined in the domain name server.

API. Application programming interface. The interface (calling conventions) by which an application program accesses operating system and other services. An API is defined at source code level and provides a level of abstraction between the application and the kernel (or other privileged utilities) to ensure the portability of the code.

B

backup. In high availability for the Dispatcher, the partner of the primary machine. It monitors the status of the primary machine and takes over if necessary. See also high availability, primary.

bandwidth. The difference between the highest and lowest frequencies of a transmission channel; the amount of data that can be sent through a given communication circuit per second.

begin range. In rules-based load balancing, a lower value specified on a rule. The default for this value depends on the type of rule.

binary logging. Allows server information to be stored in binary files, and then be processed to analyze the server information that is gathered over time.

C

Caching Proxy. A caching proxy server that can help speed up end-user response time through highly-efficient caching schemes. Flexible PICS filtering helps network administrators control access to Web-based information at one central location.

CBR. Content Based Routing. A component of Load Balancer. CBR works with Caching Proxy to load balance incoming requests, based on Web page content using specified rule types, to HTTP or HTTPS servers.

cbrcontrol. Provides the interface to the Content Based Router component of Load Balancer.

cbrserver. In Content Based Router, handles the requests from the command line to the executor, manager and advisors.

ccocontrol. In Cisco CSS Controller, provides the interface to the Cisco CSS Switch.

ccoserver. In Cisco CSS Controller, handles the requests from the command line to the Consultants.

CGI. Common Gateway Interface. A standard for the exchange of information between a Web server and an external program. The external program can be written in any language supported by the operating system, and performs tasks not usually done by the server, such as forms processing.

CGI script. A CGI program written in a scripting language such as Perl or REXX that uses the Common Gateway Interface to perform tasks not usually done by the server, such as forms processing.

Cisco CSS Controller. A component of IBM Load Balancer. Cisco CSS Controller uses Load Balancer technology to provide real-time load balancing information to the Cisco Content Services Switch.

Cisco CSS Switch. Any of Cisco's CSS 11000 series switches, used for packet forwarding and content routing.

client. A computer system or process that requests a service of another computer system or process. For example, a workstation or personal computer requesting HTML documents from a Lotus Domino® Go Webserver is a client of that server.

cluster. In the Dispatcher, a group of TCP or UDP servers that are used for the same purpose and are identified by a single hostname. See also cell.

cluster address. In the Dispatcher, the address to which clients connect.

clustered server. A server that the Dispatcher groups with other servers into a single, virtual server. Load Balancer balances TCP or UDP traffic among these clustered servers.

collocate. When Load Balancer is installed on the same machine it is load balancing.

consultant. Collects server metrics from the servers that are being load balanced, and sends server weight information to the switch that performs the load balancing.

controller. A collection of one or more consultants.

cross port affinity. Cross port affinity is the affinity (sticky) feature expanded to cover across multiple ports. See also sticky time.

D

daemon. Disk And Execution Monitor. A program that is not involved explicitly, but lies dormant waiting for some condition(s) to occur. The idea is that the perpetrator of the condition need not be aware that a daemon is lurking (though often a program will commit an action only because it knows that it will implicitly invoke a daemon).

default. A value, attribute, or option that is assumed when none is explicitly specified.

destination address. The address of the high availability partner machine to which heartbeats and responses are sent.

Dispatcher. A component of Load Balancer that efficiently balances TCP or UDP traffic among groups of individual linked servers. The Dispatcher machine is the server running the Dispatcher code.

domain name server. DNS. A general-purpose distributed, replicated, data query service chiefly used on Internet for translating hostnames into Internet addresses. Also, the style of hostname used on the Internet, though such a name is properly called a fully qualified domain name. DNS can be configured to use a sequence of name servers, based on the domains in the name being looked for, until a match is found.

dotted-decimal notation. The syntactical representation for a 32–bit integer that consists of four 8–bit numbers, written in base 10 and separated by periods (dots). It is used to represent IPv4 addresses.

dscontrol. Provides the interface to the Dispatcher component of Load Balancer.

dsserver. In Dispatcher, handles the requests from the command line to the executor, manager, and advisors.

E

end range. In rules-based load balancing, a higher value specified on a rule. The default for this value depends on the type of rule.

Ethernet. A standard type of local area network (lan). It allows multiple stations to access the transmission medium at will without prior coordination, avoids contention by using carrier sense and deference, and resolves contention by using collision detection and transmission. Software protocols used by Ethernet systems vary, but include TCP/IP.

executor. One of several Load Balancer functions. The executor routes requests to the TCP or UDP servers, and also monitors the number of new, active, and finished connections and does garbage collection of completed or reset connections. The executor supplies the new and active connections to the manager function.

F

FIN. A control bit (finis) occupying one sequence number, which indicates that the sender will send no more data or control occupying sequence space.

FIN state. The status of a transaction that has finished. When a transaction is in FIN state, the Load Balancer garbage collector can clear the memory reserved for the connection.

Firewall. A computer that connects a private network, such as a business, to a public network, such as the Internet. It contains programs that limit the access between two networks. See also *proxy gateway*.

FQDN. Fully Qualified Domain Name. The full name of a system, consisting of its local hostname and its domain name, including a top-level domain (tld). For example, "venera" is a hostname and "venera.isi.edu" is an FQDN. An FQDN should be sufficient to determine a unique Internet address for any host on the Internet. This process, called "name resolution", uses the Domain Name System (DNS).

FTP (File Transfer Protocol). An application protocol used for transferring files to and from network computers. FTP requires a user ID and sometimes a password to allow access to files on a remote host system.

G

gateway. A functional unit that interconnects two computer networks with different architectures.

GRE. Generic Routing Encapsulation. A protocol which allows an arbitrary network protocol A to be transmitted over any other arbitrary protocol B, by encapsulating the packets of A within GRE packets, which in turn are contained within packets of B.

H

heartbeat. A simple packet sent between two Load Balancer machines in high availability mode used by the standby Load Balancer to monitor the health of the active Load Balancer.

high availability. A Load Balancer feature in which one Load Balancer can take over the function of another, should that part fail.

host. A computer, connected to a network, that provides an access point to that network. A host can be a client, a server, or both simultaneously.

host name. The symbolic name assigned to a host. Host names are resolved to IP addresses through a domain name server.

HTML (Hypertext Markup Language). The language used to create hypertext documents. Hypertext documents include links to other documents that contain additional information about the highlighted term or subject. HTML controls the format of text and position of form input areas, for example, as well as the navigable links.

HTTP (Hypertext Transfer Protocol). The protocol used to transfer and display hypertext documents.

HTTPS (Hypertext Transfer Protocol, Secure). The protocol used to transfer and display hypertext documents using SSL.

I

ICMP. Internet Control Message Protocol. A message control and error-reporting protocol between a host server and a gateway to the Internet.

IMAP. Internet Message Access Protocol. A protocol allowing a client to access and manipulate electronic mail messages on a server. It permits manipulation of remote message folders (mailboxes), in a way that is functionally equivalent to local mailboxes.

Internet. The worldwide collection of interconnected networks that use the Internet suite of protocols and permit public access.

intranet. A secure, private network that integrates Internet standards and applications (such as Web browsers) with an organization's existing computer networking infrastructure.

IP. Internet Protocol. A connectionless protocol that routes data through a network or interconnected networks. IP acts as an intermediary between the higher protocol layers and the physical layer.

IP address. Internet Protocol address. The unique address that specifies the actual location of each device or workstation in a network. It is also known as an Internet address.

IPSEC. Internet Protocol Security. A developing standard for security at the network or packet processing layer of network communication.

L

LAN. Local Area Network. A computer network of devices connected within a limited geographical area for communication and which can be connected to a larger network.

loopback alias. An alternative IP address associated with the loopback interface. The alternative address has the useful side affect of not advertising on a real interface.

loopback interface. An interface that bypasses unnecessary communications functions when the information is addressed to an entity within the same system.

M

MAC address. Media Access Control address. The hardware address of a device connected to a shared network medium.

managed node. In Internet communications, a workstation, server, or router that contains a network management agent. In the Internet Protocol (IP), the managed node usually contains a Simple Network Management Protocol (SNMP) agent.

manager. One of several Load Balancer functions. The manager sets weights based on internal counters in the executor and feedback provided by the advisors. The executor then uses the weights to perform load balancing.

mark down. To break all active connections to a server and stop any new connections or packets from being sent to that server.

mark up. To allow a server to receive new connections.

metric. A process or command that returns a numeric value that can be used in load balancing on the network, for example, the number of users currently logged on.

metric address. The address where the metric server connects.

metric collector. Resides in the consultant and is responsible for collecting a metric or metrics.

Metric Server. Formerly known as Server Monitor Agent (SMA). Metric server provides system specific metrics to the Load Balancer manager.

MIB. (1) Management Information Base. A collection of objects that can be accessed by means of a network management protocol. (2) A definition for management information that specifies the information available from a host or gateway and the operations allowed.

multiple address collocation. Multiple address collocation allows the customer to specify the address of the collocated server to be different than the nonforwarding address (NFA) in the configuration. See also collocate.

mutual high availability. Mutual high availability allows two Dispatcher machines to be both primary and backup for each other. See also backup, high availability, primary.

N

nalcontrol. Provides the interface to the Nortel Alteon Controller component of Load Balancer.

nalserver. In Nortel Alteon Controller, handles the requests from the command line to the Consultants.

netmask. For IPv4, a 32–bit mask used to identify the subnetwork address bits in the host portion of an IP address.

network. Hardware and software data communication system. Networks are often classified according to their geographical extent, local area network (LAN), metropolitan area network (MAN), wide area network (WAN) and also according to the protocols used.

Network Address Translation. NAT, or Network Address Translator, Virtual LAN. A hardware device currently being developed and used to extend the Internet addresses already in use. It allows duplicate IP addresses to be used within a corporation and unique addresses outside.

Network Address Port Translation. NAPT, also known as port mapping. This allows you to configure multiple server daemons within one physical server to listen on different port numbers.

network management station. In the Simple Network Management Protocol (SNMP), a station that runs management application programs that monitor and control network elements.

network proximity. The proximity of two networked entities, such as a client and server, which Site Selector determines by measuring round-trip time.

nfa (nonforwarding address). The primary IP address of the Load Balancer machine, used for administration and configuration.

NIC. Network Interface Card. An adapter circuit board installed in a computer to provide a physical connection to a network.

NNTP. Network News Transfer Protocol. A TCP/IP protocol for transferring news items.

Nortel Alteon Controller. A component of IBM Load Balancer. Nortel Alteon Controller uses Load Balancer technology to provide real-time load balancing information to the Nortel Alteon Web Switch.

Nortel Alteon Web Switch. The Nortel Alteon ACE Director Series Switch and the Nortel Alteon 180 Series Switch from the Alteon Web Switching portfolio, used for packet forwarding and content routing.

O

owner content. Represents the owner name and the content rule for an owner, which are both defined on the Cisco CSS Switch.

P

packet. The unit of data that is routed between an origin and a destination on the Internet or any other packet-switched network.

PICS. Platform for Internet Content Selection. PICS-enabled clients allow the users to determine which rating services they want to use and, for each rating service, which ratings are acceptable and which are unacceptable.

ping. A command that sends Internet Control Message Protocol (ICMP) echo-request packets to a host, gateway, or router with the expectation of receiving a reply.

POP3. Post Office Protocol 3. A protocol used for exchanging network mail and accessing mailboxes.

port. A number that identifies an abstracted communication device. Web servers use port 80 by default.

primary. In high availability for the Dispatcher, the machine that starts out as the machine actively routing packets. Its partner, the backup machine, monitors the status of the primary machine and takes over if necessary. See also backup, high availability.

priority. In rules-based load balancing, the level of importance placed upon any given rule. The Dispatcher evaluates rules from the first priority level to the last priority level.

private network. A separate network on which Dispatcher communicates with clustered servers for performance reasons.

protocol. The set of rules governing the operation of functional units of a communication system if communication is to take place. Protocols can determine low-level details of machine-to-machine interfaces, such as the order in which bits from a byte are sent; they can also determine high-level exchanges between application programs, such as file transfer.

Q

Quality of Service (QoS). The performance properties of a network service, including throughput, transit delay and priority. Some protocols allow packets or streams to include QoS requirements.

quiesce. To end a process by allowing operations to complete normally.

R

reach. In Dispatcher, an advisor that issues pings to a given target and reports whether that target is responding.

reach address. In high availability for the Dispatcher, the address of the target to which the advisor should issue pings to see if the target is responding.

return address. A unique IP address or hostname. It is configured on the Dispatcher machine and used by Dispatcher as its source address when load balancing the client's request to the server.

RMI. Remote Method Invocation. Part of the Java programming language library which enables a Java program running on one computer to access the objects and methods of another Java program running on a different computer.

root user. The unrestricted authority to access and modify any part of the AIX, Red Hat Linux, or Solaris operating system, usually associated with the user who manages the system.

route. The path of network traffic from origin to destination.

router. A device which forwards packets between networks. The forwarding decision is based on network layer information and routing tables, often constructed by routing products.

RPM. Red Hat Package Manager.

rule. In rules-based load balancing, a mechanism for grouping servers such that a server can be chosen based on information other than the destination address and port.

rule type. In rules-based load balancing, an indicator of the information that should be evaluated to determine whether a rule is true.

S

scalable. Pertaining to the capability of a system to adapt readily to a greater or lesser intensity of use, volume, or demand. For example, a scalable system can efficiently adapt to work with larger or smaller networks performing tasks of varying complexity.

server. A computer that provides shared services to other computers over a network; for example, a file server, a print server, or a mail server.

server address. The unique code assigned to each computer that provides shared services to other computers over a network; for example, a file server, a print server, or a mail server. The server address can be either the IP address or the host name.

server machine. A server that the Dispatcher groups with other servers into a single, virtual server. The Dispatcher balances traffic among the server machines. Synonymous with clustered server.

service. (1) A function provided by one or more nodes; for example, HTTP, FTP, Telnet. (2) For Nortel Alteon Controller, a service is the function or information requested by an end user from a site. It is identified by a virtual IP address and a virtual port number on an end user request. On the switch it is identified by a virtual server identifier which is an integer and a virtual port number or service name. (3) For Cisco CSS Consultant, a service is a destination location where a piece of content physically resides. For example, a local or remote server and port.

shell. The software that accepts and processes command lines from a user's workstation. The bash shell is one of several UNIX shells available.

site name. A site name is an unresolvable host name that the client will request. For example, a web site has 3 servers (1.2.3.4, 1.2.3.5, and 1.2.3.6) configured for site name *www.dnsload.com*. When a client requests this site name, one of the three server IP addresses will be returned as the resolution. The site name must be a fully qualified domain name, for example: *dnsload.com*. An unqualified name, for example, *dnsload* is invalid for a site name.

Site Selector. A DNS-based load balancing component of Load Balancer. Site Selector balances the load on servers within a wide area network (WAN) using measurements and weights that are gathered from the Metric Server component running on those servers.

SMTP. Simple Mail Transfer Protocol. In the Internet suite of protocols, an application protocol for transferring mail among users in the Internet environment. SMTP specifies the mail exchange sequences and message format. It assumes that the Transmission Control Protocol (TCP) is the underlying protocol.

SNMP. Simple Network Management Protocol. The Internet standard protocol, defined in STD 15, RFC 1157, developed to manage nodes on an IP network. SNMP is not limited to TCP/IP. It can be used to manage and monitor all sorts of equipment including computers, routers, wiring hubs, toasters and jukeboxes.

source address. In high availability for the Dispatcher, the address of the high availability partner machine that sends heartbeats.

SPARC. Scalable processor architecture.

sscontrol. Provides the interface to the Site Selector component of Load Balancer.

SSL. Secure Sockets Layer. A popular security scheme developed by Netscape Communications Corp. along with RSA Data Security Inc. SSL allows the client to authenticate the server and all data and requests to be encrypted. The URL of a secure server protected by SSL begins with https (rather than HTTP).

ssserver. In Site Selector, handles the requests from the command line to the site name, manager and advisors.

sticky time. The interval between the closing of one connection and the opening of a new connection during which a client will be sent back to the same server used during the first connection. After the sticky time, the client may be sent to a server different from the first.

strategy. In high availability for the Dispatcher, a keyword for specifying how recovery takes place following the failure of the active machine.

subnet mask. For IPv4, a 32–bit mask used to identify the subnetwork address bits in the host portion of an IP address.

SYN. A control bit in the incoming segment, occupying one sequence number, used at the initiation of a connection, to indicate where the sequence numbering will start.

T

TCP. Transmission Control Protocol. A communications protocol used on the Internet. TCP provides reliable host-to-host exchange of information. It uses IP as the underlying protocol.

TCP/IP . Transmission Control Protocol/Internet Protocol. A suite of protocols designed to allow communication between networks regardless of the communication technologies used in each network.

TCP server machine. A server that Load Balancer links with other servers into a single, virtual server. Load Balancer balances TCP traffic among the TCP server machines. Synonymous with clustered server.

Telnet. Terminal emulation protocol, a TCP/IP application protocol for remote connection service. Telnet allows a user at one site to gain access to a remote host as if the user's workstation were connected directly to that remote host.

timeout. The time interval allotted for an operation to occur.

TOS. Type of service. A one byte field in the IP header of the SYN packet.

TTL. A DNS TTL (time to live) is the number of seconds a client can cache the name resolution response.

U

UDP. User Datagram Protocol. In the Internet suite of protocols, a protocol that provides unreliable, connectionless datagram service. It enables an application program on one machine or process to send a datagram to an application program on another machine or process. UDP uses the Internet Protocol (IP) to deliver datagrams.

URI. Universal Resource Identifier. The encoded address for any resource on the Web, such as HTML document, image, video clip, program, and so forth.

URL. Uniform Resource Locator. A standard way of specifying the location of an object, typically a web page, on the Internet. URLs are the form of address used on the World-Wide Web. They are used in HTML documents to specify the target of a hyperlink which is often another HTML document (possibly stored on another computer).

V

VPN. Virtual Private Network (VPN). A network comprised of one or more secure IP tunnels connecting two or more networks.

W

WAN. Wide Area Network. A network that provides communication services to a geographic area larger than that served by a local area network or a metropolitan area network, and that may use or provide public communication facilities.

WAP. Wireless Application Protocol. An open international standard for applications that use wireless communication, e.g. Internet access from a mobile phone.

WAS. WebSphere Application Server.

Web. The network of HTTP servers that contain programs and files, many of them hypertext documents that contain links to other documents on HTTP servers. Also World Wide Web.

wizard. A dialog within an application that uses step-by-step instructions to guide a user through a specific task.

WLM. Workload Manager. An advisor provided with Dispatcher. It is designed to work only in conjunction with servers on OS/390 mainframes running the MVS Workload Manager (WLM) component.

Index

D

N

O

P

Q

R

S

T

U

V

W

Printed in USA

Spine information:

WebSphere Edge Components Load Balancer for IPv4 Administration Guide Version 8.0