

WebSphere™ Edge Server for Multiplatforms



Network Dispatcher Administration Guide

Version 2.0

WebSphere™ Edge Server for Multiplatforms



Network Dispatcher Administration Guide

Version 2.0

Note

Before using this information and the product it supports, be sure to read the general information under "Appendix I. Notices" on page 365.

Seventh Edition (September 2001)

© Copyright International Business Machines Corporation 2001. All rights reserved.

US Government Users Restricted Rights – Use, duplication or disclosure restricted by GSA ADP Schedule Contract with IBM Corp.

Contents

Tables	ix	Overview of Dispatcher component	32
Figures	xi	Overview of Content Based Routing (CBR) component	35
Welcome	xiii	Overview of Mailbox Locator component	37
How to send your comments	xiii	Overview of Site Selector component.	38
Chapter 1. Getting started...quickly!	1	Overview of Consultant for Cisco CSS Switches component	40
What you will need?	2	How about high availability?	42
How do you prepare?	2	Dispatcher	42
Configuring the Dispatcher component	3	CBR, Mailbox Locator, Site Selector	42
Configuring using the command line	3	Chapter 4. Planning for the Dispatcher component	43
Configuring using the configuration wizard	4	Hardware and software requirements	43
Configuring using the graphical user interface (GUI)	5	Planning considerations	43
Testing your configuration	7	High availability	45
Types of cluster, port, server configurations	7	Simple high availability	45
Chapter 2. Installing Network Dispatcher	11	Mutual high availability	46
Requirements for AIX	12	Dispatcher's MAC-level routing (mac forwarding method)	47
Installing for AIX	13	Dispatcher's NAT/NAPT (nat forwarding method)	47
Before you install	14	Dispatcher's content-based routing (cbr forwarding method)	49
Installation steps	14	Chapter 5. Configuring the Dispatcher component	53
Requirements for Red Hat Linux or SuSE Linux	16	Overview of configuration tasks	53
Installing for Linux	17	Methods of configuration	53
Before you install	17	Command line	53
Installation steps	17	Scripts	54
Requirements for Solaris	19	GUI	54
Installing for Solaris	19	Configuration Wizard	55
Before you install	20	Setting up the Dispatcher machine	56
Installation steps	20	Step 1. Start the server function	57
Requirements for Windows 2000	21	Step 2. Start the executor function.	58
Installing for Windows 2000.	22	Step 3. Define the nonforwarding address (if different from hostname).	58
Installation Packages	22	Step 4. Define a cluster and set cluster options.	58
Before you install	22	Step 5. Alias the network interface card	58
Installation steps	22	Step 6. Define ports and set port options	60
Chapter 3. Introducing Network Dispatcher	25	Step 7. Define load-balanced server machines	60
What is Network Dispatcher?	25		
Why do I need Network Dispatcher?.	26		
What are the new features?	27		
What are the components of Network Dispatcher?	32		

Step 8. Start the manager function (optional)	61	CBR configuration example	85
Step 9. Start the advisor function (optional)	61	Chapter 8. Planning for the Mailbox	
Step 10. Set cluster proportions as required	61	Locator component	87
Setting up server machines for load balancing	62	Hardware and software requirements	87
Step 1. Alias the loopback device	62	Planning considerations	87
Step 2. Check for an extra route	64	Using the affinity feature.	89
Step 3. Delete any extra route	65	Overriding the POP3/IMAP inactivity timer	89
Step 4. Verify server is properly configured	65	Chapter 9. Configuring the Mailbox Locator	
Installing the Linux kernel patch (to suppress arp responses on the loopback interface)	66	component	91
Chapter 6. Planning for the Content Based Routing component	71	Overview of configuration tasks	91
Hardware and software requirements	71	Methods of configuration	91
Planning considerations	71	Command line	92
Load balancing across fully secure (SSL) connections	73	Scripts	92
Load balancing client-to-proxy in SSL and proxy-to-server in HTTP	74	GUI.	93
Chapter 7. Configuring the Content Based Routing component	75	Configuration wizard	94
Overview of configuration tasks	75	Setting up the Mailbox Locator machine	94
Methods of configuration	75	Step 1. Start the server function	95
Command line	76	Step 2. Define a cluster and set cluster options.	95
Scripts	77	Step 3. Define ports and set port options	95
GUI.	78	Step 4. Define load balanced server machines	96
Configuration wizard	79	Step 5. Start the manager function (optional)	96
Setting up the CBR machine	79	Step 6. Start the advisor function (optional)	96
Step 1. Configure Caching Proxy to use CBR.	80	Step 7. Set cluster proportions as required	96
Step 2. Start the server function	82	Chapter 10. Planning for the Site Selector component	97
Step 3. Start the executor function.	82	Hardware and software requirements	97
Step 4. Define a cluster and set cluster options.	82	Planning Considerations	97
Step 5. Alias the network interface card (optional)	83	TTL considerations	100
Step 6. Define ports and set port options	83	Using the Network Proximity feature	100
Step 7. Define load balanced server machines	84	Chapter 11. Configuring the Site Selector component	103
Step 8. Add rules to your configuration	84	Overview of configuration tasks	103
Step 9. Add servers to your rules	84	Methods of configuration	103
Step 10. Start the manager function (optional)	84	Command line.	104
Step 11. Start the advisor function (optional)	84	Scripts	104
Step 12. Set cluster proportions as required	85	GUI	105
Step 13. Start Caching Proxy	85	Configuration wizard	106
		Setting up the Site Selector machine	106
		Step 1. Start the server function	106
		Step 2. Start the Name Server.	107
		Step 3. Define a site name and set site name options	107

Step 4. Define load balanced server machines.	107	Using scripts to generate an alert or record server failure	126
Step 5. Start the manager function (optional)	107	Advisors	126
Step 6. Start the advisor function (optional)	107	How advisors work	127
Step 7. Define system metric (optional)	108	Starting and stopping an advisor.	127
Step 8. Set site name proportions as required	108	Advisor intervals	128
Setting up server machines for load balancing	108	Advisor report timeout	128
Chapter 12. Planning for the Consultant for Cisco CSS Switches component.	109	Advisor connect timeout and receive timeout for servers	129
Hardware and software requirements	109	List of advisors	129
Planning considerations.	109	Create custom (customizable) advisors	131
Chapter 13. Configuring the Consultant for Cisco CSS Switches component.	113	WebSphere Application Server advisor	132
Overview of configuration tasks	113	Naming Convention	132
Methods of configuration	113	Compilation	133
Command line	114	Run	133
Scripts	114	Required routines	134
GUI	115	Search order	134
Setting up the Consultant for Cisco CSS Switches machine	116	Naming and path	134
Step 1. Start the server function	116	Sample advisor	135
Step 2. Configure the executor function	116	Workload Manager advisor	135
Step 3. Define a cluster and set cluster options	116	Metric Server Restriction	136
Step 4. Define ports and set port options	116	Metric Server	136
Step 5. Define load-balanced server machines.	117	WLM Restriction	136
Step 6. Start the manager function	117	Prerequisites	136
Step 7. Start the advisor function (optional)	117	How to Use Metric Server	136
Step 8. Set cluster proportions as required	117	Server Partitioning: logical servers configured to one physical server (IP address)	138
Step 9. Start Metric Server (optional)	118	HTTP advisor request/response (URL) option	140
Testing your configuration	118	Using collocated servers	140
Chapter 14. Advanced Network Dispatcher Functions	119	For the Dispatcher component	141
Optimizing the load balancing provided by Network Dispatcher	122	For the CBR component.	142
Proportion of importance given to status information	122	For the Mailbox Locator component	142
Weights	123	For the Site Selector component	142
Manager intervals.	124	For the Cisco Consultant component	142
Sensitivity threshold	125	Configure wide area Dispatcher support	142
Smoothing index	125	Command Syntax	144
		Using remote advisors with wide area support	144
		Configuration example	146
		Notes	148
		GRE (Generic Routing Encapsulation) support	148
		Using Self Advisor in a two-tiered WAND configuration	149
		High availability	150
		Configure high availability.	151
		Failure detection capability using heartbeat and reach target	153

Recovery Strategy	154
Using scripts	155
Configure rules-based load balancing	157
How are rules evaluated?	158
Using rules based on the client IP address	159
Using rules based on the time of day	159
Using rules based on the connections per second on a port	160
Using rules based on the active connections total on a port	160
Using rules based on the client port	161
Using rules based on type of service (TOS)	161
Using rules based on reserved bandwidth and shared bandwidth	161
Metric all rule	163
Metric average rule	164
Using rules that are always true	164
Using rules based on the request content	165
Adding rules to your configuration	165
Server evaluation option for rules	165
Using explicit linking	167
Using a private network configuration	167
Use wildcard cluster to combine server configurations	168
Use wildcard cluster to load balance firewalls	169
Use wildcard cluster with Caching Proxy for transparent proxy	170
Use wildcard port to direct unconfigured port traffic	170
How affinity feature for Network Dispatcher works	170
Behavior when disable affinity	171
Behavior when enable affinity	171
Server Directed Affinity API to control client-server affinity	171
Cross port affinity	172
Affinity address mask	173
Rule affinity override	174
Quiesce handling for sticky connections	174
Affinity option on the rule	175
Active cookie affinity	175
Passive cookie affinity	177
URI affinity	177
Denial of service attack detection	178
Using binary logging to analyze server statistics	180
Additional information on advanced Cisco Consultant functions	182

Cisco Consultant weights	183
------------------------------------	-----

Chapter 15. Operating and managing Network Dispatcher	185
Remote Authenticated Administration	185
Using Network Dispatcher logs	187
Changing the log file paths	188
Using the Dispatcher component	188
Starting and Stopping Dispatcher	188
Using stale timeout value	188
Using FIN count to control garbage collection	189
Reporting GUI — the Monitor menu option	190
Using Simple Network Management Protocol with the Dispatcher component	190
Using ipchains or iptables to reject all traffic to (harden) the Network Dispatcher box (on Linux)	195
Using the Content Based Routing component	195
Starting and Stopping CBR	196
Controlling CBR	196
Using CBR logs	196
Using the Mailbox Locator component	196
Starting and stopping Mailbox Locator	196
Controlling Mailbox Locator	196
Using Mailbox Locator logs	197
Using the Site Selector component	197
Starting and stopping Site Selector	197
Controlling Site Selector	197
Using Site Selector logs	197
Using the Cisco Consultant component	197
Starting and stopping Cisco Consultant	197
Controlling Cisco Consultant	197
Using Cisco Consultant logs	198
Using the Metric Server component	198
Starting and stopping Metric Server	198
Using Metric Server logs	198

Chapter 16. Troubleshooting	199
Troubleshooting tables	199
Checking Dispatcher port numbers	204
Checking CBR port numbers	205
Checking Mailbox Locator port numbers	205
Checking Site Selector port numbers	206
Checking Cisco Consultant port numbers	206
Solving common problems—Dispatcher	207
Problem: Dispatcher will not run	207
Problem: Dispatcher and server will not respond	207

Problem: Dispatcher requests are not being balanced.	207	Problem: Requests not being load balanced.	214
Problem: Dispatcher high-availability function is not working.	208	Problem: On Solaris, cbrcontrol executor start command fails	215
Problem: Unable to add heartbeat (Windows 2000)	208	Problem: Syntactical or configuration error	215
Problem: Extra routes (Windows 2000)	208	Solving common problems—Mailbox Locator	215
Problem: Advisors not working correctly	208	Problem: Mailbox Locator will not run	215
Problem: SNMPD does not run correctly (Windows 2000)	208	Problem: The mlserver command is stopped	215
Problem: Dispatcher, Microsoft IIS, and SSL do not work (Windows 2000)	208	Problem: mlcontrol or ndadmin command fails	216
Problem: Dispatcher connection to a remote machine	209	Problem: Unable to add a port	216
Problem: ndcontrol or ndadmin command fails	209	Problem: Receive proxy error when trying to add a port	216
Problem: "Cannot find the file..." error message when trying to view online Help (Windows 2000)	210	Solving common problems—Site Selector	216
Problem: Spurious error message when starting ndserver on Solaris 2.7	210	Problem: Site Selector will not run	216
Problem: Graphical user interface (GUI) does not start correctly	210	Problem: Site Selector doesn't round-robin traffic from Solaris clients	217
Problem: Error running Dispatcher with Caching Proxy installed.	210	Problem: sscontrol or ndadmin command fails	217
Problem: Graphical user interface (GUI) does not display correctly	210	Problem: ssserver is failing to start on Windows 2000	217
Problem: On Windows 2000, help windows sometimes disappear behind other open windows	211	Problem: Site Selector with duplicate routes not load balancing correctly	217
Problem: Network Dispatcher cannot process and forward a frame	211	Solving common problems—Consultant for Cisco CSS Switches	218
Problem: A blue screen displays when you start the Network Dispatcher executor	211	Problem: lbcserver will not start	218
Problem: Path to Discovery prevents return traffic with Network Dispatcher. . .	211	Problem: lbccontrol or ndadmin command fails	218
Problem: Advisors show that all servers are down	212	Problem: Cannot create registry on port 14099	218
Problem: High availability in the Wide Area mode of Network Dispatcher does not work.	213	Solving common problems—Metric Server	218
Problem: GUI hangs (or unexpected behavior) when trying to load a large configuration file	213	Problem: Metric Server IOException on Windows 2000 running .bat or .cmd user metric files	218
Solving common problems—CBR	214	Problem: Metric Server not reporting loads to Network Dispatcher machine . . .	219
Problem: CBR will not run.	214	Problem: Metric Server log reports "Signature is necessary for access to agent".	219
Problem: cbrcontrol or ndadmin command fails	214		
		Appendix A. How to read a syntax diagram.	221
		Symbols and punctuation	221
		Parameters	221
		Syntax examples	222
		Appendix B. Command reference for Dispatcher, CBR, and Mailbox Locator . . .	225

Configuration differences between CBR, Mailbox Locator, and Dispatcher	226
ndcontrol advisor — control the advisor	228
ndcontrol cluster — configure clusters	234
ndcontrol executor — control the executor	239
ndcontrol file — manage configuration files	244
ndcontrol help — display or print help for this command	246
ndcontrol highavailability — control high availability	248
ndcontrol host — configure a remote machine	252
ndcontrol log — control the binary log file	253
ndcontrol manager — control the manager	254
ndcontrol metric — configure system metrics	260
ndcontrol port — configure ports	261
ndcontrol rule — configure rules.	267
ndcontrol server — configure servers	274
ndcontrol set — configure server log	280
ndcontrol status — display whether the manager and advisors are running	281
ndcontrol subagent — configure SNMP subagent.	282

Appendix C. Content rule (pattern) syntax	285
Content rule (pattern) syntax:	285
Reserved keywords	285

Appendix D. Command reference for Site Selector	289
sscontrol advisor — control the advisor	290
sscontrol file — manage configuration files	295
sscontrol help — display or print help for this command	297
sscontrol manager — control the manager	298
sscontrol metric — configure system metrics	303
sscontrol nameserver — control the NameServer	304
sscontrol rule — configure rules	305
sscontrol server — configure servers	308
sscontrol set — configure server log.	310
sscontrol sitename — configure a sitename	311
sscontrol status — display whether the manager and advisors are running	314

Appendix E. Command reference for Consultant for Cisco CSS Switches	315
lbcontrol advisor — control the advisor	316
lbcontrol cluster — configure clusters	321
lbcontrol executor — control the executor	323
lbcontrol file — manage configuration files	325
lbcontrol help — display or print help for this command	327
lbcontrol host — configure a remote machine	328
lbcontrol log — control the binary log file	329
lbcontrol manager — control the manager	330
lbcontrol metric — configure system metrics	336
lbcontrol port — configure ports	338
lbcontrol server — configure servers	340
lbcontrol set — configure server log	342
lbcontrol status — display whether the manager and advisors are running	343

Appendix F. Sample configuration files	345
Sample Network Dispatcher configuration files	345
Dispatcher Configuration file—AIX, Red Hat Linux, and Solaris	345
Dispatcher Configuration file—Windows	349
Sample advisor	352

Appendix G. Sample of a 2-tier high availability configuration using Dispatcher, CBR, and Caching Proxy	359
Server machine set up	359

Appendix H. Other resources	363
Command line access	363
Getting online help	363
Reference information	363

Appendix I. Notices	365
Trademarks	366

Glossary	369
-----------------	------------

Index	379
--------------	------------

Tables

1. AIX installp images	13	11. Example of the Cisco CSS Switch configuration mapped to the Consultant configuration	112
2. AIX install commands	15	12. Configuration tasks for the Consultant for Cisco CSS Switches component	113
3. Configuration tasks for the Dispatcher function	53	13. Advanced configuration tasks for the Network Dispatcher	119
4. Commands to alias the loopback device (lo0) for Dispatcher	62	14. Dispatcher troubleshooting table	199
5. Commands to delete any extra route for Dispatcher	65	15. CBR Troubleshooting table	201
6. Configuration tasks for the CBR component	75	16. Mailbox Locator Troubleshooting table	202
7. Commands to alias the NIC	83	17. Site Selector troubleshooting table	203
8. Configuration tasks for the Mailbox Locator component	91	18. Consultant for Cisco CSS Switches troubleshooting table	203
9. Configuration tasks for the Site Selector component	103	19. Metric Server troubleshooting table	203
10. Consultant and Cisco CSS Switch configuration terms	110		

Figures

1.	A simple local Dispatcher configuration	1
2.	The graphical user interface (GUI)	5
3.	Example of Dispatcher configured with a single cluster and 2 ports	7
4.	Example of Dispatcher configured with two clusters, each with one port	8
5.	Example of Dispatcher configured with 2 clusters, each with 2 ports	9
6.	Example of a physical representation of a site using Dispatcher to manage local servers	33
7.	Example of a site using Dispatcher and Metric Server to manage servers	34
8.	Example of a site using Dispatcher to manage local and remote servers	35
9.	Example of a site using CBR to manage local servers	36
10.	Example of a site using Mailbox Locator to manage local servers	37
11.	Example of a site using Site Selector and Metric Server to manage local and remote servers	39
12.	Example of a site using Cisco Consultant and Metric Server to manage local servers	41
13.	Example of a Dispatcher using simple high availability	45
14.	Example of a Dispatcher using mutual high availability	46
15.	Example of the IP addresses needed for the Dispatcher machine	57
16.	CBR configuration file for AIX	81
17.	CBR configuration file for Linux	81
18.	CBR configuration file for Solaris	81
19.	CBR configuration file for Windows 2000	81
20.	Example of a DNS environment	98
21.	Example of Consultant configured with 2 clusters, each with 3 ports	111
22.	Example of a configuration consisting of a single LAN segment	143
23.	Example of configuration using local and remote servers	143
24.	Wide area example configuration with remote Network Dispatchers	146
25.	Wide area example configuration with server platform that supports GRE	149
26.	Example of a two-tiered WAND configuration using the self advisor	150
27.	Example of a private network using Dispatcher	168
28.	SNMP commands for AIX and Solaris	191
29.	SNMP commands for Windows 2000	192
30.	Example of a 2-tier, high availability configuration using Dispatcher, CBR, and Caching Proxy	359

Welcome

This book explains how to plan for, install, configure, use, and troubleshoot IBM® WebSphere™ Edge Server Network Dispatcher for AIX, Linux, Solaris, and Windows 2000. Previously, this product was called SecureWay Network Dispatcher, eNetwork Dispatcher, and Interactive Network Dispatcher.

The most current version of this book is available in HTML and PDF formats on the WebSphere Edge Server Web site. To access the online book, go to the following URL:

<http://www.ibm.com/software/webservers/edgeserver/library.html>

The WebSphere Edge Server Web site gives you the latest details on how you can use Network Dispatcher to maximize the performance of your servers. Configuration examples and scenarios are included. To access this Web site, go to the following URL:

<http://www.ibm.com/software/webservers/edgeserver>

For the most current updates and usage hints about Network Dispatcher, visit the WebSphere Edge Server support Web page and click *Search for Network Dispatcher hints and tips*. To access this Web page, go to the following URL:

<http://www.ibm.com/software/webservers/edgeserver/support.html>

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 WebSphere Edge Server document:

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

Chapter 1. Getting started...quickly!

How quickly can you make Network Dispatcher work for you? Consider the following:

Assume you're the webmaster for the Intersplash Corporation. You manage a local Web site with two HTTP servers. You've been using a round-robin approach to manage the load on the two servers, but business has picked up recently and customers are starting to complain about not being able to access the site. What do you do?

Go out to <http://www.ibm.com/software/webserver/edgeserver> and download the latest version of Network Dispatcher. This product has five components: Dispatcher, Content Based Routing (CBR), Mailbox Locator, Site Selector, and Consultant for Cisco CSS Switches (Cisco Consultant). For the time being, we'll just discuss the **Dispatcher** component.

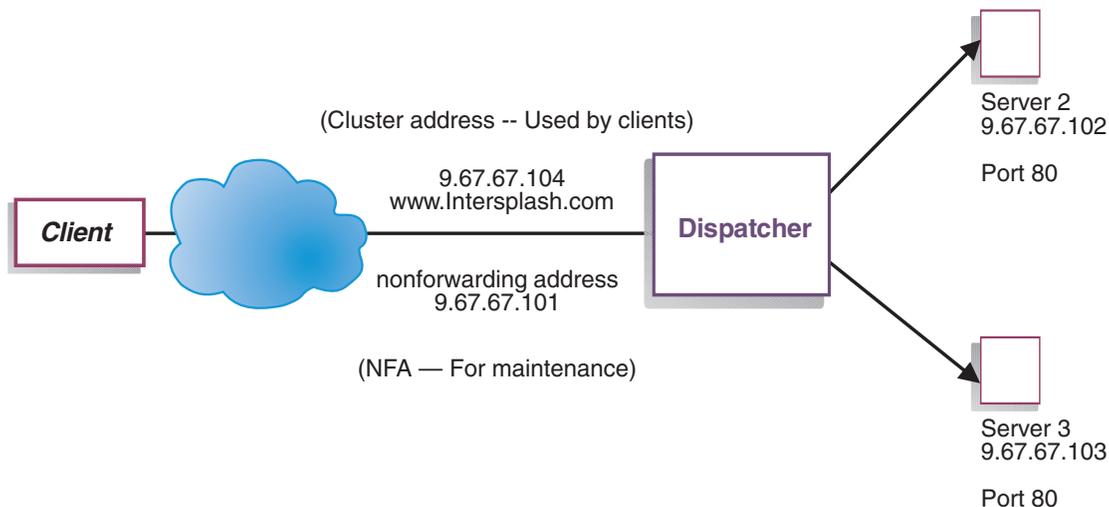


Figure 1. A simple local Dispatcher configuration

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.

Note: With the AIX, Linux, or Solaris version of Dispatcher, the configuration could be completed using only two workstations with Dispatcher located on one of the Web server workstations. This represents a collocated configuration. Procedures for setting up more complex configurations can be found at “Setting up the Dispatcher machine” on page 56.

What you will need?

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

How do you prepare?

1. Ensure you have the prerequisites, listed in “Chapter 2. Installing Network Dispatcher” on page 11.
2. Set up your workstations so that they are on the same LAN segment. Ensure that network traffic between the three machines does not have to pass through any routers or bridges.
3. 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.intersplash.com	9.67.67.101
2	server2.intersplash.com	9.67.67.102
3	server3.intersplash.com	9.67.67.103
Netmask = 255.255.255.0		

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

4. Ensure that server1.intersplash.com can ping both server2.intersplash.com and server3.intersplash.com.
5. Ensure that server2.intersplash.com and server3.intersplash.com can ping server1.intersplash.com.
6. 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.

7. Ensure that Web servers on server2.intersplash.com and server3.intersplash.com are operational. Use a Web browser to request pages directly from **http://server2.intersplash.com** and **http://server3.intersplash.com**.
8. 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.intersplash.com
IP=9.67.67.104
```
9. Configure the two Web server workstations to accept traffic for www.intersplash.com.

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

 - For AIX:


```
ifconfig lo0 alias www.intersplash.com netmask 255.255.255.0
```
 - For Solaris 7:


```
ifconfig lo0:1 www.intersplash.com 127.0.0.1 up
```
 - For other operating systems see Table 4 on page 62.
10. 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 64. 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 using the command line

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

1. Start the ndserver on Dispatcher:
 - For AIX, Linux, or Solaris, run the following command as root user:


```
ndserver
```
 - For Windows 2000, ndserver runs as a service that starts automatically.
2. Start the executor function of Dispatcher:


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


```
ndcontrol cluster add www.intersplash.com
```
4. Add the http protocol port to the Dispatcher configuration:

```
ndcontrol port add www.intersplash.com:80
```

5. Add each of the Web servers to the Dispatcher configuration:

```
ndcontrol server add www.intersplash.com:80:server2.intersplash.com
```

```
ndcontrol server add www.intersplash.com:80:server3.intersplash.com
```

6. Configure the workstation to accept traffic for the cluster address:

```
ndcontrol cluster configure www.intersplash.com
```

7. Start the manager function of Dispatcher:

```
ndcontrol manager start
```

Dispatcher will now do load balancing based on server performance.

8. Start the advisor function of Dispatcher:

```
ndcontrol 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.

Configuring using the configuration wizard

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

1. Start the ndserver on Dispatcher:

- For AIX, Linux, or Solaris, run the following as root user:

```
ndserver
```

- For Windows 2000, ndserver runs as a service that starts automatically.

2. Start the wizard function of Dispatcher, **ndwizard**.

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.

With the configuration wizard, you will see the following panels:

- Introduction to the wizard
- What is going to happen
- Preparing for the setup
- Choosing a host to configure (if necessary)
- Defining a cluster
- Adding a port
- Adding a server
- Starting an advisor
- Server machine setup

Configuring using the graphical user interface (GUI)

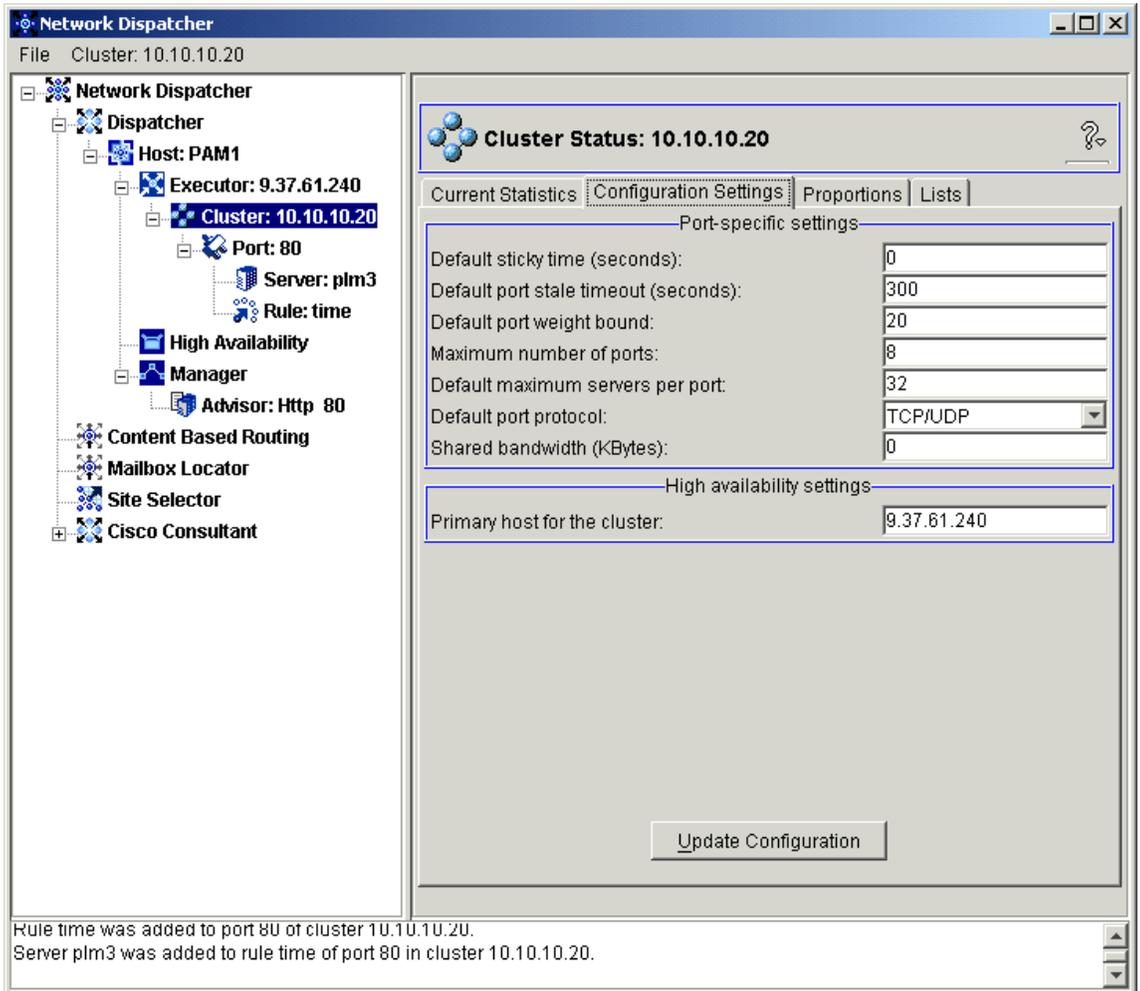


Figure 2. The graphical user interface (GUI)

To start the graphical user interface, follow these steps:

1. Ensure ndserver is running:
 - For AIX, Linux, or Solaris, run the following as root:
ndserver
 - For Windows 2000, ndserver runs as a service that starts automatically.
2. Next, do one of the following:
 - For AIX, Linux, or Solaris, type **ndadmin**.

- For Windows 2000, click **Start**, click **Programs**, click **IBM WebSphere**, click **Edge Server**, click **IBM Network Dispatcher**, and click **Network Dispatcher**.

General Instructions for using the GUI

The left side of the panel displays a tree structure with Network Dispatcher at the top level, and Dispatcher, Content Based Routing, Mailbox Locator, Site Selector, and Cisco Consultant as components. See Figure 2 on page 5.

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.

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.
- 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 14. Advanced Network Dispatcher Functions” on page 119. 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.

To access **Help**, click the question mark in the upper right hand corner of the Network Dispatcher window.

- **Field Help** — describes each field, default values
- **How do I** — lists tasks that can be done from the current screen
- **Contents** — a table of contents for all the Help information

- **Index** — an alphabetical index of the Help topics

Testing your configuration

Test to see if the configuration is working.

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

Types of cluster, port, server configurations

There are many ways that you can configure Network Dispatcher 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 Network Dispatcher communicates. See Figure 3.

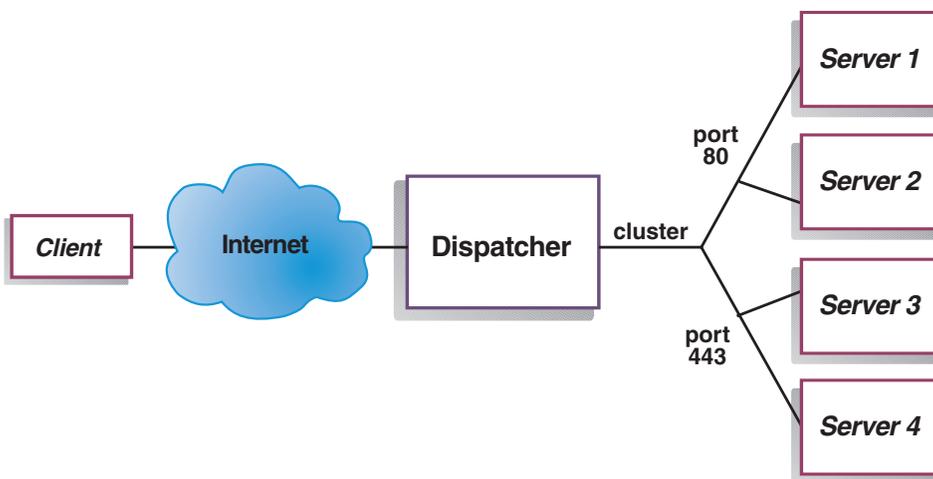


Figure 3. Example of Dispatcher configured with a single cluster and 2 ports

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) would go to a different server than a client requesting `https://www.productworks.com` (port 443).

Another way of configuring Network Dispatcher 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 4.

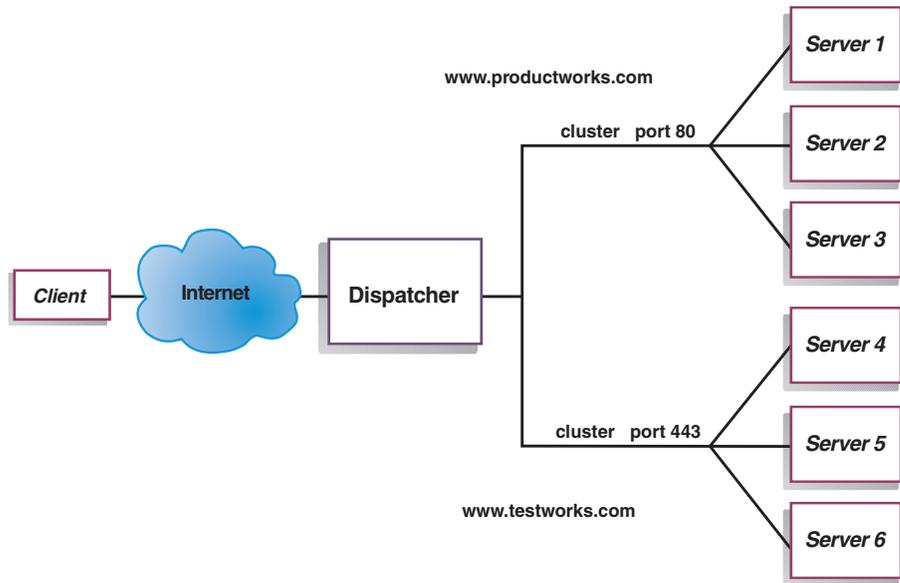


Figure 4. Example of Dispatcher configured with two clusters, each with one port

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 Network Dispatcher 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 5 on page 9.

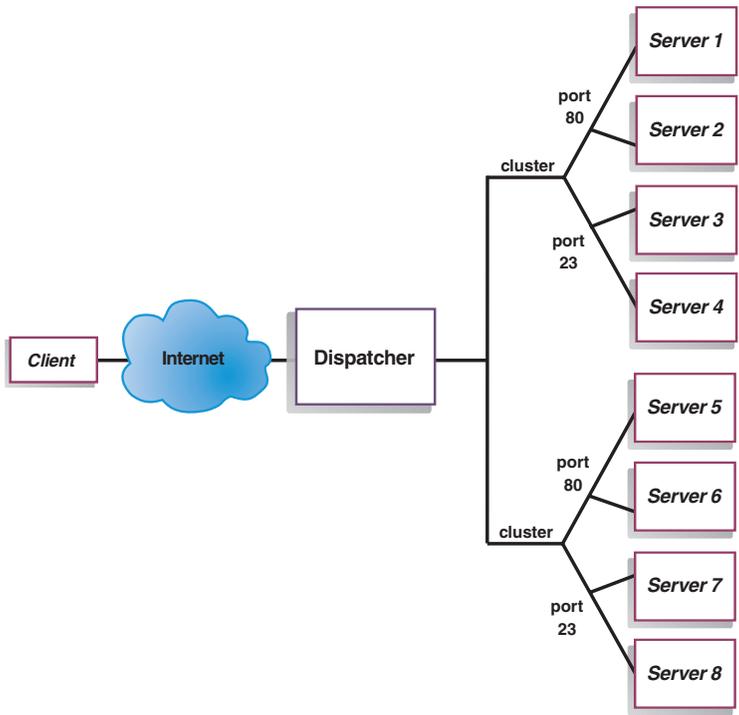


Figure 5. Example of Dispatcher configured with 2 clusters, each with 2 ports

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 2. Installing Network Dispatcher

This chapter instructs you on the hardware requirements and installation of Network Dispatcher on AIX, Linux, Solaris, and Windows 2000. Follow these instructions beginning at:

- “Requirements for AIX” on page 12
- “Requirements for Red Hat Linux or SuSE Linux” on page 16
- “Requirements for Solaris” on page 19
- “Requirements for Windows 2000” on page 21

Notes:

1. If you are migrating from a previous version, please note that the Network Dispatcher install directory structure has changed. You will need to move any of your own configuration files into the `...nd/servers/configurations/component` directory (where *component* is either dispatcher, cbr, ml, ss, or lbc). Also, you will need to move any of your own scripts (such as goIdle and goStandby) into the `...nd/servers/bin` directory in order to run them.
2. If you log off a machine after Network Dispatcher has been installed, you must restart all Network Dispatcher services when you log back on.
3. The required Java level for Network Dispatcher Release 2.0 is 1.3.0 or higher. Since some applications located on the Network Dispatcher box might require other versions of Java, it is necessary to have the correct versions of Java installed on the box when you upgrade.

To ensure that the Network Dispatcher components use the correct version of Java when multiple versions are installed, do the following:

- a. Install the correct version of Java 1.3 for your operating system, as specified in the Requirements sections of this chapter.
- b. Edit the Network Dispatcher script files to use Java 1.3. By default, the script files are located in the following directories:

Unix-based

`/usr/bin/<scriptfile>`

Windows

`C:\WINNT\System32\<scriptfile.cmd>`

Edit the script files for each component of Network Dispatcher that you are upgrading. The script files for each component are:

Administration

`ndadmin`

Dispatcher

ndserver, ndcontrol, ndwizard, ndkeys

Content Based Routing (CBR)

cbrserver, cbrcontrol, cbrwizard, cbrkeys

Site Selector

ssserver, sscontrol

Cisco Consultant

lbserver, lbcontrol

Note: By default these files are read-only; you must, therefore, change the permissions for these files before you can save the changes.

- c. Wherever a java or javaw command is found in the script files, add a path as a prefix indicating where the command is located in the Java 1.3 installation directory.

For example, on Windows 2000, if Java 1.3 is installed in C:\Program Files\IBM\Java13\jre\bin, change the line in ndserver.cmd:

```
from: javaw %END_ACCESS%  
      -DEND_INSTALL_PATH=%IBMNDPATH% ..  
  
to:   C:\Program Files\IBM\Java13\jre\bin\javaw  
      %END_ACCESS% -DEND_INSTALL_PATH=%IBMNDPATH%  
      ...
```

Requirements for AIX

- Any IBM RS/6000 based machine
- IBM AIX 5.1 with APAR IY19177. Support will be for 32-bit Power PC (*not* 64-bit kernel).
IBM AIX 4.3.3.10 plus apars (in order to support Java 1.3). Refer to the README for the IBM AIX Developer Kit for a list of required AIX apars.
- 50 MB of available disk space for installation

Note: Additional disk space will be needed for logs.

- The following Network Interface Cards (NICs) are supported:
 - 16 Mb Token ring
 - 10 Mb Ethernet
 - 100 Mb Ethernet
 - 1 Gb Ethernet
 - Fiber distributed data interface (FDDI)
 - Multi-port Ethernet NICs

Note: The implementation of the multi-port NICs vary from vendor to vendor. Therefore, support for some multi-port NICs may be limited.

- IBM AIX Developer Kit, Java 2 Technology Edition, Version 1.3.0 or higher for the Java Runtime Environment. (For information on running multiple versions of Java, see Note number 3 on page 11.)
- Edge Server Caching Proxy V2.0, if you are using the CBR component for load balancing HTTP or SSL traffic.
- Netscape Navigator 4.07 (or higher) or Netscape Communicator 4.61 (or higher) for viewing online Help
- For Consultant for Cisco CSS Switches, you must have an installed and configured Cisco CSS 11000 Series Switch.

Installing for AIX

Table 1 lists the installp images for Network Dispatcher for AIX.

Table 1. AIX installp images

Dispatcher (component, administration, license, and messages)	intnd.nd.driver intnd.nd.rte intnd.msg.nd.<language>.nd intnd.admin.rte intnd.msg.<language>.admin
Administration (only)	intnd.admin.rte intnd.msg.<language>.admin
Documentation	intnd.doc.rte
License	intnd.nd.license
Metric Server	intnd.ms.rte

where <language> is one of:

- en_US
- de
- es_ES
- fr
- it
- ja_JP
- Ja_JP
- ko_KR
- pt_BR
- zh_CN
- zh_TW
- Zh_TW

If you are downloading an evaluation copy of the product from the Web site, use the installation instructions on (<http://www.ibm.com/software/webservers/edgeserver/download.html>).

Before you install

When you install the product, you are given the option of installing any or all of the following:

- ND Administration
- ND Dispatcher Device Driver (required by ND Dispatcher)
- ND License (required by ND Dispatcher)
- ND Documentation
- ND Metric Server
- License

Installation steps

Note: If you have an earlier version installed, you should uninstall that copy before installing the current version. First, ensure that all the executors and all the servers are stopped. Then, to uninstall the entire product, enter **installp -u intnd**. To uninstall specific filesets, list them specifically instead of specifying the package name.

Follow these steps to install Network Dispatcher for AIX:

1. Log in as root.
2. Insert the product media, or if you are installing from the Web, copy the install images to a directory.
3. Install the installation image. It is recommended that you use SMIT to install Network Dispatcher for AIX because SMIT will ensure that all messages are installed automatically.

Using **SMIT**:

Select Software Installation and Maintenance

Select Install and Update Software

Select Install and update from latest Available Software

Enter The device or directory containing the installp images

Enter On the *SOFTWARE to Install line, the appropriate information to specify options (or select List)

Press **OK**

When the command completes, press **Done**, and then select **Exit Smit** from the Exit menu or press **F12**. If using SMITTY, press **F10** to exit the program.

Using the Command Line:

If installing from a CD, you must enter the following commands to mount the CD:

```
mkdir /cdrom
mount -v cdrfs -p -r /dev/cd0 /cdrom
```

Refer to the following table to determine which command(s) to enter to install the desired Network Dispatcher packages for AIX:

Table 2. AIX install commands

Network Dispatcher (with msgs). Includes: Dispatcher, CBR, Mailbox Locator, Site Selector, and Cisco Consultant	installp -acXgd <i>device</i> intnd.nd.rte intnd.admin.rte intnd.nd.driver intnd.msg.<language>.nd intnd.msg.<language>.admin
Documents	installp -acXgd <i>device</i> intnd.doc.rte intnd.msg.<language>.doc
Administration (only)	installp -acXgd <i>device</i> intnd.admin.rte intnd.msg.<language>.admin
License	installp -acXgd <i>device</i> intnd.nd.license
Metric Server	installp -acXgd <i>device</i> intnd.ms.rte intnd.msg.<language>.admin

where *device* is:

- /cdrom if you are installing from a CD.
- /*dir* (the directory containing the installp images) if you are installing from a file system.

Ensure that the result column in the summary contains SUCCESS for each part of Network Dispatcher that you are installing (APPLYing). Do not continue until all of the parts you wish to install are successfully applied.

Note: To generate a list of filesets in any installp image, including all available message catalogs, enter

```
installp -ld device
```

where *device* is:

- /cdrom if you are installing from a CD.
- /*dir* (the directory containing the installp images) if you are installing from a file system.

To unmount the CD, type:

```
umount /cdrom
```

4. Verify that the product is installed. Enter the following command:

```
lsipp -h | grep intnd
```

If you installed the full product, this command returns the following:

```
intnd.admin.rte
intnd.doc.rte
intnd.ms.rte
intnd.msg.en_US.admin.rte
intnd.msg.en_US.doc
intnd.msg.en_US.nd.rte
intnd.nd.driver
intnd.nd.license
intnd.nd.rte
```

Network Dispatcher install paths include the following:

- Administration - **/usr/lpp/nd/admin**
- Network Dispatcher components- **/usr/lpp/nd/servers**
- Metric Server - **/usr/lpp/nd/ms**
- Documentation (*Administration Guide*) - **/usr/lpp/nd/documentation**

Requirements for Red Hat Linux or SuSE Linux

- Red Hat Linux version 7.1 (Linux kernel version 2.4.2-2) or SuSE Linux version 7.1 (Linux kernel version 2.4.0-4GB). Both uniprocessor and multiprocessor kernels are supported.

Note: If you are using Dispatcher's MAC forwarding method with high availability and collocation, you will need to install a Linux kernel patch. For information on how to download and install the patch see "Installing the Linux kernel patch (to suppress arp responses on the loopback interface)" on page 66.

- 50 MB of available disk space for installation

Note: Additional disk space will be needed for logs.

- The following Network Interface Cards (NICs) are supported:
 - 10 Mb Ethernet
 - 100 Mb Ethernet
 - 1 Gb Ethernet
 - Multi-port Ethernet NICs (Only Mode 1 support. Fault tolerance (Mode 2) and port aggregation (Mode 3) are unsupported.)

Note: The implementation of the multi-port NICs vary from vendor to vendor. Therefore, support for some multi-port NICs may be limited.

- A version of the Korn Shell (ksh) must be installed
- IBM Runtime Environment for Linux, Java 2 Technology Edition, Version 1.3.0 or higher. (For information on running multiple versions of Java, see Note number 3 on page 11.)
- The JAVA_HOME and PATH environment variables must be set using the **export** command. The content of JAVA_HOME variable is dependent on where the user has Java installed. Below is an example:
 - JAVA_HOME=/opt/IBMJava2-13/jre
 - PATH=\$JAVA_HOME/bin:\$PATH
- Edge Server Caching Proxy V2.0, if you are using the CBR component for load balancing HTTP or SSL traffic
- Netscape Navigator 4.07 (or higher) or Netscape Communicator 4.61 (or higher) for viewing online Help
- For Consultant for Cisco CSS Switches, you must have an installed and configured Cisco CSS 11000 Series Switch.

Installing for Linux

This section explains how to install Network Dispatcher on Red Hat Linux or SuSE Linux using the product CD or the downloaded evaluation copy of the product from the Web site. Installation instructions can be found on the Web site (<http://www.ibm.com/software/webservers/edgeserver/download.html>).

Before you install

Before beginning the installation procedure, ensure that you have root authority to install the software.

Installation steps

Note: If you have an earlier version installed, you should uninstall that copy before installing the current version. First, ensure that all the executors and all the servers are stopped. Then to uninstall the entire product, enter **rpm -e *pkgname***. When uninstalling, reverse the order used for package installation ensuring the administration packages are last to be uninstalled.

To install Network Dispatcher:

1. Prepare to install.

- Log in as root.
- Insert the product media or download the product from the Web site and install the installation image using RPM (Red Hat Packaging Manager).

Note: The install package for Red Hat Linux and the install package for SuSE Linux cannot be run on any other product version of Linux.

The installation image is a file in the format **ndlinux-version.tar**.

- Untar the tar file in a temporary directory by typing: **tar -xf ndlinux-version.tar**. The result will be a set of files with the .rpm extension.

The following is a list of the RPM installable packages.

- **ibmnd-adm-release-version.i386.rpm** (ND Administration)
 - **ibmnd-doc-release-version.i386.rpm** (Documentation)
 - **ibmnd-ms-release-version.i386.rpm** (Metric Server)
 - **ibmnd-srv-release-version.i386.rpm** (Network Dispatcher Runtime)
 - **ibmnd-lic-release-version.i386.rpm** (License)
- The order in which the packages are installed is important. Below is a list of packages needed for each component and the order in which they should be installed:
 - Administration (adm)
 - License (lic)
 - Network Dispatcher components (srv)
 - Metric Server (ms)
 - Documentation (doc)

The command to install the packages should be issued from the same directory where the RPM files reside. Issue the following command to install each package: **rpm -i package.rpm**.

Note: At least one of the RPM files requires that Java be installed and registered in the RPM database. If Java is installed but not registered in the RPM database, use the install command with a 'no dependencies' option as follows:

rpm -i --nodeps package.rpm

- Network Dispatcher install paths include the following:
 - Administration - **/opt/nd/admin**
 - Network Dispatcher components - **/opt/nd/servers**
 - Metric Server - **/opt/nd/ms**
 - Documentation (*Administration Guide*) - **/opt/nd/documentation**
 - To uninstall the packages, reverse the order used for package installation ensuring the administration packages are last to be uninstalled.
2. Verify that the product is installed. Enter the following command:
- rpm -qa | grep ibmnd**
- Installing the full product should produce a listing like the following:

- *ibmnd-adm-release-version*
- *ibmnd-doc-release-version*
- *ibmnd-ms-release-version*
- *ibmnd-srv-release-version*
- *ibmnd-lic-release-version*

Requirements for Solaris

- Any SPARC workstation or Ultra 60 server supported by Solaris Version 7 or Solaris Version 8. Network Dispatcher only supports 32-bit mode for Solaris platforms.
- 50 MB of available disk space for installation

Note: Additional disk space will be needed for logs.

- The following Network Interface Cards (NICs) are supported:
 - 10 Mb Ethernet
 - 100 Mb Ethernet
 - 1 Gb Ethernet (only supported on Ultra 60 server)
 - Multi-port Ethernet NICs (Only Mode 1 support. Fault tolerance (Mode 2) and port aggregation (Mode 3) are unsupported.)

Note: The implementation of the multi-port NICs vary from vendor to vendor. Therefore, support for some multi-port NICs may be limited.

- Java 2 JRE, Standard Edition, Version 1.3.0 or higher. (For information on running multiple versions of Java, see Note number 3 on page 11.)
- Edge Server Caching Proxy V2.0, if using the CBR component for load balancing HTTP or SSL traffic
- For Solaris 7, Sun Microsystems HotJava Browser 1.0.1 or higher for viewing online Help
For Solaris 8, Netscape Navigator 4.07 (or higher) or Netscape Communicator 4.61 (or higher) for viewing online Help
- For Consultant for Cisco CSS Switches, you must have an installed and configured Cisco CSS 11000 Series Switch.

Installing for Solaris

This section explains how to install Network Dispatcher on Solaris using the product CD. If you are downloading an evaluation copy of the product from the internet, use the installation instructions on the Web site (<http://www.ibm.com/software/webservers/edgeserver/download.html>).

Before you install

Before beginning the installation procedure, ensure that you have root authority to install the software.

Installation steps

Note: If you have an earlier version installed, you should uninstall that copy before installing the current version. First, ensure that you have stopped both the executor and the server. Then, to uninstall Network Dispatcher enter **pkgrm *pkgname***.

To install Network Dispatcher:

1. Prepare to install.

- Log in as root user.
- Insert the CD-ROM that contains the Network Dispatcher software into the appropriate drive.

At the command prompt, enter **pkgadd -d *pathname***, where **-d *pathname*** is the device name of the CD-ROM drive or the directory on the hard drive where the package is located; for example: **pkgadd -d /cdrom/cdrom0/**.

You will be given a list of packages to install. They are:

- **ibmdsp** IBM ND for Solaris (Network Dispatcher components)
- **ibmndadm** IBM ND Base Administration for Solaris
- **ibmnddoc** IBM ND Documentation for Solaris
- **ibmndms** IBM ND Metric Server for Solaris
- **ibmdsplic** License for Solaris

If you want to install all of the packages, simply type “all” and press return. If you want to install some of the components, enter the name(s) corresponding to the packages to be installed separated by a space or comma and press return. You may be prompted to change permissions on existing directories or files. Simply press return, or answer “yes.” You need to install prerequisite packages (because it installs in alphabetical order not prerequisite order). If you say “all” then just answer “yes” to all prompting and the install will complete successfully.

All of the packages depend on the common package, **ibmndadm**. This common package must be installed along with any of the other packages.

If you want to install the entire Network Dispatcher product, you must install five pieces: **ibmdsp**, **ibmdsplic**, **ibmndadm**, **ibmnddoc**, and **ibmndms**. If you want to install the remote administration, you only have to install one piece: **ibmndadm**.

The Network Dispatcher components reside in `/opt/nd/servers` install directory.

2. The installed Administration resides in the directory `/opt/nd/admin`
3. The installed Metric Server resides in the directory `/opt/nd/ms`
4. The installed documentation (*Administration Guide*) resides in the directory `/opt/nd/documentation`
5. Verify that the product is installed. Issue the following command: `pkginfo | grep ibm`.

Installing the full product should produce a listing like the following:

- `ibmdsp`
- `ibmndadm`
- `ibmnddoc`
- `ibmndms`
- `ibmdsplic`

Requirements for Windows 2000

- Any Intel x86 PC supported by Microsoft Windows 2000
- Windows 2000 Professional, Server, or Advanced Server
- 50 MB of available disk space for installation

Note: Additional disk space will be needed for logs.

- The following Network Interface Cards (NICs) are supported:
 - 16 Mb Token ring
 - 10 Mb Ethernet
 - 100 Mb Ethernet
 - 1 Gb Ethernet
 - Multi-port Ethernet NICs

Note: The implementation of the multi-port NICs vary from vendor to vendor. Therefore, support for some multi-port NICs may be limited.

- IBM Cross Platform Technologies for Windows v2.0 (SDK 1.3.0 or higher)
You must download both the Developer Kit installable package and the Runtime Environment installable package prior to running the InstallShield program. (For information on running multiple versions of Java, see Note number 3 on page 11.)
- Edge Server Caching Proxy V2.0, if using the CBR component for load balancing HTTP or SSL traffic

- Ensure your default browser is either Netscape Navigator 4.07 (or higher), Netscape Communicator 4.61 (or higher), or Internet Explorer 4.0 (or higher). The default browser is used for viewing online Help.
- For Consultant for Cisco CSS Switches, you must have an installed and configured Cisco CSS 11000 Series Switch.

Installing for Windows 2000

This section explains how to install Network Dispatcher on Windows 2000 using the product CD. If you are downloading an evaluation copy of the product from the Web site, use the installation instructions on the Web site (<http://www.ibm.com/software/webservers/edgeserver/download.html>).

Installation Packages

You will be given a choice of packages to install.

They are:

- Runtime
- Administration
- License
- Documentation
- Metric Server

Before you install

Windows 2000 version of Network Dispatcher is supported on the following:

- Windows 2000 Professional
- Windows 2000 Server
- Windows 2000 Advanced Server

Note: The Windows 2000 version of Network Dispatcher will *not* run on any other version of Windows.

Restrictions: The Windows 2000 version of Network Dispatcher cannot be installed on the same machine with IBM Firewall.

Before beginning the installation procedure, ensure you have logged in as the Administrator or as a user with administrative privileges.

Installation steps

If you have an earlier version installed, you should uninstall that copy before installing the current version. To uninstall using the **Add/Remove Program**, do the following:

1. Click **Start**→**Settings**→**Control Panel**
2. Double-click **Add/Remove Programs**
3. Select *Network Dispatcher*

4. Click **Change/Remove** button

To install Network Dispatcher:

1. Insert the Network Dispatcher CD-ROM into your CD-ROM drive, and the install window should come up automatically.
2. The following step is only required if autorun of the CD did not work on your computer. Using your mouse, click mouse button 1 to perform these tasks:
 - Click **Start**.
 - Select **Run**.
 - Specify the CD-ROM disk drive, followed by setup.exe, for example:
`E:\setup`
3. Select **Language** in which to read the install process.
4. Click **OK**.
5. Follow the instructions of the setup program.
6. If you want to change the drive or directory destination, click **Browse**.
7. You have the choice of selecting "All of the ND product" or "your choice of components."
8. After installation is complete, a message will tell you to reboot your system before using Network Dispatcher. This is necessary to make sure that all files are installed and the IBMNDPATH environment variable is added to the registry.

Network Dispatcher install paths include the following:

- Administration – `c:\Program~1\IBM\edge\nd\admin`
- Network Dispatcher components – `c:\Program~1\IBM\edge\nd\servers`
- Metric Server – `c:\Program~1\IBM\edge\nd\ms`
- Documentation (Administration Guide) – `c:\Program~1\IBM\edge\nd\documentation`

Chapter 3. Introducing Network Dispatcher

This chapter gives an overview of Network Dispatcher and includes the following sections:

- “What is Network Dispatcher?”
- “Why do I need Network Dispatcher?” on page 26
- “What are the new features?” on page 27
- “What are the components of Network Dispatcher?” on page 32
- “How about high availability?” on page 42

What is Network Dispatcher?

Network Dispatcher is a software solution for load-balancing 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. Network Dispatcher 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, Network Dispatcher 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't get through at times of greatest demand, use Network Dispatcher to automatically find the optimal server to handle incoming requests, thus enhancing your customers' satisfaction and your profitability.

Network Dispatcher consists of five components that can be used 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, 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.

- 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.

- For IMAP or POP3 protocols, you can use the **Mailbox Locator** component that functions as a proxy and chooses an appropriate server based on the userID and password provided by the client.
- 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 **Consultant for Cisco CSS Switches** component to generate server weighting metrics that are then sent to the Cisco CSS Switch for optimal server selection, load optimization, and fault tolerance.

For more information on the Dispatcher, CBR, Mailbox Locator, Site Selector, and Consultant for Cisco CSS Switches components, see “What are the components of Network Dispatcher?” on page 32.

Why do I need Network Dispatcher?

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

Currently, network administrators are using numerous methods to try to maximize access. Some of these methods allow users to 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 blindly 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 Network Dispatcher. 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

Network Dispatcher uses standard TCP/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 simple MAC level forwarding method, Dispatcher needs only to look 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 compared with other approaches and can result in improved network performance.

High availability

Dispatcher offers built-in high availability, utilizing a backup machine that remains ready at all times to take over load balancing should the primary Dispatcher machine fail. Dispatcher also offers mutual high availability which allows two machines to be both active and standby for each other. See “How about high availability?” on page 42.

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.

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.

What are the new features?

Network Dispatcher for IBM WebSphere Edge Server Version 2.0 contains a number of new features. The most significant are listed here.

- **AIX v5.1 Support**

This feature applies to all the Network Dispatcher components.

Network Dispatcher now supports a newer version of AIX: AIX v5.1. See “Requirements for AIX” on page 12 for more information.

- **SuSE Linux v7.1 Support**

This feature applies to all the Network Dispatcher components.

Network Dispatcher now supports SuSE Linux v7.1 (kernel version 2.4.0–4GB). Previously, Network Dispatcher only supported Red Hat Linux. See “Requirements for Red Hat Linux or SuSE Linux” on page 16 for more information.

- **Red Hat Linux v7.1 Support**

This feature applies to all the Network Dispatcher components.

Network Dispatcher now supports a newer version of Red Hat Linux: Red Hat Linux v7.1 (kernel version 2.4.2–2). See “Requirements for Red Hat Linux or SuSE Linux” on page 16 for more information.

- **Linux and Solaris NLS (National Language Support)**

This feature applies to all the Network Dispatcher components.

On Linux and Solaris operating systems, Network Dispatcher offers NLS for the Group 1 countries.

- **New Chinese NLS Standard Support**

This feature applies to all the Network Dispatcher components.

Network Dispatcher provides NLS for the new Chinese Standard GB 18030.

- **Consultant for Cisco CSS Switches component (Cisco Consultant)**

This feature is a new component for Network Dispatcher.

Collaboration with Cisco and their Content Distribution Network (CDN) has led to the development of an additional component for Network Dispatcher — Cisco Consultant. This component (which was first introduced as a standalone Preview) allows the Network Dispatcher to generate weights and make load balancing decisions for the Cisco CSS Switch.

See “Chapter 12. Planning for the Consultant for Cisco CSS Switches component” on page 109 and “Chapter 13. Configuring the Consultant for Cisco CSS Switches component” on page 113 for more information.

- **Site Selector component**

This feature is a new component for Network Dispatcher.

The Site Selector component balances the load among a group of servers by selecting the “right” server’s IP address for a name service request. This allows the client to connect directly to the server for all its communication. Site Selector replaces Interactive Session Support (ISS), a Network Dispatcher component in prior releases. Site Selector provides similar functionality as ISS but requires fewer steps when setting up a typical DNS load balancing configuration.

See “Chapter 10. Planning for the Site Selector component” on page 97 and “Chapter 11. Configuring the Site Selector component” on page 103 for more information.

- **Metric Server**

This feature applies to all the Network Dispatcher components.

Metric Server provides server load information to the Network Dispatcher in the form of system-specific metrics. Metric Server agent is a component of Network Dispatcher that can be installed and run on servers that Network Dispatcher is load balancing. Metric Server replaces the System Monitoring Agent (SMA), which was supported on Linux, in previous releases. Metric Server is supported on all platforms. It is recommended that Metric Server be used in conjunction with the Site Selector component.

See “Metric Server” on page 136 for more information.

- **Mailbox Locator component**

This feature is a new component for Network Dispatcher.

The Mailbox Locator component was formerly a feature within the CBR component that load balanced across IMAP and POP3 mail servers based on userID and password. Separating CBR into two components permits Mailbox Locator (formerly known as “CBR for IMAP/POP3”) and CBR with Caching Proxy to be run on the same machine.

See “Chapter 8. Planning for the Mailbox Locator component” on page 87 and “Chapter 9. Configuring the Mailbox Locator component” on page 91 for more information.

- **Usability improvements to the Content Based Routing (CBR) component**

Configuring Caching Proxy configuration file (ibmproxy.conf) to use CBR has been streamlined and CBR has been enhanced so that multiple instances of the Caching Proxy can run simultaneously on the same machine while interfacing with CBR. For more information on how to configure CBR with Caching Proxy, see “Setting up the CBR machine” on page 79.

- **Network Address Translation (NAT) and Network Address Port Translation (NAPT) support**

This feature applies to the Dispatcher component.

NAT/NAPT removes the limitation for backend servers to be located on a locally attached network. It also enables Dispatcher to load balance the client’s TCP requests to multiple server daemons running on the same physical machine. You can configure servers 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 to listen on different port numbers.

The advantage of Dispatcher's nat forwarding method is that it is configured at the port level giving you much better granularity.

Note: For Network Dispatcher, NAT/NAPT will not work with any application protocols, such as FTP, that imbed the addresses or port numbers in the data portion of the messages. This is a well-known limitation of header-based NAT/NAPT.

See "Dispatcher's NAT/NAPT (nat forwarding method)" on page 47 for more information.

- **Dispatcher's content-based routing feature (using content rule and SSL session ID affinity)**

This feature applies to the Dispatcher component.

In prior Network Dispatcher releases, content-based routing was only available using the CBR component in conjunction with Caching Proxy. The Dispatcher component now allows you to perform content-based routing for HTTP (using the "content" type rule) and HTTPS (using SSL session ID affinity) without Caching Proxy. For HTTP and HTTPS traffic, the Dispatcher component can provide faster content-based routing than the CBR component.

See "Dispatcher's content-based routing (cbr forwarding method)" on page 49 for more information on using the content rule and SSL session ID affinity.

- **Passive Cookie Affinity**

This feature applies to the Dispatcher component's content-based routing feature (cbr forwarding method) and the CBR component.

Passive cookie affinity allows you to load-balance Web traffic with affinity to the same server based upon self-identifying cookies generated by the servers. See "Passive cookie affinity" on page 177 for more information.

- **URI Affinity (load balancing to caching proxies)**

This feature applies to the Dispatcher component's content-based routing feature (cbr forwarding method) and the CBR component.

URI affinity allows you to load-balance Web traffic to caching-proxy servers in a manner that effectively increases the size of the cache. See "URI affinity" on page 177 for more information.

- **Cluster (or Site) Specific Proportions**

This feature applies to all Network Dispatcher components.

In earlier releases, the proportion of importance (given to active connections, new connections, port, and system metrics) for determining load-balancing decisions was set from the manager function. These proportions were applied to every cluster in the configuration for the component. All clusters were measured using the same proportions regardless of the site it was load balancing.

With this enhancement, you can set the proportion of importance on a per cluster (or site) basis. See “Proportion of importance given to status information” on page 122 for more information.

- **Server Partitioning**

This feature applies to all the Network Dispatcher components.

Network Dispatcher now provides the ability to partition one physical server into several logical servers. This allows you to query, for example, a particular service on the machine to detect if a servlet engine or a database request is running faster, or not running at all. This enhancement provides you with the capability to distribute load based on more granular service-specific workload. See “Server Partitioning: logical servers configured to one physical server (IP address)” on page 138 for more information.

- **HTTP Advisor Request/Response (URL) Option**

This feature applies to the Dispatcher and CBR components.

With this enhancement for the HTTP advisor, you can assess the health of individual services within a server. For each 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. See “HTTP advisor request/response (URL) option” on page 140 for more information.

- **Cluster (or Site) Specific Advisors**

This feature applies to all the Network Dispatcher components.

Network Dispatcher allows you to start different advisors running on the same port but configured on different clusters (sites). For example, this feature will allow you to use an HTTP advisor on port 80 for one cluster (site) and a custom advisor on port 80 for another cluster (site). See “Starting and stopping an advisor” on page 127 for more information.

- **Denial of Service (DoS) Attack Detection**

This feature applies to the Dispatcher component.

With this enhancement, Dispatcher provides the ability to detect potential denial of service attacks and notify administrators via an alert. Dispatcher does this by analyzing incoming requests for a conspicuous amount of half-open connections, a common trait of simple denial of service attacks. See “Denial of service attack detection” on page 178 for more information.

- **Enhanced User Exits**

This feature applies to all components except Consultant for Cisco CSS Switches and Site Selector.

Network Dispatcher provides additional user exits that trigger scripts which you can customize. You can create the scripts to perform automated actions, such as logging when a high availability state has changed or alerting an Administrator when servers are marked down. Network Dispatcher provides the following new sample script files:

- serverDown, serverUp, managerAlert, and managerClear — (see “Using scripts to generate an alert or record server failure” on page 126 for more information)
- highavailChange — (see “Using scripts” on page 155 for more information)
- halfOpenAlert — a probable Denial of Service (DoS) attack has been detected (see “Denial of service attack detection” on page 178 for more information)
- halfOpenAlertDone — the DoS attack has finished (see “Denial of service attack detection” on page 178 for more information)
- **DB2 Advisor**
This feature applies to the Dispatcher component.
Dispatcher provides a DB2 advisor which communicates with the DB2 servers. See “List of advisors” on page 129 for more information on the DB2 advisor.

What are the components of Network Dispatcher?

The five components of Network Dispatcher are: Dispatcher, Content Based Routing (CBR), Mailbox Locator, Site Selector, and Consultant for Cisco CSS Switches. Network Dispatcher 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 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, 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 NAT (network address translation)/ NAPT (network address port translation) 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 GRE/WAND encapsulation 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 appears to be a single, virtual server. Your site thus appears 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

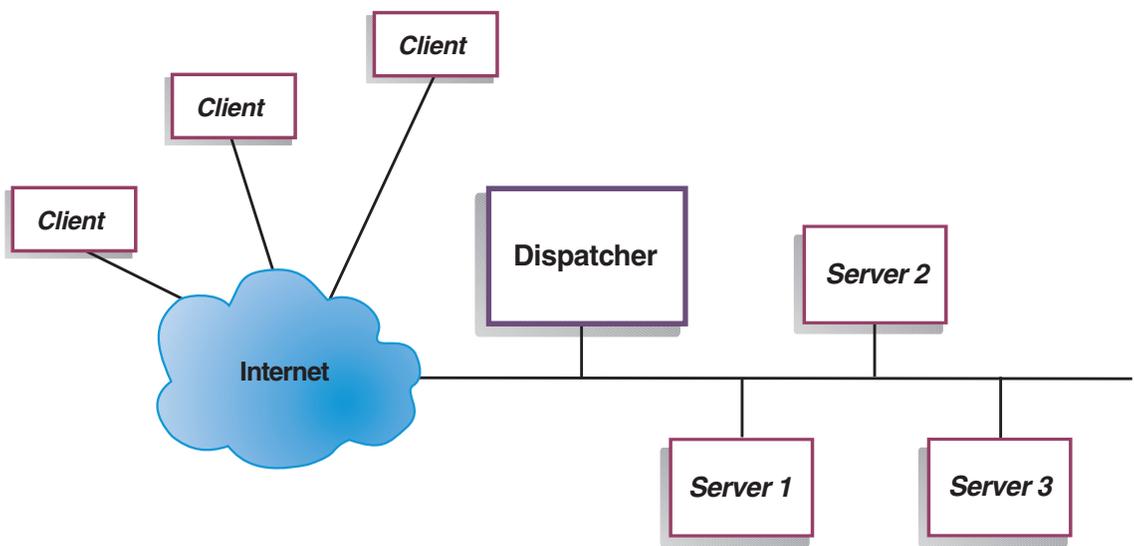


Figure 6. Example of a physical representation of a site using Dispatcher to manage local servers

Figure 6 on page 33 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 Metric Server

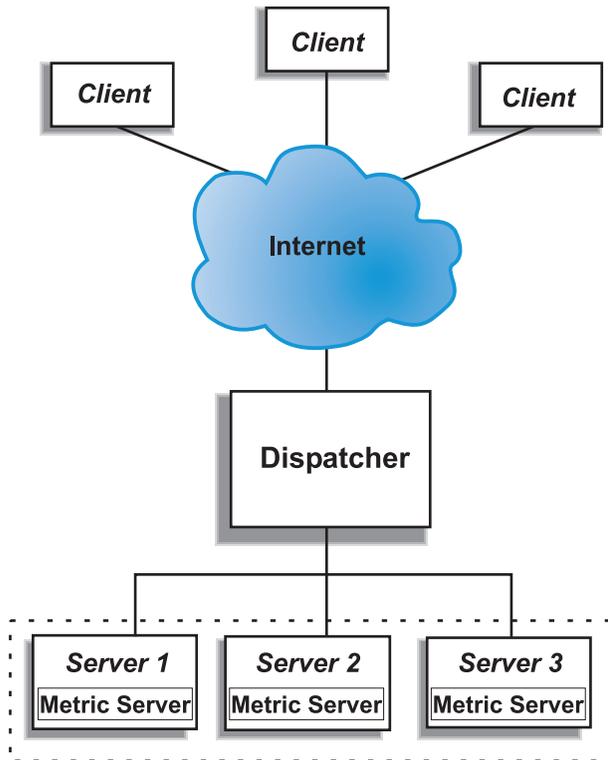


Figure 7. Example of a site using Dispatcher and Metric Server to manage servers

Figure 7 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 Network Dispatcher components.

Managing local and remote servers with Dispatcher

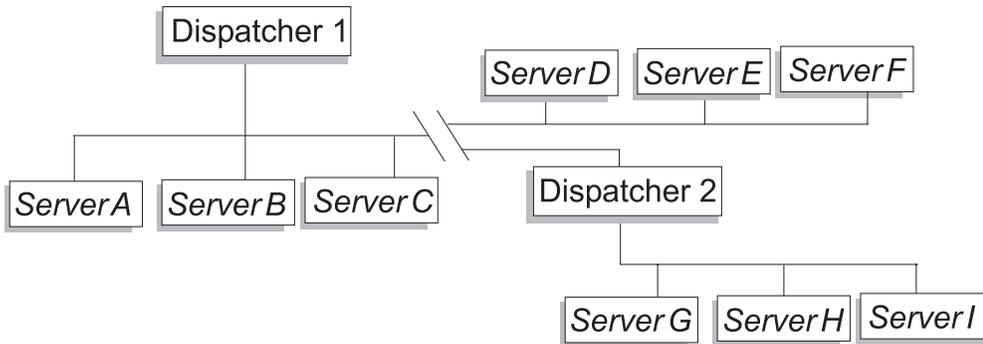


Figure 8. Example of a site using Dispatcher to manage local and remote servers

Wide area support in Dispatcher enables you to use both local and remote servers (servers on different subnets). Figure 8 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 47 and “GRE (Generic Routing Encapsulation) support” on page 148 for more information.

Overview of 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 along with Caching Proxy examines HTTP requests using specified rule types.

CBR gives you the ability to specify a set of servers that should 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 will also detect when one server in a set has failed, and stop 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, 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

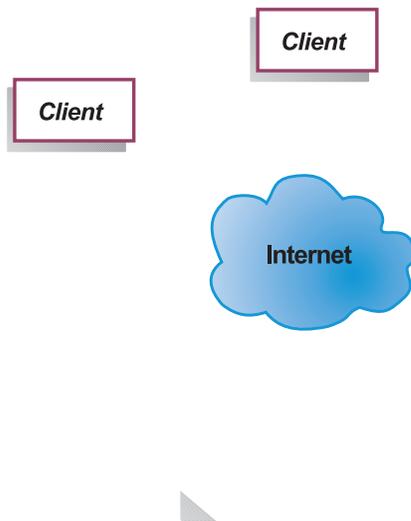


Figure 9 shows a logical representation of a site in which CBR is being used to proxy some content from local servers. The CBR component uses Caching Proxy to forward client requests (HTTP or HTTPS) to the servers based on the content of the URL.

Overview of Mailbox Locator component

Mailbox Locator can provide a single point of presence for many IMAP or POP3 servers. Each server can have a subset of all user mailboxes serviced by the point of presence. For IMAP and POP3 traffic, Mailbox Locator is a proxy that chooses an appropriate server based on userID and password provided by the client. Mailbox Locator does not support rules-based load balancing.

Note: The Mailbox Locator component was formerly a feature within the CBR component that load balanced across IMAP and POP3 mail servers. Separating CBR into two components *removes* the limitation that "CBR for IMAP/POP3" (Mailbox Locator) and "CBR for HTTP/HTTPS" (CBR with Caching Proxy) cannot be run on the same machine.

Managing local servers with Mailbox Locator

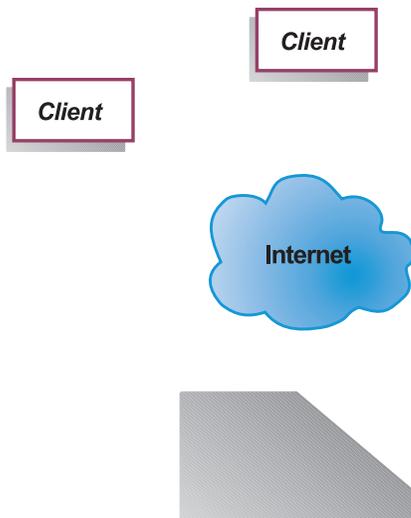


Figure 10 shows a logical representation of a site in which Mailbox Locator is being used to proxy client requests (IMAP or POP3 protocol) to the appropriate server, based on userID and password.

Overview of 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. Name server returns the IP address to the client.

Metric Server is a system monitoring component of Network Dispatcher 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, taking into account 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

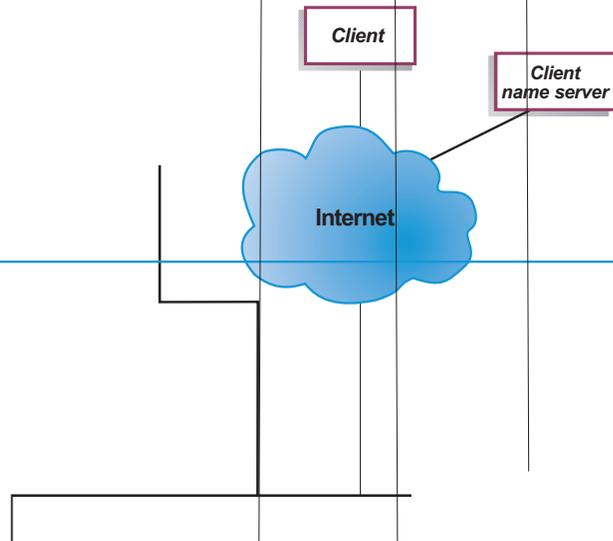


Figure 11 illustrates a site in which the Site Selector component is used to answer requests. Server1, Server2, and Server3 are local. Server4, Server5, and Server6 are remote.

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.

Once the client receives the IP address of the server, the client routes subsequent 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. You should use Metric Server in conjunction with Site Selector, otherwise Site Selector can only use a round-robin selection method for load balancing.

Overview of Consultant for Cisco CSS Switches component

Consultant for Cisco CSS Switches 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 Network Dispatcher's sophisticated awareness algorithms for determining backend server, application, and database availability and loading information. The Cisco Consultant function utilizes Network Dispatcher's manager, standard and custom advisors, and Metric Server to determine the metrics, health, and loading of the backend servers, applications, and databases. With this information Cisco Consultant generates server weighting metrics, which it sends to the Cisco CSS Switch for optimal server selection, load optimization, and fault tolerance.

The Cisco CSS Switch makes load-balancing decisions based on user-specified criteria.

Cisco Consultant tracks many criteria, 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

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

Cisco Consultant:

- Uses a published SNMP interface to obtain connection information from the Cisco CSS Switch
- Uses advisor input to analyze the connection information
- Uses Metric Server information to analyze relative server health
- Generates weights for each server in the configuration

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.

These weights are provided to the Cisco CSS Switch using SNMP. As the weight of any server is set higher, the Cisco CSS Switch directs more requests to that server.

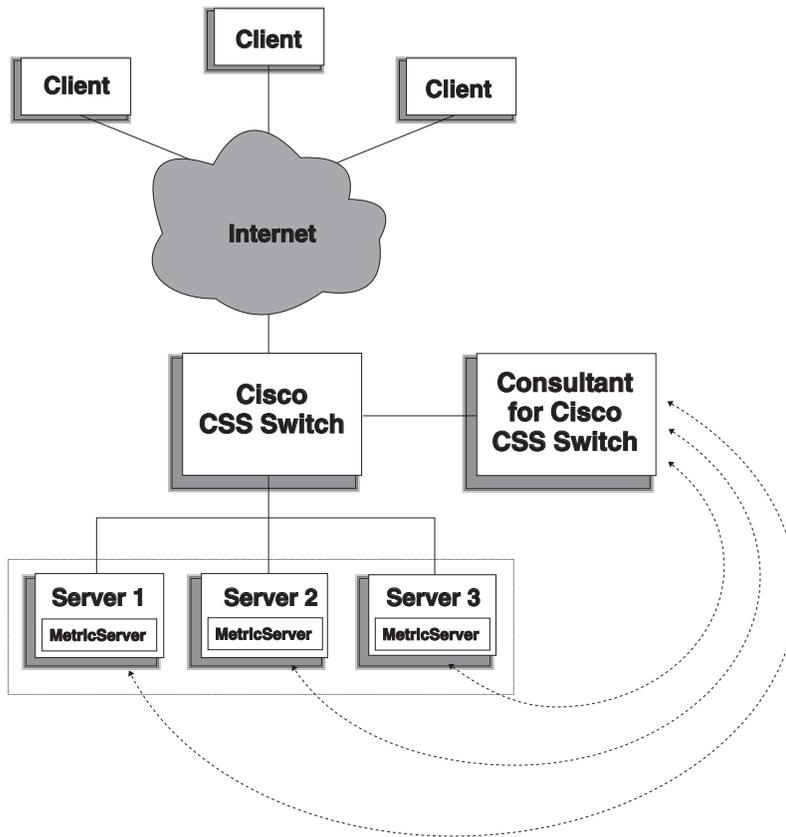


Figure 12. Example of a site using Cisco Consultant and Metric Server to manage local servers

Cisco Consultant, 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 server load optimization. Cisco Consultant is part of an overall complementary solution between the Cisco CSS Switch and IBM’s WebSphere Edge Server.

Refer to “Chapter 2. Installing Network Dispatcher” on page 11 for a list of Cisco Consultant requirements.

How about high availability?

Dispatcher

The Dispatcher component offers a built-in high availability feature. 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 151.

CBR, Mailbox Locator, Site Selector

When using a two-tier configuration with a Dispatcher machine load balancing traffic to two or more server machines that have either CBR, Mailbox Locator or Site Selector, you can achieve a level of high availability for these components of Network Dispatcher.

Chapter 4. Planning for the Dispatcher component

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

- See “Chapter 5. Configuring the Dispatcher component” on page 53 for information on configuring the load-balancing parameters of Dispatcher.
- See “Chapter 14. Advanced Network Dispatcher Functions” on page 119 for information on how to set up Network Dispatcher for more advanced functions.
- See “Chapter 15. Operating and managing Network Dispatcher” on page 185 for information on remote authenticated administration, Network Dispatcher logs, and usage of the Network Dispatcher components.

This chapter includes the following sections:

- “Hardware and software requirements”
- “Planning considerations”
- “High availability” on page 45
- “Dispatcher’s MAC-level routing (mac forwarding method)” on page 47
- “Dispatcher’s NAT/NAPT (nat forwarding method)” on page 47
- “Dispatcher’s content-based routing (cbr forwarding method)” on page 49

Hardware and software requirements

Platform requirements:

- For AIX, see “Requirements for AIX” on page 12
- For Linux, see “Requirements for Red Hat Linux or SuSE Linux” on page 16
- For Solaris, see “Requirements for Solaris” on page 19
- For Windows 2000, see “Requirements for Windows 2000” on page 21

Planning considerations

Dispatcher consists of the following functions:

- **ndserver** 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 will be 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. Currently there are advisors available for the following protocols: HTTP, FTP, SSL, SMTP, NNTP, IMAP, POP3, 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 129.

You also have the option of writing your own advisors (see “Create custom (customizable) advisors” on page 131).

Using the advisors is optional but recommended.

- To configure and manage the executor, advisors, and manager, use the command line (**ndcontrol**) or the graphical user interface (**ndadmin**).
- A **sample configuration file** is provided to use for configuration and administration of the Dispatcher machine. See “Appendix F. Sample configuration files” on page 345. After you have installed the product, this file can be found in the **nd/servers/samples** subdirectory where Network Dispatcher is located.
- 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.

High availability

Simple high availability

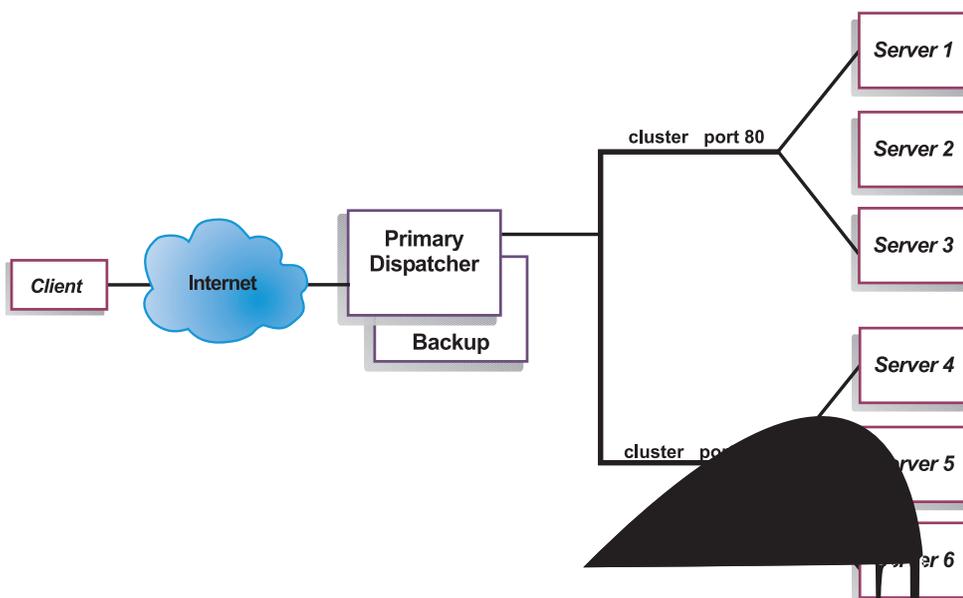


Figure 13. Example of a Dispatcher using simple high availability

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 would in a single Dispatcher configuration. The second Dispatcher machine monitors the “health” of the first, and will take 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.

For information about configuring high availability, see “High availability” on page 150.

Mutual high availability

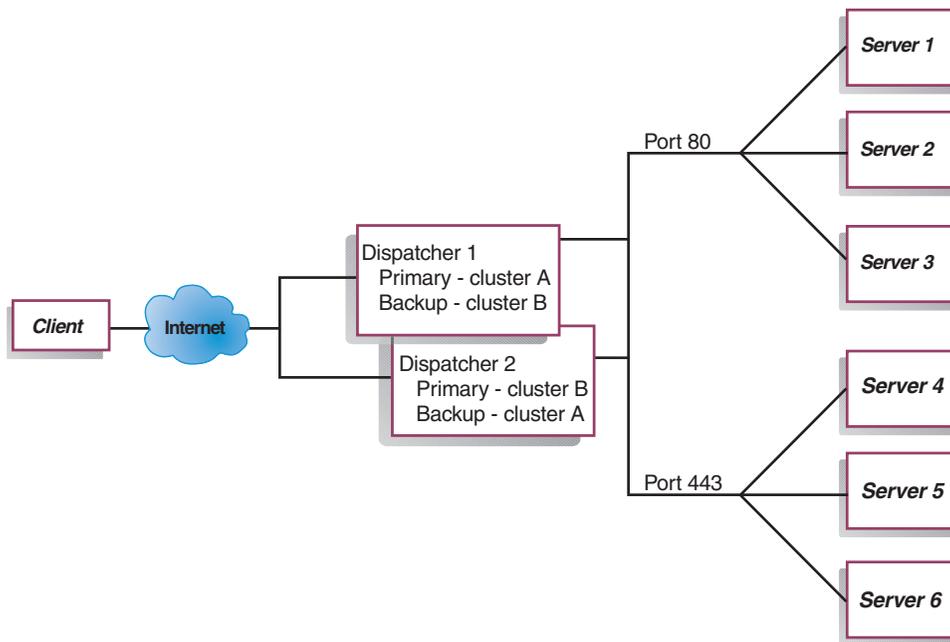


Figure 14. Example of a Dispatcher using mutual high availability

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 on page 46. 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.

For information about configuring high availability and mutual high availability, see "High availability" on page 150.

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 **ndcontrol 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. Once you add the port, you cannot change the setting of the forwarding method. See "ndcontrol port — configure ports" on page 261 for more information.

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, etc.

Limitations:

- Dispatcher's implementation of NAT/NAPT is a *simple* implementation of this feature. It only analyzes and operates upon 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.

To implement NAT/NAPT:

- Set the **clientgateway** parameter on the **ndcontrol 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 Network Dispatcher to clients. This value must be set to a nonzero IP address before you can use NAT/NAPT. See "ndcontrol executor — control the executor" on page 239 for more information.
- Add a port using the **ndcontrol 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. Once you add the port, you cannot change the setting of the forwarding method. See "ndcontrol port — configure ports" on page 261 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 **ndcontrol** command. For example:

```
ndcontrol server add cluster:port:server mapport value returnaddress  
rtrnaddress router rtraddress
```

– **mapport**

This will 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. Mapport allows Network Dispatcher 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 may 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 hostname 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 will return 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.

– **router**

The address of the router to the remote server.

For more information on the `ndcontrol server` command using the `mapport`, `returnaddress`, and `router` parameters, see “`ndcontrol server` — configure servers” on page 274.

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

In prior Network Dispatcher releases, content-based routing was only available using the CBR component in conjunction with Caching Proxy. The Dispatcher component now allows you to perform content-based routing for HTTP (using the “content” type rule) and HTTPS (using SSL session ID affinity) without Caching Proxy. For HTTP and HTTPS traffic, the Dispatcher component can provide faster content-based routing than the CBR component.

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 will save CPU cycles and the client will get a quicker response. In order to enable SSL session ID affinity, 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.

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

- Set the **clientgateway** parameter on the **ndcontrol 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 "ndcontrol executor — control the executor" on page 239 for more information.
- Add a port using the **method** parameter on the **ndcontrol port add** command. The forwarding method value should be set to **cbr**. See "ndcontrol port — configure ports" on page 261 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
ndcontrol server add cluster:port:server mapport value returnaddress rtrnaddress router rtraddress

Note: For information on configuring the server using **mapport**, **returnaddress** and **router** parameters, see page 48.

- **For HTTP:** Configure using rules based on the client request content (rule type **content**). For example,
ndcontrol 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 165. For more information on valid expressions for *pattern*, see "Appendix C. Content rule (pattern) syntax" on page 285.

For HTTPS (SSL): To configure the SSL ID session affinity, set the **stickytime** parameter on the port to a nonzero value. For more information about **stickytime** on the port command, see “ndcontrol rule — configure rules” on page 267.

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.

Chapter 5. Configuring the Dispatcher component

Before following the steps in this chapter, see “Chapter 4. Planning for the Dispatcher component” on page 43. This chapter explains how to create a basic configuration for the Dispatcher component of Network Dispatcher.

- See “Chapter 14. Advanced Network Dispatcher Functions” on page 119 for more complex configurations of Network Dispatcher.
- See “Chapter 15. Operating and managing Network Dispatcher” on page 185 for information on remote authenticated administration, Network Dispatcher logs, and usage of the Network Dispatcher components.

Overview of configuration tasks

Note: 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 3. 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 56
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 62

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 Dispatcher from the command line:

- Issue the **ndserver** command from the command prompt. For Windows 2000, ndserver runs as a service that starts automatically.

Note: To stop the service, issue the following: **ndserver stop**.

- Next, issue the 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 **ndcontrol**. For more information about commands, see “Appendix B. Command reference for Dispatcher, CBR, and Mailbox Locator” on page 225.

You can enter a minimized version of the ndcontrol 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 **ndcontrol he f** instead of **ndcontrol help file**.

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

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

Scripts

The commands for configuring the Dispatcher can be entered into a configuration script file and executed together. See “Sample Network Dispatcher configuration files” on page 345.

Note: To quickly execute the content of a script file (e.g. myscript), use either of the following commands:

- For updating the current configuration, run the executable commands from your script file using —
ndcontrol file appendload myscript
- For completely replacing the current configuration, run the executable commands from your script file using —
ndcontrol file newload myscript

GUI

For an example of the graphical user interface (GUI), see Figure 2 on page 5.

To start the GUI, follow these steps

1. Ensure ndserver is running
 - For AIX, Linux, or Solaris, run the following as root:
ndserver
 - For Windows 2000, ndserver runs as a service that starts automatically
2. Next, do one of the following:

- For AIX, Linux, or Solaris: enter **ndadmin**
- For Windows 2000: click **Start**, click **Programs**, **IBM WebSphere**, click **Edge Server**, click **IBM Network Dispatcher**, and click **Network Dispatcher**

In order to configure the Dispatcher component from the GUI, you must first select **Dispatcher** in the tree structure. You can start the executor and manager once 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 **ndcontrol** command. For example, to define a cluster using the command line, you would enter **ndcontrol 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 Network Dispatcher components.

The configuration commands can also be run remotely. For more information, see “Remote Authenticated Administration” on page 185.

You can access **Help** by clicking the question mark icon in the upper right hand corner of the Network Dispatcher window.

- **Field Help** — describes each field, default values
- **How do I** — lists tasks that can be done from that screen
- **Contents** — a table of contents of all the Help information
- **Index** — an alphabetical index of the help topics

For more information about using the GUI, see “General Instructions for using the GUI” on page 6.

Configuration Wizard

For more information about using the configuration wizard, see “Configuring using the configuration wizard” on page 4.

Setting up the Dispatcher machine

Before setting up the Dispatcher machine, you must be the root user (for AIX, Linux, or Solaris) or the Administrator on Windows 2000.

For AIX, Linux, and Solaris only, the Network Dispatcher can have a **collocated** server. This simply means that the Network Dispatcher can physically reside on a server machine which it is load balancing.

You will need at least two valid IP addresses for the Dispatcher machine:

- 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 via 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.

Solaris Only:

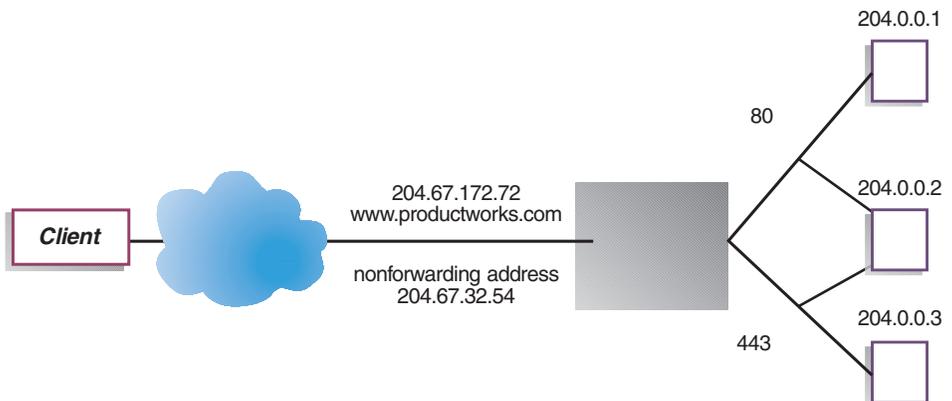
1. By default, Dispatcher is configured to load balance traffic on 100Mbps Ethernet network interface cards. To change the default setting, you must edit the `/opt/nd/servers/ibmnd.conf` file as follows:
 - The default 100Mbps Ethernet adapter is specified in `ibmnd.conf` as `hme`.
 - To use a 10 Mbps Ethernet adapter, replace `hme` with `1e`.
 - To use a 1Gbps Ethernet adapter, replace `hme` with `ge`.
 - To use a multi-port adapter, replace `hme` with `qfe`.
 - To support multiple types of adapters, replicate the line in the `ibmnd.conf` file and modify each line to match your device type.
For example, if you plan to use two 100Mbps Ethernet adapters, the `ibmnd.conf` file should have a single line specifying the `hme` device. If you plan to use one 10Mbps Ethernet adapter and one 100Mbps Ethernet adapter, you will have two lines in the `ibmnd.conf` file: one line specifying the `1e` device and one line specifying the `hme` device.
2. Starting or stopping the Dispatcher executor will unconfigure all aliases on the adapters listed in the `ibmnd.conf` file. To automatically reconfigure

aliases on those adapters (except those for use by the Dispatcher component of Network Dispatcher) use the **goAliases** script file. A sample script is located in the **...nd/servers/samples** directory and *must* be moved to the **...nd/servers/bin** before it will run. The **goAliases** script is automatically executed when the Dispatcher executor starts or stops.

For example, if clusters X and Y are configured for use by the Mailbox Locator component on any of the adapters listed in **ibmnd.conf**, clusters X and Y are unconfigured when the **ndcontrol executor start** or **ndcontrol 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.

Windows 2000 Only: Ensure that IP forwarding is not enabled for the TCP/IP protocol. (See your Windows 2000 TCP/IP configuration.)

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



For help with commands used in this procedure, see "Appendix B. Command reference for Dispatcher, CBR, and Mailbox Locator" on page 225.

For a sample configuration file, see "Sample Network Dispatcher configuration files" on page 345.

Step 1. Start the server function

AIX, Linux, and Solaris: To start the server function, type **ndserver**.

Windows 2000: The server function starts automatically as a service.

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

Step 2. Start the executor function

To start the executor function, enter the **ndcontrol executor start** command. You may also change various executor settings at this time. See “Appendix B. Command reference for Dispatcher, CBR, and Mailbox Locator” on page 225.

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 **ndcontrol executor set nfa *IP_address*** command or edit the sample configuration file. *IP_address* is either the symbolic name or the dotted-decimal 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 **ndcontrol cluster add**. To set cluster options, issue the command **ndcontrol 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 168, “Use wildcard cluster to load balance firewalls” on page 169, and “Use wildcard cluster with Caching Proxy for transparent proxy” on page 170 for more information.

Step 5. Alias the network interface card

Once 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 **ndcontrol cluster 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:

```
ndcontrol cluster configure 204.67.172.72
```

Circumstances where you would not want to configure the cluster address are for 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 cluster configure command if, in stand-alone mode, you use the sample **goldle** script. For information on the goldle script, see “Using scripts” on page 155.

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

Some examples are:

```
ndcontrol cluster configure 204.67.172.72 en0 255.255.0.0
(AIX)
ndcontrol cluster configure 204.67.172.72 eth0:1 255.255.0.0
(Linux)
ndcontrol cluster configure 204.67.172.72 le0:1 255.255.0.0
(Solaris 7)
ndcontrol cluster configure 204.67.172.72 le0 255.255.0.0
(Solaris 8)
ndcontrol cluster configure 204.67.172.72 en0 255.255.0.0
(Windows 2000)
```

Windows 2000

In order to use the second form of the cluster configure command on Windows 2000, you must determine the interface name to use.

If you have only one Ethernet card in your machine, the interface name will be en0. Likewise, if you have only one Token Ring card, the interface name will be tr0. If you have multiple cards of either type, you will need to determine the mapping of the cards. Use the following steps:

1. Start **regedit** at the command prompt.
2. Click **HKEY_LOCAL_MACHINE**, click **Software**, click **Microsoft**, click **Windows NT**, click **Current Version**.
3. And then, click **Network Cards**.

The network interface adapters are listed under Network Cards. Click each one to determine whether it is an Ethernet or Token Ring interface. The type of interface is listed in the *Description* column. The names assigned by **ndconfig** map to the interface types. For example, the first Ethernet interface in the list is assigned by ndconfig to en0, the second to en1, and so on; the first Token Ring interface is assigned to tr0, the second to tr1, and so on.

Note: The Windows 2000 registry begins numbering adapters with **1**, not **0**.

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

Using ifconfig/ndconfig to configure cluster aliases

The cluster configure command merely runs ifconfig (or ndconfig on Windows 2000) commands, so you can still use the ifconfig (ndconfig) commands if you wish.

Windows 2000: The ndconfig command is supplied with the Dispatcher component to configure cluster aliases using the command line. The ndconfig command has the same syntax as a UNIX ifconfig command.

```
ndconfig en0 alias 204.67.172.72 netmask 255.255.0.0
```

Note: The netmask parameter is required. It should be in dotted-decimal (255.255.0.0) or hex (0xffff0000) form.

To determine the interface name, use the same technique as for the second form of the cluster configure command.

Solaris: 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 Network Dispatcher machine. For example:

```
arp -s <cluster> <Network Dispatcher MAC address> pub
```

Step 6. Define ports and set port options

To define a port, enter the **ndcontrol port add *cluster:port*** command, edit the sample configuration file, or use the GUI. *Cluster* is either the symbolic name or the dotted-decimal 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 "Appendix B. Command reference for Dispatcher, CBR, and Mailbox Locator" on page 225.

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 will be used to configure rules and servers for any port. This function could also be used if you have an identical server/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 170 for more information about when you might want to use a wildcard port.

Note: The wildcard port cannot be used to handle FTP traffic.

Step 7. Define load-balanced server machines

To define a load-balanced server machine, enter the **ndcontrol 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 dotted-decimal 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. Since 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 packets/requests destined for the cluster.

Note: For Solaris and Linux: 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 **ndcontrol server** command. For more information on collocated servers, see “Using collocated servers” on page 140.

For more information on ndcontrol server command syntax, see “ndcontrol server — configure servers” on page 274.

Step 8. Start the manager function (optional)

The manager function improves load balancing. To start the manager, enter the **ndcontrol 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:

```
cbrcontrol advisor start http port
```

For a list of advisors along with their default ports, see “Appendix B. Command reference for Dispatcher, CBR, and Mailbox Locator” on page 225. For a description of each advisor, see “List of advisors” on page 129.

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 **ndcontrol cluster set cluster proportions** command. For more information, see “Proportion of importance given to status information” on page 122.

Setting up server machines for load balancing

If the server is collocated (Dispatcher resides on the same machine it load balances) or if using nat or cbr forwarding methods, do *not* perform the following procedures.

When using mac forwarding method, Dispatcher will only work with backend 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, Linux, Solaris, or Windows 2000), 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.

Note: There are a few **Linux** kernel versions which require a patch in order to alias the loopback device. See “Installing the Linux kernel patch (to suppress arp responses on the loopback interface)” on page 66, to determine whether a Linux kernel patch is required.

For **Linux** kernel versions 2.2.14 or higher, issue the following commands prior to the **ifconfig** command:

```
echo 1 > /proc/sys/net/ipv4/conf/lo/hidden
echo 1 > /proc/sys/net/ipv4/conf/all/hidden
```

If you have a server with an operating system that does not support aliases, such as HP-UX and OS/2, you must set the loopback device to the cluster address.

Use the command for your operating system as shown in Table 4 to set or alias the loopback device.

Table 4. Commands to alias the loopback device (lo0) for Dispatcher

AIX	<code>ifconfig lo0 alias cluster_address netmask netmask</code>
-----	---

Table 4. Commands to alias the loopback device (lo0) for Dispatcher (continued)

HP-UX	<code>ifconfig lo0 cluster_address</code>
Linux	<code>ifconfig lo:1 cluster_address netmask 255.255.255.255 up</code>
OS/2	<code>ifconfig lo cluster_address</code>
Solaris 7	<code>ifconfig lo0:1 cluster_address 127.0.0.1 up</code>
Solaris 8	<code>ifconfig lo0:1 plumb cluster_address netmask netmask up</code>
Windows 2000	<ol style="list-style-type: none"> 1. Click Start, click Settings, then click Control Panel. 2. If you have not done so already, add the MS Loopback Adapter Driver. <ol style="list-style-type: none"> a. Double-click Add/Remove Hardware. This launches the Add/Remove Hardware Wizard. b. Click Next, select Add/Troubleshoot a Device, then click Next. c. The screen blinks off/on, then presents the Choose a Hardware Device panel. d. If the MS Loopback Adapter is in the list, it is already installed— click Cancel to exit. e. If the MS Loopback Adapter is <i>not</i> in the list— select Add a New Device and click Next. f. To select the hardware from a list, for the Find New Hardware panel, click No and then click Next. g. Select Network Adapters and click Next. h. On the Select Network Adapter panel, select Microsoft in the Manufacturers list, then select Microsoft Loopback Adapter. i. Click Next, then click Next again to install the default settings (or select Have Disk, then insert CD and install from there). j. Click Finish to complete installation. 3. From the Control Panel, Double-click Network and Dial-up Connections. 4. Select the connection with Device Name “Microsoft Loopback Adapter” and right-click on it. 5. Select Properties from the dropdown. 6. Select Internet Protocol (TCP/IP), then click Properties. 7. Click Use the following IP address. Fill in <i>IP address</i> with the cluster address, and <i>Subnet mask</i> with the default subnet mask (255.0.0.0). Note: Don’t enter a router address. Use the localhost as the default DNS server.

Table 4. Commands to alias the loopback device (lo0) for Dispatcher (continued)

OS/390	<p>Configuring a loopback alias on OS/390 system</p> <ul style="list-style-type: none"> In the IP parameter member (file), an Administrator will need to create an entry in the Home address list. For example <pre>HOME ;Address Link 192.168.252.11 tr0 192.168.100.100 1tr1 192.168.252.12 loopback</pre> <ul style="list-style-type: none"> Several addresses can be defined for the loopback. The 127.0.0.1 is configured by default.
--------	--

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 2000 with the following command:
route print
- Check for an extra route on all UNIX systems with the following command:
netstat -nr

Windows 2000 Example:

- After **route print** is entered, a table similar to the following 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

- 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.
- 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 will be 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 5 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 5. Commands to delete any extra route for Dispatcher

HP-UX	<code>route delete cluster_address cluster_address</code>
Windows 2000	<code>route delete network_address cluster_address</code> (at an MS-DOS prompt) Note: You must delete the extra route every time you reboot the server.

Using the example shown in Figure 15 on page 57, and setting up a server machine that is running AIX, 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 Network Dispatcher 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.

Note: For **Linux** kernel versions 2.2.12 and 2.2.13, ensure that there is a "1" in `/proc/sys/net/ipv4/conf/lo/arp_invisible`.

For **Linux** kernel versions 2.2.14 or higher ensure that there is a "1" in `/proc/sys/net/ipv4/conf/lo/hidden` and `/proc/sys/net/ipv4/conf/all/hidden`.

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.

Installing the Linux kernel patch (to suppress arp responses on the loopback interface)

For Linux servers only, a specific patch (depending on the Linux kernel version) is required to alias the loopback device.

The patch ensures that an ARP response is sent only from a network adapter port that has the IP address requested in the ARP request. Without this patch, Linux will issue ARP responses on the network for loopback aliases. The patch also corrects an ARP race condition when multiple network adapter ports with different IP addresses are on the same physical network.

You must install the patch under the following conditions.

- **Linux kernel versions 2.4.x**

- If you are using Dispatcher’s MAC forwarding method with high availability and collocation, then you must install the patch to the Dispatcher box.

Note: Dispatcher can be considered collocated even when it is just load balancing another component of Edge Server (such as Caching Proxy, Mailbox Locator, CBR, etc.) on the same machine that it resides.

- If you are using the 2.4 kernel on a backend server being load balanced by a Dispatcher configured with the MAC forwarding method, then you must install the patch on the backend server machine.
- If the machine has multiple network adapter ports on the same physical network, then you must install the patch on the machine.

- **Linux kernel versions 2.2.12 and 2.2.13**

If you are using the 2.2.12 or 2.2.13 kernel on a backend server.

Notes:

1. Network Dispatcher will not run on a 2.2 kernel.
2. The patch is incorporated in the 2.2.14 kernel.
3. This Linux kernel patch was used to test the IBM product and was found to be satisfactory in the IBM testing environment. You should evaluate the usefulness of this code in your own environment, and decide if it meets your needs. This code may or may not be included in future versions of the Linux base source code.

Linux kernel versions 2.4.x

The kernel patch is not required for all configurations. You must install a patch for Linux kernel 2.4.x versions under the following conditions:

- If you are using Dispatcher's MAC forwarding method with high availability and collocation, then you must install the patch on the Dispatcher box.

Note: Dispatcher can be considered collocated even when it is just load balancing another component of Edge Server (such as Caching Proxy, Mailbox Locator, CBR, etc.) on the same machine that it resides.

- If you are using the 2.4 kernel on a backend server being load balanced by a Dispatcher configured with the MAC forwarding method, then you must install the patch on the backend server.
- If the machine has multiple network adapter ports on the same physical network, then you must install the patch on the machine.

You can download this patch from:

<http://oss.software.ibm.com/developerworks/opensource/cvs/naslib>.

Select CVS Tree in the Download list.

To apply the patch:

1. Obtain the loopback patch from
<http://oss.software.ibm.com/developerworks/opensource/cvs/naslib>.
2. Install the kernel RPMs:
 - a. Copy the patch file **arp.c.2.4.0.patch** to `/usr/src/linux-2.4/net/ipv4/`
 - b. Issue the following commands:

```
cd /usr/src/linux-2.4/net/ipv4
patch -p0 -l < arp.c.2.4.0.patch
```

Note: This has been tested with Linux kernel versions 2.4.0 and 2.4.2.

3. Change to the `/usr/src/linux-2.4` directory.
4. Edit Makefile and append **-arppatch** to the EXTRAVERSION value.
5. Issue the command: `make mrproper`

6. Issue the command: `make config`, and select the appropriate values for your system. Make sure you configure module support.
7. Issue the following commands:


```
make dep;make clean;make bzImage;make modules;make modules_install
cd arch/i386/boot
cat bzImage > /boot/vmlinuz-2.4.2-2-arppatch
cd /usr/src/linux-2.4
cp System.map /boot/System.map-2.4.2-2-arppatch
cd /etc
```
8. Edit `lilo.conf` and copy the **image=** paragraph. In the new copy, make the following changes:
 - `/boot/vmlinuz-2.4.2-2` to `/boot/vmlinuz-2.4.2-2-arppatch`
 - `label=linux` to `label=linux-arppatch`
 - `default=linux` to `default=linux-arppatch`
9. Run the command: `/sbin/lilo`.
10. Reboot into your new kernel.

Linux kernel versions 2.2.12 and 2.2.13

A patch for Linux kernel versions 2.2.12 and 2.2.13 must be installed on any server box using the MAC forwarding method. You can download this patch from: <http://www.ibm.com/developer/linux>.

To apply the patch:

1. Obtain the loopback patch from <http://www.ibm.com/developer/linux>.
2. Install the kernel source. For installation instructions, refer to the **README.kernel-sources** file in the `/usr/src/linux` directory.
3. Apply the patch by issuing the patch command from `/usr/src` directory. For example:


```
patch -p0 < patchfile
```
4. Compile the kernel. For compile instructions, refer to the **README** file in `/usr/src/linux-2.4/` directory.
5. Install the new kernel and run the **lilo** command. For instructions, refer to the **README** file in `/usr/src/linux` directory.
6. Reboot with the new kernel.
7. Check for the following file: `/proc/sys/net/ipv4/conf/lo/arp_invisible`. If the file is present then the kernel was patched successfully. If the file is *not* present, then either the patch was unsuccessful, or an unpatched kernel was booted. Check `/usr/src/linux/README` to make sure all the installation steps were followed correctly.
8. Issue the command:


```
echo 1 > /proc/sys/net/ipv4/conf/lo/arp_invisible
```

This command will only last until the machine is rebooted. Once rebooted it will be necessary to follow this and the subsequent steps

Chapter 6. Planning for the Content Based Routing component

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

- See “Chapter 7. Configuring the Content Based Routing component” on page 75 for information on configuring the load-balancing parameters of CBR.
- See “Chapter 14. Advanced Network Dispatcher Functions” on page 119 for information on how to set up Network Dispatcher for more advanced functions.
- See “Chapter 15. Operating and managing Network Dispatcher” on page 185 for information on remote authenticated administration, Network Dispatcher logs, and usage of the Network Dispatcher components.

This chapter includes the following sections:

- “Hardware and software requirements”
- “Planning considerations”

Hardware and software requirements

Platform requirements:

- For AIX, see “Requirements for AIX” on page 12
- For Linux, see “Requirements for Red Hat Linux or SuSE Linux” on page 16
- For Solaris, see “Requirements for Solaris” on page 19
- For Windows 2000, see “Requirements for Windows 2000” on page 21

Planning considerations

The CBR component allows you to load balance HTTP and SSL traffic using Caching Proxy to proxy the request

Note: You must install the reverse proxy mode of Caching Proxy in order to run CBR as a plugin.

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 will be 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. Network Dispatcher provides a Caching Proxy (ibmproxy) advisor. See “Advisors” on page 126 for more information.
- To configure and manage the executor, advisors, and manager, use the command line (**cbrcontrol**) or the graphical user interface (**ndadmin**).

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.

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 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 will be transparent to clients accessing your site. 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.

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.

Caching Proxy communicates with CBR through its plugin interface. Caching Proxy, must be installed on the same machine. Multiple instances of Caching Proxy running on the same machine can now communicate with CBR simultaneously. In earlier releases, only one instance of Caching Proxy could communicate 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.

Once you have defined 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.)

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.

A configuration statement needs to be added to the `ibmproxy.conf` file for IBM 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.

Since 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 7. Configuring the Content Based Routing component

Before following the steps in this chapter, see “Chapter 6. Planning for the Content Based Routing component” on page 71. This chapter explains how to create a basic configuration for the CBR component of Network Dispatcher.

- See “Chapter 14. Advanced Network Dispatcher Functions” on page 119 for more complex configurations of Network Dispatcher.
- See “Chapter 15. Operating and managing Network Dispatcher” on page 185 for information on remote authenticated administration, Network Dispatcher logs, and usage of the Network Dispatcher components.

Overview of configuration tasks

Note: 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.

Table 6. 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 84

Methods of configuration

To create a basic configuration for the CBR component of Network Dispatcher, 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). It is recommended that you modify the Caching Proxy service to start manually rather than automatically.

- For AIX, Linux, and Solaris: 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: 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:

- As root user, issue **cbrserver** command from the command prompt.

Note: To stop the service, issue the following: **cbrserver 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 “Appendix B. Command reference for Dispatcher, CBR, and Mailbox Locator” on page 225.
- Start Caching Proxy. Issue **ibmproxy** command from the command prompt. (You must start the executor prior to starting Caching Proxy.)

Note: For Windows 2000: Start Caching Proxy from the Services panel:

Start-> Settings-> 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 2000, the Dispatcher component’s ndserver starts automatically. If you are using only CBR and not the Dispatcher component, you can stop ndserver from starting automatically as follows:
 - a. In the Windows 2000 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 client=181.0.153.222&uri=http://10.1.203.4/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 "client=181.0.153.222&uri=http://10.1.203.4/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

The commands for configuring CBR can be entered into a configuration script file and executed together.

Note: To quickly execute the content of a script file (e.g. `myscript`), use either of the following commands:

- For updating the current configuration, run the executable commands from your script file using —
cbrcontrol file appendload *myscript*
- For completely replacing the current configuration, run the executable commands from your script file using —
cbrcontrol file newload *myscript*

GUI

For an example of the graphical user interface (GUI), see Figure 2 on page 5.

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. Next, do one of the following:
 - For AIX, Linux, or Solaris: enter **ndadmin**
 - For Windows 2000: click **Start**, click **Programs**, **IBM WebSphere**, click **Edge Server**, click **IBM Network Dispatcher**, and click **Network Dispatcher**
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, Linux, or Solaris: To start Caching Proxy, enter **ibmproxy**
 - For Windows 2000: To start Caching Proxy, go to the Services panel: **Start-> Settings-> 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 once 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 Network Dispatcher components.

You can access **Help** by clicking the question mark icon in the upper right hand corner of the Network Dispatcher window.

- **Field Help** — describes each field, default values
- **How do I** — lists tasks that can be done from that screen

- **Contents** — a table of contents of all the Help information
- **Index** — an alphabetical index of the help topics

For more information about using the GUI, see “General Instructions for using the GUI” on page 6.

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, Linux, or Solaris: To start Caching Proxy, enter **ibmproxy**
For Windows 2000: To start Caching Proxy, go to the Services panel:
Start-> Settings-> 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 setup a cluster that enables CBR to load balance traffic between a group of servers.

With the CBR configuration wizard, you will see the following panels:

- Introduction to the wizard
- What to expect
- Before you start
- Choosing a host to configure (if necessary)
- Defining a cluster
- Adding a port
- Adding a server
- Adding a rule
- Starting an advisor

Setting up the CBR machine

Before setting up the CBR machine, you must be the root user (for AIX, Linux, or Solaris) or the Administrator on Windows 2000.

You will need one IP address for each cluster of servers that will be 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 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. It is recommended that you modify the Caching Proxy service to start manually rather than automatically.

- For AIX, Linux, and Solaris: 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: Stop Caching Proxy from the Services panel.

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

Change the incoming URL directive **CacheByIncomingUrl** to specify "on".

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

- ServerInit
- PreExit
- 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 AIX, Linux, Solaris, and Windows 2000 follow.

Figure 16. CBR configuration file for AIX

```
ServerInit /usr/lpp/nd/servers/lib/libndcbr.so:ndServerInit
PreExit /usr/lpp/nd/servers/lib/libndcbr.so:ndPreExit
PostExit /usr/lpp/nd/servers/lib/libndcbr.so:ndPostExit
ServerTerm /usr/lpp/nd/servers/lib/libndcbr.so:ndServerTerm
```

Figure 17. CBR configuration file for Linux

```
ServerInit /opt/nd/servers/lib/libndcbr.so:ndServerInit
PreExit /opt/nd/servers/lib/libndcbr.so:ndPreExit
PostExit /opt/nd/servers/lib/libndcbr.so:ndPostExit
ServerTerm /opt/nd/servers/lib/libndcbr.so:ndServerTerm
```

Figure 18. CBR configuration file for Solaris

```
ServerInit /opt/nd/servers/lib/libndcbr.so:ndServerInit
PreExit /opt/nd/servers/lib/libndcbr.so:ndPreExit
PostExit /opt/nd/servers/lib/libndcbr.so:ndPostExit
ServerTerm /opt/nd/servers/lib/libndcbr.so:ndServerTerm
```

Figure 19. CBR configuration file for Windows 2000

Common install directory path:

```
ServerInit c:\Progra~1\IBM\edge\nd\servers\lib\libndcbr.dll:ndServerInit
PreExit c:\Progra~1\IBM\edge\nd\servers\lib\libndcbr.dll:ndPreExit
PostExit c:\Progra~1\IBM\edge\nd\servers\lib\libndcbr.dll:ndPostExit
ServerTerm c:\Progra~1\IBM\edge\nd\servers\lib\libndcbr.dll:ndServerTerm
```

Native install directory path:

```
ServerInit c:\Program~1\IBM\nd\servers\lib\libndcbr.dll:ndServerInit
PreExit c:\Program~1\IBM\nd\servers\lib\libndcbr.dll:ndPreExit
PostExit c:\Program~1\IBM\nd\servers\lib\libndcbr.dll:ndPostExit
ServerTerm c:\Program~1\IBM\nd\servers\lib\libndcbr.dll:ndServerTerm
```

Step 2. Start the server function

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

- For AIX, Linux, and Solaris: 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: Stop Caching Proxy from the Services panel.

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 will be 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 “ndcontrol executor — control the executor” on page 239.

Step 4. Define a cluster and set cluster options

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

The cluster address is either a symbolic name or a dotted decimal address. This address will be located in the host portion of the URL.

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 “Appendix B. Command reference for Dispatcher, CBR, and Mailbox Locator” on page 225.

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 Network Dispatcher box. Otherwise, this step can be omitted.

For **AIX, Linux, or Solaris**: To add the cluster address to the network interface, use the `ifconfig` command. Use the command for your operating system as shown in Table 7.

Table 7. Commands to alias the NIC

AIX	<code>ifconfig interface_name alias cluster_address netmask netmask</code>
Linux	<code>ifconfig interface_name cluster_address netmask netmask up</code>
Solaris 7	<code>ifconfig interface_name cluster_address netmask netmask up</code>
Solaris 8	<code>ifconfig addif interface_name cluster_address netmask netmask up</code>

Note: For Linux and Solaris, *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**: To add the cluster address to the network interface, do the following:

1. Click **Start**, click **Settings**, then click **Control Panel**.
2. Double-click **Network and Dial-up Connections**.
3. Right-click **Local Area Connection**.
4. Select **Properties**.
5. Select **Internet Protocol (TCP/IP)** and click **Properties**.
6. Select **Use the following IP address** and click **Advanced**.
7. 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 that the server applications are listening on. 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:

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

To set port options, issue the following:

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

For more information, see “Appendix B. Command reference for Dispatcher, CBR, and Mailbox Locator” on page 225.

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 w/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 will be compared to the URL in each client request. For more information on how to configure the pattern, see “Appendix C. Content rule (pattern) syntax” on page 285.

Some other rule types defined in Dispatcher can also be used in CBR. For more information, see “Configure rules-based load balancing” on page 157.

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 122.

Step 13. Start Caching Proxy

- AIX platform: Add to your LIBPATH environment variable:
`/usr/lpp/nd/servers/lib`
- Linux or Solaris platform: Add to your LD_LIBRARY_PATH environment variable:
`/opt/nd/servers/lib`
- Windows 2000 platform: Add to your PATH environment variable:
Common install directory path:
`c:\Program Files\IBM\edge\nd\servers\lib`

Native install directory path:
`c:\Program Files\IBM\nd\servers\lib`

In the new environment, start Caching Proxy: From the command prompt, issue **ibmproxy**

Note: For Windows 2000: Start Caching Proxy from the Services panel: **Start-> Settings-> 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 will be given. 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 2000, 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 you configured above.
 - Server `"s"` is invoked
 - The following Web page is displayed `http://s/`

Chapter 8. Planning for the Mailbox Locator component

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

Note: The Mailbox Locator component was formerly a feature within the CBR component that load balanced across IMAP and POP3 mail servers, based on userID and password. Separating CBR into two components *removes* the limitation that "CBR for IMAP/POP3" (Mailbox Locator) and "CBR for HTTP/HTTPS" (CBR with Caching Proxy) cannot be run on the same machine.

- See "Chapter 9. Configuring the Mailbox Locator component" on page 91 for information on configuring the load-balancing parameters of Mailbox Locator.
- See "Chapter 14. Advanced Network Dispatcher Functions" on page 119 for information on how to set up Network Dispatcher for more advanced functions.
- See "Chapter 15. Operating and managing Network Dispatcher" on page 185 for information on remote authenticated administration, Network Dispatcher logs, and usage of the Network Dispatcher components.

This chapter includes the following sections:

- "Hardware and software requirements"
- "Planning considerations"

Hardware and software requirements

- For AIX, see "Requirements for AIX" on page 12
- For Linux, see "Requirements for Red Hat Linux or SuSE Linux" on page 16
- For Solaris, see "Requirements for Solaris" on page 19
- For Windows 2000, see "Requirements for Windows 2000" on page 21

Planning considerations

The Mailbox Locator component allows you to proxy IMAP and POP3 traffic based on userID and password of the client request.

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

- **mlserver** handles requests from the command line to the executor, manager, and advisors.
- The **executor** supports load balancing of client requests. The executor always runs when the Mailbox Locator component is being used.
- 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 will be 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. See “Advisors” on page 126 for more information.
- To configure and manage the executor, advisors, and manager, use the command line (**mlcontrol**) or the graphical user interface (**ndadmin**).

The three key functions of Mailbox Locator (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.

To start Mailbox Locator, issue the **mlserver** command from the command prompt.

Mailbox Locator can provide a single point of presence for many IMAP or POP3 servers. Each server can have a subset of all mailboxes serviced by the point of presence. For IMAP and POP3, Mailbox Locator is a proxy that chooses an appropriate server based on userID and password provided by the client.

Note: The Mailbox Locator does *not* support **rules**-based load balancing.

An example of a method for distributing requests based on client userID is the following. If you have two (or more) POP3 servers, you can choose to divide the mailboxes alphabetically by userID. Client requests with userID's beginning with letters A–I can be distributed to server 1. Client requests with userID's beginning with letters J–R can be distributed on server 2, and so on.

You can also choose to have each mailbox represented on more than one server. In that case, the content of each mailbox must be available to all servers with that mailbox. In the event of a server failure, another server can still access the mailbox.

In order to have only one address representing multiple POP3 mail servers, Mailbox Locator can be configured with a single cluster address that becomes the POP3 mail server address for all clients. The commands to configure this are the following:

```
mlcontrol cluster add pop3MailServer
mlcontrol port add pop3MailServer:110 protocol pop3
mlcontrol server add pop3MailServer:110:pop3Server1+pop3Server2+pop3Server3
```

In this example, *pop3MailServer* represents the cluster address. Port 110 with proxy protocol POP3 is added to the *pop3MailServer*. *Pop3Server1*, *pop3Server2*, and *pop3Server3* represent POP3 mail servers which are added to the port. With this configuration, you can configure your mail clients' incoming POP3 requests with the *pop3MailServer* cluster address.

Using the affinity feature

When a POP3 or IMAP request arrives at the proxy, the proxy attempts to contact all the configured servers for the port using the client's userID and password. The client's request is directed to the first server that responds. You should use the sticky/affinity feature in conjunction with the Mailbox Locator for IMAP or POP3 servers. The affinity feature allows subsequent requests from the same client's userID to be directed to the same server. Set **stickytime** for the port to a value greater than zero to enable this affinity feature. For more information on the affinity feature, see "How affinity feature for Network Dispatcher works" on page 170.

Overriding the POP3/IMAP inactivity timer

The inactivity autologout timer for POP3 and IMAP protocols is a minimum of 10 minutes and 30 minutes respectively. This timeout is the number of seconds during which there can be no activity on a connection before that connection is removed. To optimize performance, Mailbox Locator overrides the inactivity timeout value to 60 seconds. In order to change the inactivity timeout, change the **staletimeout** value on the **mlcontrol port** command. For information on configuring this command, see "ndcontrol port — configure ports" on page 261.

Chapter 9. Configuring the Mailbox Locator component

Before following the steps in this chapter, see “Chapter 8. Planning for the Mailbox Locator component” on page 87. This chapter explains how to create a basic configuration for the Mailbox Locator component of Network Dispatcher.

Note: The Mailbox Locator component was formerly a feature within the CBR component that load balanced across IMAP and POP3 mail servers, based on userID and password. Separating CBR into two components *removes* the limitation that “CBR for IMAP/POP3” (Mailbox Locator) and “CBR for HTTP/HTTPS” (CBR with Caching Proxy) cannot be run on the same machine.

- See “Chapter 14. Advanced Network Dispatcher Functions” on page 119 for more complex configurations of Network Dispatcher.
- See “Chapter 15. Operating and managing Network Dispatcher” on page 185 for information on remote authenticated administration, Network Dispatcher logs, and usage of the Network Dispatcher components.

Overview of configuration tasks

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

Table 8. Configuration tasks for the Mailbox Locator component

Task	Description	Related information
Set up the Mailbox Locator machine.	Finding out about the requirements.	“Setting up the Mailbox Locator machine” on page 94
Set up machines to be load-balanced.	Set up your load balancing configuration.	“Step 4. Define load balanced server machines” on page 96

Methods of configuration

To create a basic configuration for the Mailbox Locator component of Network Dispatcher, there are four basic methods:

- Command line
- Scripts

- Graphical user interface (GUI)
- Configuration wizard

Command line

This is the most direct means of configuring Mailbox Locator. 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 Mailbox Locator from the command line:

- Issue the **mlserver** command from the command prompt.

Note: To stop the service, issue the following: **mlserver stop**.

- Next, issue the Mailbox Locator 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 **mlcontrol**. For more information about commands, see “Appendix B. Command reference for Dispatcher, CBR, and Mailbox Locator” on page 225.

You can enter a minimized version of the **mlcontrol** 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 **mlcontrol he f** instead of **mlcontrol help file**.

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

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

Note: On Windows 2000, the Dispatcher component’s **ndserver** starts automatically. If you are using only Mailbox Locator and not the Dispatcher component, you can stop **ndserver** from starting automatically as follows:

1. In the Windows 2000 Services window, 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 Mailbox Locator can be entered into a configuration script file and executed together.

Note: To quickly execute the content of a script file (e.g. **myscript**), use either of the following commands:

- For updating the current configuration, run the executable commands from your script file using —
mlcontrol file appendload *myscript*
- For completely replacing the current configuration, run the executable commands from your script file using —
mlcontrol file newload *myscript*

GUI

For an example of the GUI, see Figure 2 on page 5.

To start the GUI, follow these steps

1. Ensure mlserver is running. As root user or administrator, issue the following from a command prompt: **mlserver**
2. Next, do one of the following:
 - For AIX, Linux, or Solaris: Enter **ndadmin**
 - For Windows 2000: click **Start**, click **Programs**, **IBM WebSphere**, click **Edge Server**, click **IBM Network Dispatcher**, and click **Network Dispatcher**

In order to configure the Mailbox Locator component from the GUI, you must first select **Mailbox Locator** in the tree structure. You can start the manager once 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 **mlcontrol** command. For example, to define a cluster using the command line, you would enter **mlcontrol 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 Mailbox Locator 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 Mailbox Locator 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 Network Dispatcher components.

You can access **Help** by clicking the question mark icon in the upper right hand corner of the Network Dispatcher window.

- **Field Help** — describes each field, default values
- **How do I** — lists tasks that can be done from that screen

- **Contents** — a table of contents of all the Help information
- **Index** — an alphabetical index of the help topics

For more information about using the GUI, see “General Instructions for using the GUI” on page 6.

Configuration wizard

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

1. Issue the **mlserver** command at a command prompt as root user or administrator.
2. Start the wizard function of Mailbox Locator, **mlwizard**.

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

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

With the Mailbox Locator configuration wizard, you will see the following panels:

- Introduction to the wizard
- What to expect
- Before you start
- Choosing a host to configure (if necessary)
- Defining a cluster
- Adding a port
- Adding a server
- Starting an advisor

Setting up the Mailbox Locator machine

Before setting up the Mailbox Locator machine, you must be the root user (for AIX, Linux, or Solaris) or the Administrator on Windows 2000.

You will need one IP address for each cluster of servers that will be set up. 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. All requests made to the same cluster address are load balanced by Mailbox Locator.

Step 1. Start the server function

To start the server function, type `mlserver` on the command line.

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

Step 2. Define a cluster and set cluster options

Mailbox Locator will balance the requests sent for the cluster address to the corresponding servers configured on the ports for that cluster.

The cluster address is either the symbolic name or a dotted decimal address.

To define a cluster, issue the following command:

```
mlcontrol cluster add cluster
```

To set cluster options, issue the following command:

```
mlcontrol cluster set cluster option value
```

For more information, see “Appendix B. Command reference for Dispatcher, CBR, and Mailbox Locator” on page 225.

Step 3. Define ports and set port options

The port number is the port that the server applications are listening on. For IMAP traffic, this is typically port 143. And, for POP3 traffic, this is typically port 110.

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

```
mlcontrol port add cluster:port protocol [pop3|imap]
```

To set port options, issue the following:

```
mlcontrol port set cluster:port option value
```

Note: When adding a port, you must specify the proxy protocol (`pop3` or `imap`). Once you add the port, you cannot change (set) the existing protocol value for this port.

For more information, see “Appendix B. Command reference for Dispatcher, CBR, and Mailbox Locator” on page 225.

Step 4. Define load balanced server machines

The mail servers 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 from step 3, issue the following command:

```
mlcontrol server add cluster:port:server
```

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

Step 5. Start the manager function (optional)

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

```
mlcontrol 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 Network Dispatcher supplies IMAP and POP3 advisors. For example, to start the IMAP advisor, issue the following command:

```
mlcontrol advisor start imap port
```

For a list of advisors along with their default ports, see “Appendix B. Command reference for Dispatcher, CBR, and Mailbox Locator” on page 225. For a description of each advisor, see “List of advisors” on page 129.

Step 7. 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 **mlcontrol cluster set *cluster proportions*** command. For more information, see “Proportion of importance given to status information” on page 122.

Chapter 10. Planning for the Site Selector component

This chapter describes what the network planner should consider before

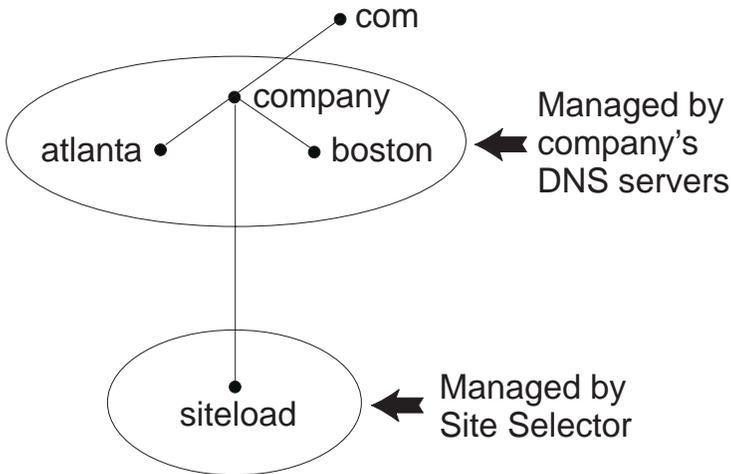


Figure 20. Example of a DNS environment

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 20), 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, 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 11 on page 39 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 will be 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 Network Dispatcher that you install on the backend server machine. (If you collocate Network Dispatcher on a server machine that is being load balanced, then you would install Metric Server on the Network Dispatcher 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, taking into account 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 136 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 126 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 (**ndadmin**).

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 311, 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 will be 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 will be 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 "Appendix D. Command reference for Site Selector" on page 289 for more information.

Chapter 11. Configuring the Site Selector component

Before following the steps in this chapter, see “Chapter 10. Planning for the Site Selector component” on page 97. This chapter explains how to create a basic configuration for the Site Selector component of Network Dispatcher.

- See “Chapter 14. Advanced Network Dispatcher Functions” on page 119 for more complex configurations of Network Dispatcher.
- See “Chapter 15. Operating and managing Network Dispatcher” on page 185 for information on remote authenticated administration, Network Dispatcher logs, and usage of the Network Dispatcher 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 9. 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 106
Set up machines to be load-balanced.	Set up your load balancing configuration.	“Step 4. Define load balanced server machines” on page 107

Methods of configuration

To create a basic configuration for the Site Selector component of Network Dispatcher, 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:

- Issue the **ssserver** command from the command prompt.

Note: To stop the service, issue the following: **ssserver stop**.

- 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 “Appendix D. Command reference for Site Selector” on page 289.

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 hf** 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 2000, the Dispatcher component’s **ndserver** starts automatically. If you are using only Site Selector and not the Dispatcher component, you can stop **ndserver** from starting automatically as follows:

1. In the Windows 2000 Services window, 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 executed together.

Note: To quickly execute the content of a script file (e.g. *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`

GUI

For an example of the GUI, see Figure 2 on page 5.

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, Linux, or Solaris: enter **ndadmin**
 - For Windows 2000: click **Start**, click **Programs**, **IBM WebSphere**, click **Edge Server**, click **IBM Network Dispatcher**, and click **Network Dispatcher**

In order to configure the Site Selector component from the GUI, you must first select **Site Selector** in the tree structure. You can start the manager once you connect to a Host. You can also create site names containing ports and servers, and start advisors for the manager.

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 Network Dispatcher components.

You can access **Help** by clicking the question mark icon in the upper right hand corner of the Network Dispatcher window.

- **Field Help** — describes each field, default values
- **How do I** — lists tasks that can be done from that screen
- **Contents** — a table of contents of all the Help information
- **Index** — an alphabetical index of the help topics

For more information about using the GUI, see “General Instructions for using the GUI” on page 6.

Configuration wizard

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

1. Start the ssserver on Site Selector by issuing **ssserver** on the command prompt as root user or administrator.
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.

With the Site Selector configuration wizard, you will see the following panels:

- Introduction to the wizard
- What to expect
- Before you start
- Choosing a host to configure (if necessary)
- Defining a site name
- Adding a server
- Starting an advisor
- Setting network proximity

Setting up the Site Selector machine

Before setting up the Site Selector machine, you must be the root user (for AIX, Linux, or Solaris) or the Administrator on Windows 2000.

You will need an unresolvable DNS 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

AIX, Linux, and Solaris: To start the server function, enter **ssserver**.

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 will be 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 (e.g. `www.dnsdownload.com`). When a client requests this site name, one of the server IP addresses associated with the site name will be 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 “Appendix D. Command reference for Site Selector” on page 289.

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 improves 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 Network Dispatcher 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 136 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 122.

Setting up server machines for load balancing

It is recommended to use Metric Server with the Site Selector component. Refer to “Metric Server” on page 136 for information on setting up Metric Server on all server machines that Site Selector is load balancing.

Chapter 12. Planning for the Consultant for Cisco CSS Switches component

This chapter describes what the network planner should consider before installing and configuring the Consultant for Cisco CSS Switches component.

- See “Chapter 13. Configuring the Consultant for Cisco CSS Switches component” on page 113 for information on configuring the load-balancing parameters of the Consultant for Cisco CSS Switches component.
- See “Chapter 14. Advanced Network Dispatcher Functions” on page 119 for information on how to set up Network Dispatcher for more advanced functions.
- See “Chapter 15. Operating and managing Network Dispatcher” on page 185 for information on remote authenticated administration, Network Dispatcher logs, and usage of the Network Dispatcher components.

This chapter includes:

- “Hardware and software requirements”
- “Planning considerations”

Hardware and software requirements

- For AIX, see “Requirements for AIX” on page 12
- For Linux, see “Requirements for Red Hat Linux or SuSE Linux” on page 16
- For Solaris, see “Requirements for Solaris” on page 19
- For Windows 2000, see “Requirements for Windows 2000” on page 21

Planning considerations

The configuration for Cisco Consultant is dependent on the configuration for the Cisco CSS Switch (see Table 10 on page 111). After you complete planning and configuration for the Cisco CSS Switch, you can configure and use Cisco Consultant. Refer to the Cisco CSS Switch documentation for planning and configuration instructions.

Consultant consists of the following:

- **lbserver** contains the configuration information and interacts with the Cisco CSS Switch. The “lbc” prefix means load-balancing consultant. lbserver is comprised of:
 - The **executor**, which holds the configuration information and contains the information required to connect to the Cisco CSS Switch.

- The **manager**, which uses information gathered to generate weights and send them to the Cisco CSS Switch. The manager collects information from:
 - The Cisco CSS Switch
 - Servers (using the advisors)

The advisors query the servers and analyze results by protocol before calling the manager to set weights as appropriate. Currently, Cisco Consultant provides advisors such as HTTP, FTP, SSL, SMTP, NNTP, IMAP, POP3 (and others). You also have the option of writing your own advisors (see “Create custom (customizable) advisors” on page 131). Using the advisors is recommended, but is optional.
 - Servers (using the Metric Server)

Metric Server provides server load information to Consultant in the form of system-specific metrics, reporting on the health of the servers. The manager queries the Metric Server residing on each of the servers, using the metrics gathered from the agents to assist in assigning weights to the load-balancing process. The results are also placed into the manager report.
- Both a command line and a graphical user interface are provided to configure and manage the executor, advisors, and manager.
 - **lbcontrol** is the command line interface to Consultant.
 - **ndadmin** is the graphical user interface used to configure Consultant and monitor its status.

The manager collects information from the Cisco CSS Switch, the advisors, and Metric Server. Based on the information the manager receives, it adjusts how the server machines are weighted on each port and gives the Cisco CSS Switch the new weighting for use in its balancing of new connections. When the manager discovers that a server is down, it assigns that server a weight of zero, and the server is suspended. Subsequently, the Cisco CSS Switch stops forwarding traffic to that server.

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.

To properly configure Consultant, your configuration must mirror the Cisco CSS Switch configuration. First, refer to the *Cisco Services Switch Getting Started Guide* to configure the Cisco CSS Switch. Ensure that the switch is working correctly, then configure Consultant.

The Cisco CSS Switch configuration consists of owners, content rules, and services that map to a Consultant configuration as follows:

Table 10. Consultant and Cisco CSS Switch configuration terms

Cisco CSS Switch	Consultant
virtual IP address (VIP) of one or more of the owner's content rules	cluster
port contained in the content rule	port
service	server

The Consultant configuration tree consists of:

- *Cluster*, which is either a resolvable name or the dotted-decimal address.
- *Port*, which is the number of the port you are using for that protocol.
- *Servers*.

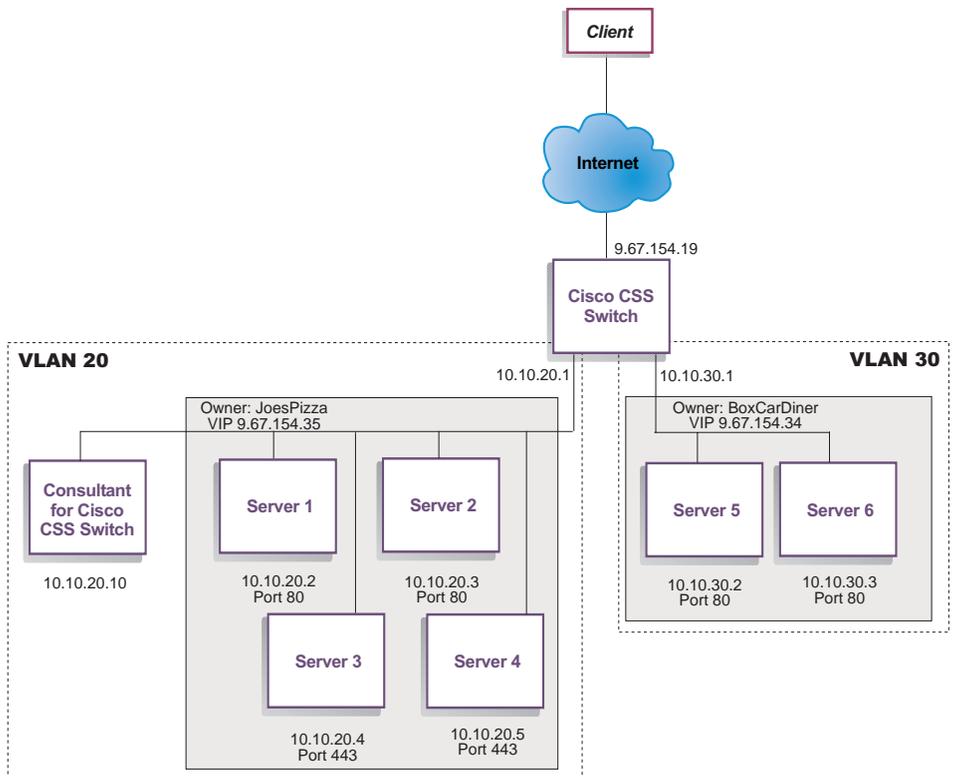


Figure 21. Example of Consultant configured with 2 clusters, each with 3 ports

In Figure 21:

- 9.67.154.19 is the network connection to the Internet.
- There are two VLANs (20 and 30) configured.

When configuring the executor, you must configure an address and SNMP community name, and these must match the corresponding attributes on the Cisco CSS Switch. See “lbcontrol executor — control the executor” on page 323 for information on configuring the executor.

Table 11. Example of the Cisco CSS Switch configuration mapped to the Consultant configuration

Cisco CSS Switch configuration	Consultant configuration
username admin superuser snmp community <i>community</i> private read-write	lbcontrol executor set address 10.10.20.1 lbcontrol executor set communityname <i>community</i>
content rule1 port <i>80</i> balance weightedrr add service <i>server1</i> add service <i>server2</i> vip address <i>9.67.154.35</i> active	lbcontrol cluster add <i>9.67.154.35</i> lbcontrol port add 9.67.154.35: <i>80</i>
content rule 2 protocol tcp port <i>443</i> balance weightedrr add service server3 add service server4 vip address 9.67.154.35 active	lbcontrol port add 9.67.154.35: <i>443</i>
service server1 ip address <i>10.10.20.2</i> port <i>80</i> weight 4 active	lbcontrol server add 9.67.154.35: <i>80</i> :server1 address <i>10.10.20.2</i>
service server3 ip address <i>10.10.20.4</i> port <i>443</i> weight 4 active	lbcontrol server add 9.67.154.35: <i>443</i> :server3 address <i>10.10.20.4</i>

Chapter 13. Configuring the Consultant for Cisco CSS Switches component

Before following the steps in this chapter, see “Chapter 12. Planning for the Consultant for Cisco CSS Switches component” on page 109. This chapter explains how to create a basic configuration for the Consultant for Cisco CSS Switches component of Network Dispatcher.

- See “Chapter 14. Advanced Network Dispatcher Functions” on page 119 for more complex configurations of Network Dispatcher.
- See “Chapter 15. Operating and managing Network Dispatcher” on page 185 for information on remote authenticated administration, Network Dispatcher logs, and usage of the Network Dispatcher components.

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 Consultant, ensuring that executor’s address and SNMP community name match the corresponding attributes on the Cisco CSS Switch. See “lbcontrol executor — control the executor” on page 323 for information on configuring the executor.

Table 12. Configuration tasks for the Consultant for Cisco CSS Switches component

Task	Description	Related information
Set up the Consultant for Cisco CSS Switches machine	Finding out about the requirements	“Setting up the Consultant for Cisco CSS Switches machine” on page 116
Test your configuration	Confirm that the configuration is working	“Testing your configuration” on page 118

Methods of configuration

To create a basic configuration for the Consultant for Cisco CSS Switches component of Network Dispatcher, there are three methods:

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

Command line

This is the most direct means of configuring Cisco Consultant. 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 cluster and server commands) and file names.

To start Cisco Consultant from the command line:

- Issue the **lbserver** command from the command prompt.

Note: To stop the service, issue the following: **lbserver stop**.

- Next, issue the Cisco Consultant control commands you want in order to set up your configuration. The command is **lbcontrol**. For more information about commands, see “Appendix B. Command reference for Dispatcher, CBR, and Mailbox Locator” on page 225.

You can enter an abbreviated version of the **lbcontrol** 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 **lbcontrol he f** instead of **lbcontrol help file**.

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

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

Note: On Windows 2000, the Dispatcher component’s **ndserver** starts automatically. If you are using only Cisco Consultant and not the Dispatcher component, you can stop **ndserver** from starting automatically as follows:

1. In the Windows 2000 Services window, 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 Consultant for Cisco CSS Switches can be entered into configuration script file and executed together.

Note: To quickly execute the content of a script file (e.g. *myscript*), use either of the following commands:

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

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

GUI

For an example of the graphical user interface (GUI), see Figure 2 on page 5.

To start the GUI, follow these steps

1. If lbserver is not already running, start it now by running the following as root:
lbserver.
2. Next, do one of the following:
 - For AIX, Linux, or Solaris: enter **ndadmin**
 - For Windows 2000: click **Start**, click **Programs**, **IBM WebSphere**, click **Edge Server**, click **IBM Network Dispatcher**, and click **Network Dispatcher**

To configure the Cisco Consultant component from the GUI:

1. Right-click Cisco Consultant in the tree structure
2. Connect to a Host
3. Create clusters containing ports and servers
4. Start the manager
5. Start advisors for the manager

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

You can use 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 to load pre-existing Cisco Consultant configuration files. Select **Save Configuration File As** option to periodically save your Cisco Consultant configuration to a file. Click **File** on the menu bar to save your current host connections to a file or restore connections in existing files across all Network Dispatcher components.

To access **Help** click the question mark icon in the upper right corner of the Network Dispatcher window.

- **Field Help** — describes each field, default values
- **How do I** — lists tasks that can be done from that screen

- **Contents** — a table of contents of all the Help information
- **Index** — an alphabetical index of the help topics

For more information about using the GUI, see “General Instructions for using the GUI” on page 6.

Setting up the Consultant for Cisco CSS Switches machine

Before setting up the Consultant for Cisco CSS Switches machine, you must be the root user (for AIX, Linux, or Solaris) or the Administrator on Windows 2000.

Consultant must be able to connect to the Cisco CSS Switch as a Cisco CSS Switch administrator.

When configuring the executor, you must configure the address and SNMP community name must match the corresponding attributes on the Cisco CSS Switch.

For help with commands used in this procedure, see “Appendix E. Command reference for Consultant for Cisco CSS Switches” on page 315.

Step 1. Start the server function

If the `lbserver` is not already running, start it now by running the following as root:

```
lbserver
```

Step 2. Configure the executor function

You must configure an address and SNMP community name. These values must match the corresponding attributes on the Cisco CSS Switch.

Step 3. Define a cluster and set cluster options

Cluster is either a resolvable name or the dotted-decimal address. The cluster corresponds to the Cisco CSS Switch’s virtual IP address of an owner’s content rule.

To define a cluster, type `lbcontrol cluster add cluster`. To set cluster options, type `lbcontrol cluster set`.

Step 4. Define ports and set port options

To define a port, type `lbcontrol port add cluster:port`. The port corresponds to the port configured in the Cisco CSS Switch content rule for the owner.

Port is the number of the port you are using for that protocol as specified in the owner content rule for the Cisco CSS Switch. See “`lbcontrol port — configure ports`” on page 338 for more information.

Step 5. Define load-balanced server machines

You can configure multiple instances of the same server within any cluster and port. (Remember that the address and SNMP community name must match the corresponding attributes on the Cisco CSS Switch.) When you configure multiple instances of the same server, you can distinguish different application servers that reside on the same physical machine and respond to the same IP address on the same port.

To define a load-balanced server machine, type:

```
lbcontrol server add cluster:port:server address x.x.x.x | hostname
```

The *server* corresponds to the Cisco CSS Switch service name.

You must define more than one server to a port on a cluster to perform load balancing, or traffic will be directed to only one server. See “Server Partitioning: logical servers configured to one physical server (IP address)” on page 138.

For more information on lbcontrol server command syntax, see “lbcontrol server — configure servers” on page 340.

Step 6. Start the manager function

To start the manager, type the **lbcontrol manager start** command. See “lbcontrol manager — control the manager” on page 330 for more information.

Step 7. 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:

```
lbcontrol advisor start http port
```

For a list of advisors along with their default ports, see “lbcontrol advisor — control the advisor” on page 316. For a description of each advisor, see “List of advisors” on page 129.

Step 8. Set cluster proportions as required

If you start any advisors, you must change the cluster proportions so that advisor information is included in the load-balancing decisions. Use the **lbcontrol cluster proportions** command. See “Proportion of importance given to status information” on page 122.

Note: If you start an advisor and the **Proportion given to system metrics** is 0, it is increased to 1. Since cluster proportions must total 100, in this case, the proportion with the highest value is decreased by 1.

Step 9. Start Metric Server (optional)

See “Metric Server” on page 136 for information on using Metric Server.

Testing your configuration

Test to see if the configuration is working.

1. Set the manager 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 manager loglevel back to the desired level (1).
5. View the manager.log file located in the .../nd/servers/logs/lbc directory, and look for **setServerWeights setting service**.

Chapter 14. Advanced Network Dispatcher Functions

This chapter explains how to configure the load balancing parameters of Network Dispatcher and how to set up Network Dispatcher for advanced functions.

Note: When reading this chapter, if you are *not* using the Dispatcher component, then substitute "ndcontrol" with the following:

- For CBR, use **cbrcontrol**
- For Mailbox Locator, use **mlcontrol**
- For Site Selector, use **sscontrol** (see "Appendix D. Command reference for Site Selector" on page 289)
- For Cisco Consultant, use **lbcontrol** (see "Appendix E. Command reference for Consultant for Cisco CSS Switches" on page 315)

Table 13. Advanced configuration tasks for the Network Dispatcher

Task	Description	Related information
Optionally, change load-balancing settings	<p>You can change the following load-balancing settings:</p> <ul style="list-style-type: none"> • Proportion of importance given to status information <p>The default ratio is 50-50-0-0. If you use the default, information from advisors and from Metric Server is not used.</p> <ul style="list-style-type: none"> • Weights • Manager fixed weights • Manager intervals • Sensitivity threshold • Smoothing index 	"Optimizing the load balancing provided by Network Dispatcher" on page 122
Use scripts to generate an alert or record server failure when manager marks server(s) down/up	Network Dispatcher provides user exits that trigger scripts that you can customize when the manager marks server(s) down/up	"Using scripts to generate an alert or record server failure" on page 126
Use advisors and create custom advisors	Describes the advisors and how to write your own custom advisors for reporting on specific statuses of your servers	"Advisors" on page 126 "Create custom (customizable) advisors" on page 131
Use Workload Manager advisor (WLM)	WLM advisor provides system load information to Network Dispatcher	"Workload Manager advisor" on page 135

Table 13. Advanced configuration tasks for the Network Dispatcher (continued)

Task	Description	Related information
Use Metric Server agent	Metric Server provides system load information to Network Dispatcher	“Metric Server” on page 136
Use Server partitioning	Define logical servers to distribute load based on services provided	“Server Partitioning: logical servers configured to one physical server (IP address)” on page 138
Use advisor request/response (URL) option	Define a unique client HTTP URL string, specific for a service that you want to query on the machine	“HTTP advisor request/response (URL) option” on page 140
Collocate Network Dispatcher on a machine that it is load balancing	Set up a collocated Network Dispatcher machine.	“Using collocated servers” on page 140
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 142
Configure high availability or mutual high availability	Set up a second Dispatcher machine to provide a backup.	“High availability” on page 150
Configure rules-based load balancing	Define conditions under which a subset of your servers will be used.	“Configure rules-based load balancing” on page 157
Use explicit linking	Avoid bypassing the Dispatcher in your links.	“Using explicit linking” on page 167
Use a private network	Configure the Dispatcher to load balance servers on a private network.	“Using a private network configuration” on page 167
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 168
Use wildcard cluster to load balance firewalls	All traffic will be load balanced to firewalls.	“Use wildcard cluster to load balance firewalls” on page 169
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 170
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 170
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 Network Dispatcher works” on page 170

Table 13. Advanced configuration tasks for the Network Dispatcher (continued)

Task	Description	Related information
Use Server Directed Affinity API	Provides an API which allows an external agent to influence the Dispatcher affinity behavior	"Server Directed Affinity API to control client-server affinity" on page 171
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 172
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" on page 173
Use rule 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.	"Rule affinity override" on page 174
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 175
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 177
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 177
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 178
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 180
Using Cisco Consultant (additional information)	How Cisco Consultant interacts with Cisco CSS Switch and additional information on configuring weights.	"Additional information on advanced Cisco Consultant functions" on page 182

Optimizing the load balancing provided by Network Dispatcher

The manager function of Network Dispatcher performs load balancing based on the following settings:

- “Proportion of importance given to status information”
- “Weights” on page 123
- “Manager intervals” on page 124
- “Advisor intervals” on page 128
- “Advisor report timeout” on page 128
- “Sensitivity threshold” on page 125
- “Smoothing index” on page 125

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. For Cisco Consultant, the manager obtains the first two values (active connections and new connections) from the Cisco CSS Switch.

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. Since 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. Since 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 will be fairly low. If the client connections are slower (such as a database query), then the number of active connections will be higher.

You should avoid setting active and new connections proportions values too low. You will disable Network Dispatcher's 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 **ndcontrol cluster set cluster proportions** command. See "ndcontrol cluster — configure clusters" on page 234 for more information.

Weights

Note: If you are using the Cisco Consultant component, for additional information see "Cisco Consultant weights" on page 183.

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 `ndcontrol server` command. For a description of the `fixedweight` option, see “Manager fixed weights”.

Weights are applied to all servers on a port. For any particular port, the requests will be 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, enter the `ndcontrol port set weightbound` command. This command affects how much difference there can be between the number of requests each server will get. If you set the maximum weight 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 weight of 2, one server could get twice as many requests as another. At a maximum weight of 10, one server could get 10 times as many requests as another. The default maximum weight 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.

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 `ndcontrol server` command. For example:

```
ndcontrol server set cluster:port:server fixedweight yes
```

After `fixedweight` is set to yes, use the `ndcontrol 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 `ndcontrol server` command with `fixedweight` set to no. For more information, see “`ndcontrol server` — configure servers” on page 274.

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 `ndcontrol manager interval` and `ndcontrol 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:

```
ndcontrol 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:

```
ndcontrol manager refresh 3
```

This will cause the manager to wait for 3 intervals before asking the executor for status.

Sensitivity threshold

Network Dispatcher provides other methods 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:

```
ndcontrol 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:

```
ndcontrol manager smoothing 4
```

In most cases, you will not need to change this value.

Using scripts to generate an alert or record server failure

Network Dispatcher 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 `...nd/servers/samples` install directory. In order to run the files, you must move them to the `...nd/servers/bin` directory and remove the `.sample` file extension. 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.

Advisors

Advisors are agents within Network Dispatcher. 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 Network Dispatcher. (For instance, you would not use the Telnet advisor with the CBR component.) Network Dispatcher also supports the concept of a “custom advisor” that allows users to write their own advisors.

Limitation for bind-specific server applications on Linux: For Linux, Network Dispatcher does not support the use of advisors when load-balancing servers with bind-specific server applications (including other Network Dispatcher components such as Mailbox Locator or Site Selector) when they are binding to the cluster 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.

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 Network Dispatcher 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:

```
ndcontrol 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:

```
ndcontrol 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 131.)

Note: The group advisor will advise on all clusters/sites that do not currently have a cluster/site specific advisor.

Using the above 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:

```
ndcontrol advisor stop ADV_custom clusterB:80
```

- To stop the custom advisor for port 80 on *clusterB* and *clusterC*, specify just the port:

```
ndcontrol 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:

```
ndcontrol 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:

```
ndcontrol advisor timeout http 80 30
```

For more information on setting the advisor report timeout, see “ndcontrol advisor — control the advisor” on page 228.

Advisor connect timeout and receive timeout for servers

For the Network Dispatcher, you can set the advisor’s timeout values at which it detects a server 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:

```
ndcontrol advisor connecttimeout http 80 9
ndcontrol advisor receivetimeout http 80 9
```

The default for connect and receive timeout is 3 times the value specified for the advisor interval time.

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 “HTTP advisor request/response (URL) option” on page 140 for more information on how to change the type of request sent by the HTTP advisor.
- 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 **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 opens a connection, sends a CLIENT HELLO request, waits for a response, closes the connection, and returns the elapsed time as a load.

Note: 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 74, for more information
- The Caching Proxy (**ibmproxy**) 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 **ibmproxy** advisor, Caching Proxy needs to be running on all servers being load balanced. The machine on which the Network Dispatcher resides does not need to have Caching Proxy installed unless it is on the same machine it is load balancing.

- 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 **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 **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 was 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.
- The **reach** advisor pings its target machines. This advisor was 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 **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 **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 135.
- 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 Network Dispatcher. 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 WAND configuration” on page 149 for more information.
- 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”.
- The **WAS** (WebSphere Application Server) advisor works in conjunction with the WebSphere Application servers. Customizable sample files for this advisor are provided in the install directory. For more information, see “WebSphere Application Server advisor” on page 132.

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 Network Dispatcher product. After installing Network Dispatcher, you may find the sample code in **...nd/servers/samples/CustomAdvisors** install directory.

The default install directories are:

- AIX: /usr/lpp/nd
- Linux: /opt/nd
- Sun: /opt/nd
- Windows 2000: c:\Program Files\IBM\nd

WebSphere Application Server advisor

Sample custom advisor files specifically for the WebSphere Application Server advisor are provided in the Network Dispatcher install directory.

- ADV_was.java is the file to be compiled and run on the Network Dispatcher machine
- NDAdvisor.java.servlet (to be renamed NDAdvisor.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. You must get and install a Java 1.3 compiler for your machine. These files are referenced during compilation:

- the custom advisor file
- the base classes file, `ibmnd.jar`, found in the `...nd/servers/lib` directory where Network Dispatcher is installed.

Your classpath must point to both the custom advisor file and the base classes file during the compile.

For Windows 2000, a compile command might look like this:

```
javac -classpath <install_dir>\nd\servers\lib\ibmnd.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 `...nd/servers/lib/CustomAdvisors` directory where Network Dispatcher is installed.

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 2000, copy the class file (in binary) to an AIX machine, and run the custom advisor there.

For AIX, Linux, and Sun, the syntax is similar.

Run

To run the custom advisor, you must first copy the class file to the proper Network Dispatcher subdirectory:

```
.../nd/servers/lib/CustomAdvisors/ADV_fred.class
```

Configure the component, start its manager function, and issue the command to start your custom advisor:

```
ndcontrol 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

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

Network Dispatcher first looks at the list of native advisors that it provides. If it does not find a given advisor there, Network Dispatcher then looks at the customer's list of customized advisors.

Naming and path

- The custom advisor class must be located within the subdirectory of **...nd/servers/lib/CustomAdvisors/** in the Network Dispatcher base directory. The defaults for this directory vary by operating system:
 - AIX
/usr/lpp/nd/servers/lib/CustomAdvisors/
 - Linux
/opt/nd/servers/lib/CustomAdvisors/
 - Solaris

/opt/nd/servers/lib/CustomAdvisors/

- Windows 2000

Common install directory path:

C:\Program Files\IBM\edge\nd\servers\lib\CustomAdvisors

Native install directory path:

C:\Program Files\IBM\nd\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 name must be prefixed with **ADV_**.

Sample advisor

The program listing for a sample advisor is included in “Sample advisor” on page 352. After installation, this sample advisor can be found in the `...nd/servers/samples/CustomAdvisors` directory.

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. Since 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 (i.e. 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 cluster:port:server configuration. 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.

Metric Server

This feature is available for all the Network Dispatcher components.

Metric Server provides server load information to the Network Dispatcher in the form of system-specific metrics, reporting on the health of the servers. The Network Dispatcher 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 a configuration example see Figure 11 on page 39.

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 servers that Network Dispatcher is load balancing.

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 Network Dispatcher.

- Network Dispatcher manager (Network Dispatcher side)
 1. Start **ndserver**.
 2. Issue command: **ndcontrol 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: **ndcontrol 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 - **cpupload** 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. This numeric value should represent a load measurement, not an availability value.

Note: For Site Selector, cpupload and memload run automatically.

Limitation: For Windows 2000, 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 Network Dispatcher machine, create a key file for the component that is running (using **ndkeys create** command). See "Remote Authenticated Administration" on page 185, for more information on ndkeys.
- On the server machine, copy the resulting key file to the **.../nd/admin/key** directory. 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 Network Dispatcher install.
 2. Check the **metricserver** script in the **/usr/bin** directory to verify that the desired RMI port is being used. (For Windows 2000, the directory is C:\WINNT\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 Network Dispatcher machine.

3. The following two scripts are already provided for the customer: **cpupload** (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 **...nd/ms/script** directory.

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 **...nd/ms/script** directory. 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 UNIX-based platforms, scripts must begin with the shell declaration, otherwise they may not properly execute.

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 occurrence 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 2000: You will also need to alias the *OTHER_ADDRESS* on the Microsoft stack. To alias an address on the Microsoft stack, see page 157.

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

When defining a server in the Network Dispatcher configuration, you are able to distribute the load based on the health of the overall server (using Metric Server agent) and/or the health of any port-specific application (using the advisor function).

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, database requests, and so on. Network Dispatcher 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 Network Dispatcher 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.

Within the Network 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 dotted-decimal format. Or, if you configure 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 **ndcontrol server add** command. See “ndcontrol server — configure servers” on page 274 for more information.

Below 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).

Note: There is a limitation that Server Directed Affinity does not work with the server partitioning feature since SDA requires that the server addresses be unique, in the configuration, for searching capabilities. See “Server Directed Affinity API to control client-server affinity” on page 171 for more information.

HTTP advisor request/response (URL) option

The URL option for the HTTP advisor is available for the Dispatcher and CBR components.

After you have started an HTTP 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 HTTP 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 138 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 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 advisor response that the HTTP advisor scans for in the HTTP response. The HTTP 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 **ndcontrol>>** 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/2.0"  
server set cluster:port:server advisorresponse "HTTP 200 OK"
```

- When issuing the **ndcontrol** command from the operating system prompt, you must precede the text with `"\"` and follow the text with `\"`. For example:

```
ndcontrol server set cluster:port:server advisorrequest "\"head / http/2.0\""  
ndcontrol server set cluster:port:server advisorresponse "\"HTTP 200 OK\""
```

Note: After starting an HTTP advisor for a specified HTTP port number, the advisor request/response value is enabled for servers under that HTTP port.

See “ndcontrol server — configure servers” on page 274 for more information.

Using collocated servers

Network Dispatcher 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, Site Selector, Mailbox Locator, and Cisco Consultant 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 Network Dispatcher 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

Red Hat Linux v7.1 (Linux kernel version 2.4.2-2) or SuSE Linux v7.1 (Linux kernel version 2.4.0-4GB): In order to configure both collocation and high availability at the same time, when running the Dispatcher component using the mac forwarding method, you must install a Linux kernel patch. For more information on installing the patch, see “Installing the Linux kernel patch (to suppress arp responses on the loopback interface)” on page 66. However, when following these instructions, skip the step to alias the loopback adapter. You should add the ifconfig instruction to alias the loopback adapter in the goStandby high-availability script file that gets executed when a Dispatcher goes into standby state.

Solaris: There is a limitation that you cannot configure WAND advisors when the entry-point Dispatcher is collocated. See “Using remote advisors with wide area support” on page 144.

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 **ndcontrol 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.

Note: For **Windows 2000:** You can collocate Dispatcher, but do *not* use the collocated keyword. Collocation is supported when using Dispatcher’s nat and cbr forwarding methods, but it is not supported when using Dispatcher’s mac forwarding method. For more information on Dispatcher’s forwarding methods see “Dispatcher’s NAT/NAPT (nat forwarding method)” on page 47, “Dispatcher’s content-based routing (cbr forwarding method)” on page 49, and “Dispatcher’s MAC-level routing (mac forwarding method)” on page 47.

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 **ndcontrol executor set nfa IP_address** command. And, add the server using the NFA address with the **ndcontrol 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:
ndcontrol server add cluster:port:server collocated yes.

See “ndcontrol server — configure servers” on page 274, for more information on ndcontrol server command syntax.

For the CBR component

CBR supports collocation on all platforms with no additional configurations required. However, the Web servers and Caching Proxy that you use must be bind-specific.

For the Mailbox Locator component

Mailbox Locator supports collocation on all platforms. However, the server must be bound to a different address than Network Dispatcher for this to work. In order to collocate a POP3 or IMAP server on the same machine it must be bound to an IP address which is different from the cluster address. This can be achieved through the use of the loopback address.

For the Site Selector component

Site Selector supports collocation on all platforms with no additional configurations required.

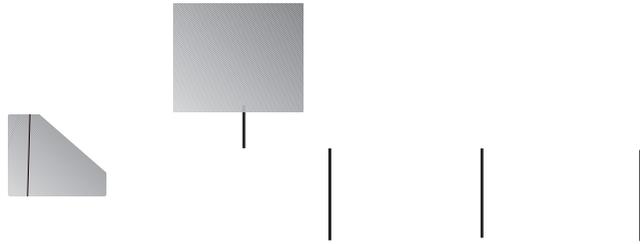
For the Cisco Consultant component

Cisco Consultant supports collocation on all platforms with no additional configurations required.

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 22 on page 143). A client’s packet comes into the ND machine and is sent to the server, and then from the server directly back to the client.



The Wide Area Dispatcher enhancement adds support for offsite servers, known as *remote servers* (see Figure 23). If GRE is not supported at the remote site and you are not using Dispatcher's nat forwarding method, then the remote site must consist of a remote Dispatcher machine (Dispatcher 2) and its locally attached servers (ServerG, ServerH, and ServerI). All the Dispatcher machines must be on the same operating system. A client's packet can now go from the Internet to a Dispatcher machine, from there to a geographically remote Dispatcher machine to one of its locally attached 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

Wide area commands are not complex. 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 **ndcontrol 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 24 on page 146, if you are adding *ND 2* as a remote server under *ND 1*, you must define *router 1* as the router address. General syntax:

```
ndcontrol server add cluster:port:server router address
```

For more information on the router keyword, see “ndcontrol server — configure servers” on page 274.

2. Configure aliases. On the first Dispatcher machine (where the client request arrives from the Internet), the cluster address must be aliased using **cluster configure**, **ifconfig** or **ndconfig**, as before. On the remote Dispatcher machines, however, the cluster address is **not** aliased to a network interface card.

Using remote advisors with wide area support

On entry-point Dispatchers, advisors will work correctly without any special configuration for most platforms.

Linux: There is a limitation on using remote advisors with wide area support configurations. Protocol-specific advisors, such as the HTTP advisor, that are running on the entry-point Dispatcher machine will not correctly assess the status of server machines at the remote site. To alleviate this problem, do one of the following:

- Run the protocol-independent ping advisor on the entry-point Dispatcher machine.
- Run a protocol-specific advisor on the entry-point Dispatcher machine along with a matching protocol-specific server daemon (such as a Web server) on the remote Dispatcher machine.

Either of these options will provide the advisor running on the entry point Dispatcher machine with an assessment of the status of the remote Dispatcher machine.

Solaris: On entry-point Network Dispatchers, you must use the arp configuration method (instead of the ifconfig or cluster configuration methods). For example:

```
arp -s <my_cluster_address> <my_mac_address> pub
```

Note: There are Solaris limitations as follows:

- WAND advisors work only with the arp method of cluster configuring.
- Advisors for bind-specific servers work only with the arp method of cluster configuring.
- Collocation works only with the ifconfig method of cluster configuring.

On remote Dispatchers, you will need to perform the following configuration steps for each remote cluster address. For a high-availability configuration at the remote Network Dispatcher location, you must perform these steps on both machines.

AIX

- Alias the cluster address to the loopback adapter. The netmask value must be set to 255.255.255.255. For example:

```
ifconfig lo0 alias 9.67.34.123 netmask 255.255.255.255
```

Note: Advisors running on both the local and remote Dispatcher machines are necessary.

Linux

- Alias the cluster address to the loopback adapter. For example:

```
ifconfig lo:1 9.67.34.123 netmask 255.255.255.255 up
```

Note: Advisors running on both the local and remote Dispatcher machines are necessary.

Solaris

- No additional configuration steps are required.

Windows 2000

1. The Dispatcher requires two IP addresses. One for the Microsoft TCP/IP stack and one for the Network Dispatcher stack. Configure the NFA using the IP address of the Network Dispatcher stack. For example:

```
ndconfig en0 alias 9.55.30.45 netmask 255.255.240.0
```

2. Configure the loopback adapter with the remote cluster address as an alias. The netmask value must be set to 255.255.255.255. For example:

```
ndconfig lo0 alias 9.67.34.123 netmask 255.255.255.255
```

3. Delete any entries in the arp table for the remote cluster address.

- a. To view the contents of the arp table, enter:

```
arp -a
```

b. To delete an entry if one exists, enter:

```
arp -d 9.67.34.123
```

Note: To determine the MAC address of your interface, enter:

1) `ping your_hostname`

2) `arp -a`

and look for the address of your machine.

4. Add a route to the remote cluster (9.67.34.123) using the NFA (IP address of the Network Dispatcher stack). The mask value must be set to 255.255.255.255. For example:

```
route add 9.67.34.123 mask 255.255.255.255 9.55.30.45
```

Configuration example

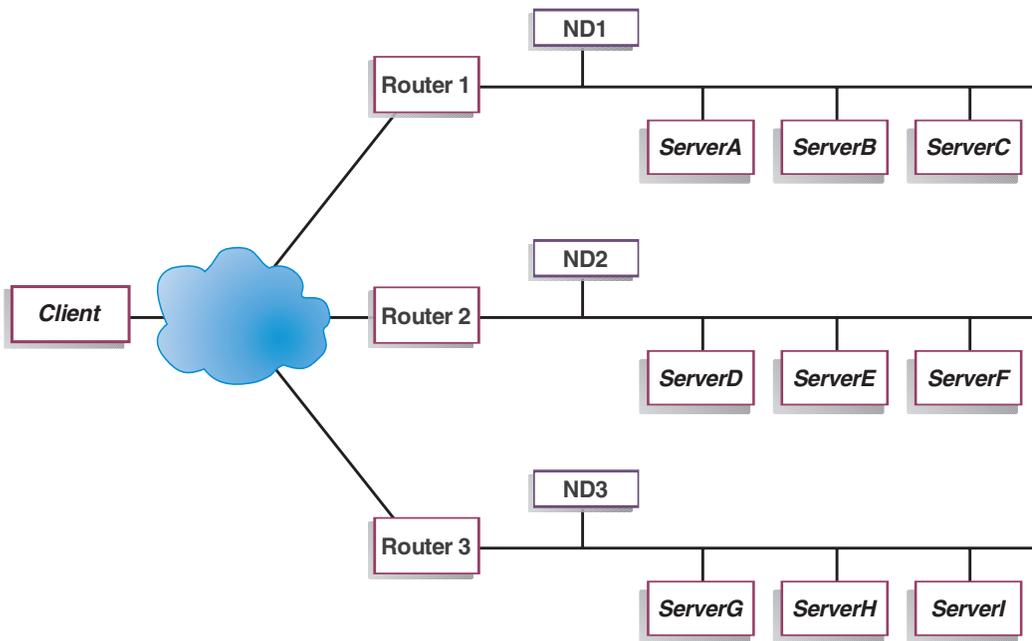


Figure 24. Wide area example configuration with remote Network Dispatchers

This example applies to the configuration illustrated in Figure 24.

Here is how to configure the Dispatcher machines to support cluster address xebec on port 80. ND1 is defined as the “entry point,”. An ethernet connection is assumed. Note that ND1 has five servers defined: three local (ServerA, ServerB, ServerC) and two remote (ND2 and ND3). Remotes ND2 and ND3 each have three local servers defined.

At the console of the first Dispatcher (ND1), do the following:

1. Start the executor.
ndcontrol executor start
2. Set the nonforwarding address of the Dispatcher machine.
ndcontrol executor set nfa ND1
3. Define the cluster.
ndcontrol cluster add xebec
4. Define the port.
ndcontrol port add xebec:80
5. Define the servers.
 - a. **ndcontrol server add xebec:80:ServerA**
 - b. **ndcontrol server add xebec:80:ServerB**
 - c. **ndcontrol server add xebec:80:ServerC**
 - d. **ndcontrol server add xebec:80:ND2 router Router1**
 - e. **ndcontrol server add xebec:80:ND3 router Router1**
6. If using Windows 2000, configure the NFA of the Dispatcher LAN adapter.
ndcontrol cluster configure ND1 and also configure xebec as clusteraddr.
7. Configure the cluster address.
ndcontrol cluster configure xebec

At the console of the second Dispatcher (ND2):

1. Start the executor.
ndcontrol executor start
2. Set the nonforwarding address of the Dispatcher machine.
ndcontrol executor set nfa ND2
3. Define the cluster.
ndcontrol cluster add xebec
4. Define the port.
ndcontrol port add xebec:80
5. Define the servers.
 - a. **ndcontrol server add xebec:80:ServerD**
 - b. **ndcontrol server add xebec:80:ServerE**
 - c. **ndcontrol server add xebec:80:ServerF**
6. If using Windows 2000, configure the NFA of the Dispatcher LAN adapter.
ndcontrol cluster configure ND2

At the console of the third Dispatcher (ND3):

1. Start the executor.

ndcontrol executor start

2.

the remote servers support the encapsulated GRE packets. Network Dispatcher encapsulates WAND packets with the GRE key field set to decimal value 3735928559.

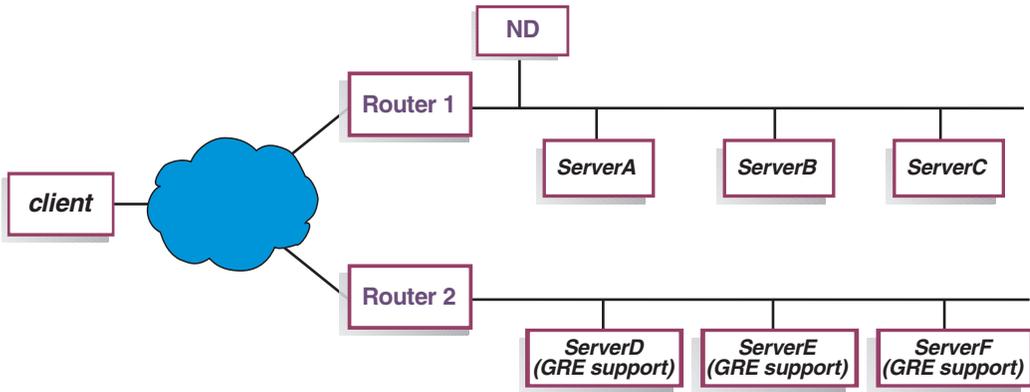


Figure 25. Wide area example configuration with server platform that supports GRE

For this example (Figure 25), to add remote ServerD, which supports GRE, define it within your Network Dispatcher configuration as if you are defining a WAND server in the `cluster:port:server` hierarchy:

```
ndcontrol server add cluster:port:ServerD router Router1
```

Using Self Advisor in a two-tiered WAND configuration

The self advisor is available on the Dispatcher component.

For Network Dispatcher in a two-tiered WAND (wide area Network Dispatcher) configuration, Dispatcher provides a *self* advisor that collects load status information on backend servers.

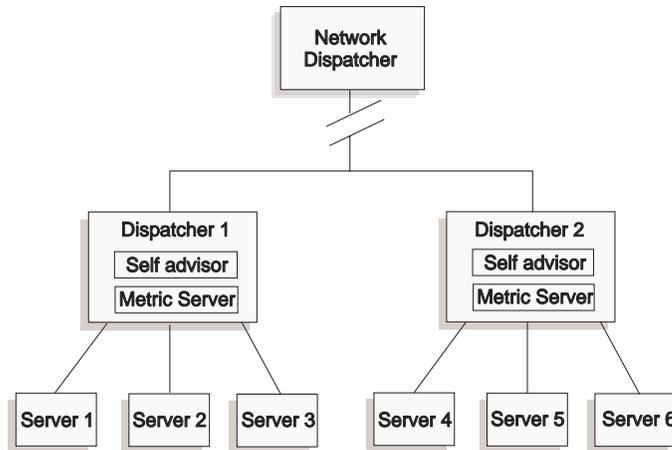


Figure 26. Example of a two-tiered WAND configuration using the self advisor

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 Network Dispatcher. 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 `ndloadstat` file. Network Dispatcher also provides an external metric called `ndload`. The Metric Server agent on each Dispatcher machine runs its configuration that calls the external metric `ndload`. The `ndload` script extracts a string from the `ndloadstat` 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 Network Dispatcher for use in determining which Dispatcher to forward client requests.

The `ndload` executable resides in the `.../nd/ms/script` directory for Network Dispatcher.

High availability

The high availability feature is only available for the Dispatcher 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 be using the same operating system.

- 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, it is recommended that at least one of the heartbeat pairs 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 153.
- 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 **ndcontrol highavailability** is in “ndcontrol highavailability — control high availability” on page 248.

For a more complete discussion of many of the tasks below, see “Setting up the Dispatcher machine” on page 56.

1. Start the server on both Dispatcher server machines.
2. Start the executor on both machines.
3. 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.

Windows 2000 only: In addition, configure each nonforwarding address using the **ndconfig** command. For example:

```
ndconfig en0 nfa_addr netmask netmask
```

4. Set up the cluster, port, and server is3(c. TD(2.)2ws)-333(2000)-33361 73tti59(esdvar

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:

```
ndcontrol cluster set clusterA primaryhost NFAdispatcher1
ndcontrol cluster set clusterB primaryhost NFAdispatcher2
```

- For Dispatcher 2 issue:

```
ndcontrol cluster set clusterB primaryhost NFAdispatcher2
ndcontrol cluster set clusterA primaryhost NFAdispatcher1
```

5. Start the manager and advisors on both machines. The reach advisor starts automatically by the manager function.
6. Create alias script files on each of the 2 Dispatcher machines. See “Using scripts” on page 155.
7. Add the heartbeat information on both machines:

```
ndcontrol highavailability heartbeat add sourceaddress destinationaddress
```

Note: *Sourceaddress* and *destinationaddress* are the IP addresses (either DNSnames or dotted-decimal addresses) of the Dispatcher machines. The values will be reversed on each machine. 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, it is recommended that at least one of the heartbeat pairs 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.

8. 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. Example:

```
ndcontrol 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 153, for more information.

9. Add the backup information to each machine:
 - For the **primary** machine:

```
ndcontrol highavailability backup add primary [auto | manual] port
```
 - For the **backup** machine:

```
ndcontrol highavailability backup add backup [auto | manual] port
```

- For mutual high availability each Dispatcher machine has **both** primary and backup roles:

```
ndcontrol highavailability backup add both [auto | manual] port
```

Note: Select an unused port on your machines as the *port*. Your two machines will communicate over this port.

10. Check the high availability status on each machine:

```
ndcontrol 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.

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, it is recommended that you enter all `ndcontrol` 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, when configuring high availability and collocation at same time using the Dispatcher component's MAC port forwarding method, you must install a Linux kernel patch. For more information on installing the patch, see "Installing the Linux kernel patch (to suppress arp responses on the loopback interface)" on page 66.

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.

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. Switchover 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, you must also start the *reach advisor*. The reach advisor starts automatically by the manager function. For more information on the reach advisor, see page 130.

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).
- 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.

Since the Dispatcher machines will change states when a failure is detected, the commands above must be issued automatically. Dispatcher will execute user-created scripts to do that. Sample scripts can be found in the **...nd/servers/samples** directory and *must* be moved to the **...nd/servers/bin** directory in order to run.

Note: For a mutual high availability configuration, each “go” script will be called by the Dispatcher with a parameter identifying the primary Dispatcher address. The script must query this parameter and perform the **ifconfig** commands (or **ndconfig** commands, if Windows 2000) for those cluster addresses associated with that primary Dispatcher.

The following sample scripts may be used:

goActive

The goActive script executes 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 executes 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 executes when a Dispatcher executor is stopped and before it is started for the first time.

- 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 standalone 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 goldle script executes 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 **ndcontrol cluster configure** command or manually configure the aliases each time the executor is started.

highavailChange

The highavailChange script executes 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.

Note: For Windows 2000: 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.

Configure rules-based load balancing

You can use rules-based load balancing to fine tune when and why packets are sent to which servers. Network Dispatcher 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 falls through the other higher priority rules. This can be a "Sorry, the site is currently down, please 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.

Note: Configuration using rules does *not* apply to Mailbox Locator (which proxies IMAP or POP3 requests to specific servers based on userID and password) or to Cisco Consultant (which uses the manager and advisors functions to provide load balancing information to the Cisco CSS Switch).

You can choose from the following types of rules:

- For Dispatcher:

- Client IP address
- Time of day
- Connections per second for a port
- Active connections total for a port
- Client port
- Type of service (TOS)
- Reserved bandwidth
- Shared bandwidth
- Always true
- Content of a request
- For CBR:
 - Client IP address
 - Time of day
 - Connections per second for a port
 - Active connections total for a port
 - Always true
 - Content of a request
- For Site Selector:
 - Client IP address
 - Time of day
 - Metric all
 - Metric average
 - Always true

We recommend that you 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 C. Content rule (pattern) syntax” on page 285.)

Rules are evaluated in priority order. In other words, a rule with a priority of 1 (lower number) will be evaluated before a rule with a priority of 2 (higher number). The first rule that is satisfied will be used. Once 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 of them must be available 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 have noticed 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 **ndcontrol rule** command, for example:

```
ndcontrol rule add 9.67.131.153:80:ni type ip
  beginrange 9.0.0.0 endrange 9.255.255.255
```

This "ni" rule would screen out any connection from IBM clients. You would then add to the rule the servers which you wanted accessible to IBMers, or if you do not add any servers to the rule, requests coming from 9.x.x.x addresses would not be served by any of your servers.

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 servers to HTTP during full-time, then adding another five 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 would set up a rule that excludes those servers during the necessary maintenance period.

Using rules based on the connections per second on a port

This rule type is available in the Dispatcher and CBR component.

Note: The manager must be running for the following to work.

You may want to use rules based on connections per second on a port 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 > 100 then use these 2 servers
2. If connections per second on port 80 > 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 increase above a certain level. This way, Dispatcher would balance the load across all five servers at peak times.

Using rules based on the active connections total on a port

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 **ndcontrol rule** command or the **cbrcontrol rule** command, for example:

```
ndcontrol 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 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 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 **ndcontrol 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:

```
ndcontrol rule add 9.67.131.153:80:tsr type service tos 0xx1010x
```

Using rules based on reserved bandwidth and shared bandwidth

Capacity utilization and bandwidth rules are only available in the Dispatcher component.

Using the capacity utilization feature, Dispatcher measures the amount of data delivered by each of its servers. Dispatcher tracks capacity at the server, rule, port, cluster, and executor levels. For each of these levels, there is a new byte counter value: kilobytes transferred per second. The rate value (kilobytes transferred per second) is calculated over a 60 second interval. You can view these capacity values from the GUI or from the output of a command line report.

Dispatcher allows you to allocate a specified bandwidth to sets of servers within your configuration using the *reserved bandwidth* rule. When traffic exceeds the reserved bandwidth threshold, you can do either of the following:

- Send the traffic to another server, using an always true rule, that responds with a “site busy” type response
- Or, share a specified amount of bandwidth at the cluster level or executor level using the *shared bandwidth* rule. And, when the overall shared

bandwidth threshold is approached, you can then direct traffic to another server, using an always true rule, that responds with a “site busy” type response.

By using the shared bandwidth rule in conjunction with the reserved bandwidth rule, as described above, you can provide preferred clients with increased server access and therefore optimize performance for their transactions. For example, using the shared bandwidth rule to recruit unused bandwidth, you can allow online trading customers executing trades on server clusters to receive greater access than customers using other server clusters for investment research.

Note the following to determine whether bandwidth rules can help you manage the volume of response traffic that flows from servers to clients:

- Bandwidth rules can help to manage the volume of response traffic that flows from a set of server machines, based upon the client requests, that flow through Network Dispatcher. If some client traffic goes directly to the server machines and is unseen by Network Dispatcher, then results may be unpredictable.
- Bandwidth rules can help to manage the volume of response traffic flowing on a link from a set of server machines to the network when all servers use the same link to the network. If servers use different links, or multiple links, to get to the network, then results for each individual link may be unpredictable.
- Bandwidth rules are helpful only when all servers are on the same local network as the Network Dispatcher machine. If some servers are remote, having different paths to the network, then results may be unpredictable.

Reserved bandwidth rule

This rule type is only available in the Dispatcher component.

The reserved bandwidth rule allows you to load balance based on 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:

```
ndcontrol rule add 9.67.131.153:80:rbw type reservedbandwidth
  beginrange 0 endrange 300
```

The begin range and endrange are specified in kilobytes per second.

Shared bandwidth rule

This rule type is only available in the Dispatcher component.

If the amount of data transferred exceeds the limit for the reserved bandwidth rule, the shared bandwidth rule provides you the ability to recruit unused bandwidth available at the site. You can configure this rule to share bandwidth at either the cluster or the executor level. Sharing bandwidth at the cluster level allows a port (or ports) to share a maximum amount of bandwidth across several ports (applications/ protocols) 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.

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 **ndcontrol executor** or **ndcontrol cluster** command with the `sharedbandwidth` option. The following are examples of the command syntax:

```
ndcontrol executor set sharedbandwidth size
ndcontrol 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. You should specify a maximum `sharedbandwidth` value that does not exceed the total bandwidth (total server capacity) available.

The following are examples of adding or setting a `sharedbandwidth` rule:

```
ndcontrol rule add 9.20.30.4:80:shbw type sharedbandwidth sharelevel value
ndcontrol 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.

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 fire.

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 fire.

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 will be 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 **ndcontrol rule** command like:

```
ndcontrol 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 executed by changing their priority levels.

Using rules based on the request content

This rule type is available in the Dispatcher or CBR component.

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 C. Content rule (pattern) syntax” on page 285.

Adding rules to your configuration

You can add rules using the **ndcontrol 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. She knows which IP addresses are given to each division. She would create the first set of rules based on client IP address to separate each division’s load:

```
ndcontrol rule add 130.40.52.153:80:div1 type ip b 9.1.0.0 e 9.1.255.255
ndcontrol rule add 130.40.52.153:80:div2 type ip b 9.2.0.0 e 9.2.255.255
ndcontrol rule add 130.40.52.153:80:div3 type ip b 9.3.0.0 e 9.3.255.255
```

Next, she would 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:

```
ndcontrol rule useserver 130.40.52.153:80:div1 207.72.33.45
ndcontrol rule useserver 130.40.52.153:80:div2 207.72.33.63
ndcontrol 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 **ndcontrol 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 Network Dispatcher, you could only measure each rule's condition across all servers on the port.)

Note: 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.

The following are examples of adding or setting the evaluate option on a reserved bandwidth rule:

```
ndcontrol rule add 9.22.21.3:80:rbweval type reservedbandwidth evaluate level
ndcontrol rule set 9.22.21.3:80:rbweval evaluate level
```

The evaluate *level* can be set to either port or rule. 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 will be 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.

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, it is recommended that you 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, 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 2000: Execute the following command:

```
ndconfig en1 10.0.0.x netmask 255.255.255.0
```

Where `en1` is the name of the second interface card in the Dispatcher machine, `10.0.0.x` is the network address of the second interface card, and `255.255.255.0` is the netmask of the private network.

The servers added using the **ndcontrol server add** command must be added using the private network addresses; for example, referring to the Apple server example in Figure 27 on page 168, the command should be coded as:

```
ndcontrol server add cluster_address:80:10.0.0.1
```

not

```
ndcontrol 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.

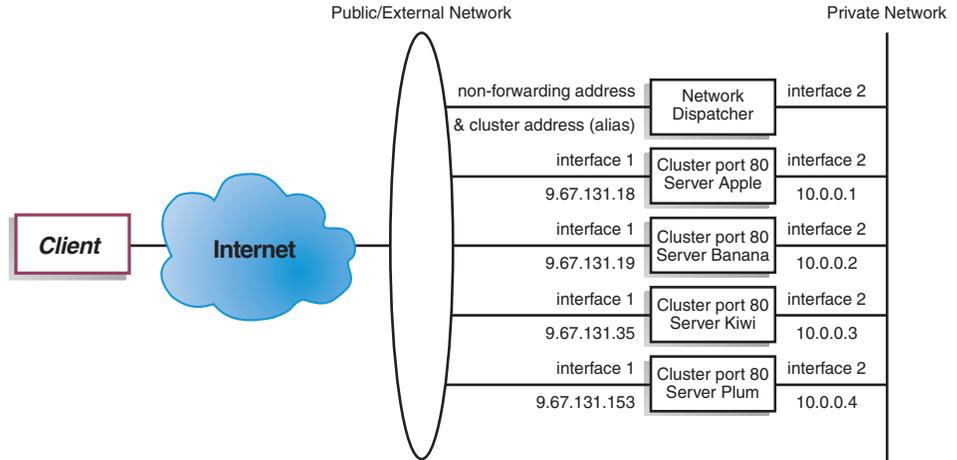


Figure 27. Example of a private network using Dispatcher

Using a private network configuration only applies to the Dispatcher component.

Use wildcard cluster to combine server configurations

The “wildcard” refers to the cluster’s ability to match multiple IP addresses (i.e. 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 star 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 `ndcontrol 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 will be 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 will be 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.

Using wildcard cluster to combine server configurations only applies to the Dispatcher component.

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.

Once Dispatcher has been configured as a default route, any TCP or UDP traffic through the Dispatcher machine will be load balanced using the wildcard cluster configuration.

One application of this is to load balance firewalls. Since 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. Since 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 box 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. 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 will be delivered to the Caching Proxy server running on the Dispatcher workstation. The client request will then be proxied as usual, and the response will be 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 will be discarded rather than delivered back to the operating system. Port number 0 (zero) is used to specify a wildcard port, for example:

```
ndcontrol port add cluster:0
```

Note: The wildcard port cannot be used to handle FTP traffic.

How affinity feature for Network Dispatcher works

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 "port stickytime" to some number of seconds. The feature is disabled by setting stickytime to zero.

Interaction with cross port affinity: 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 172, for more information.

Behavior when disable affinity

With the feature disabled, whenever a new TCP connection is received from a client, Dispatcher 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, Dispatcher treats it as an unrelated new connection, and again picks the right server at that moment in time.

Behavior when enable affinity

With the 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.

Server Directed Affinity API to control client-server affinity

The Server Directed Affinity API only applies to the Dispatcher component.

The SDA feature provides an API that allows an external agent to influence the Dispatcher affinity behavior.

Note: There is a limitation that Server Directed Affinity does not work with the server partitioning feature since SDA requires that the server addresses be unique, in the configuration, for searching capabilities. SDA also does not work with the SSL ID Affinity feature because, with SDA, servers control the affinity table.

SDA Features

Your application may have indicated that their server systems have the knowledge to direct client requests to particular server machines better than Dispatcher can. Rather than having a client be "directed" to the same server as selected by Dispatcher's load balancing selection, you may want the client to be "directed" to the server of your choosing. The SDA feature provides this API. You can now write your own software to implement an SDA agent, which communicates with a listener in Dispatcher. It can then manipulate the Dispatcher affinity tables to:

- query the contents
- insert new records
- remove records

Records inserted in an affinity table by an SDA agent remain in the table indefinitely. They do not timeout. They are removed only when the SDA agent removes them or if a Dispatcher advisor detects that the server is dead.

Dispatcher's SDA components

Dispatcher implements a new socket listener to accept and handle requests from an SDA agent. When an SDA agent opens a connection with Dispatcher, the listener will accept it and leave the connection open. Multiple requests and responses can flow over this persistent connection. The socket will close when the SDA agent closes it or if Dispatcher detects an unrecoverable error. Inside Dispatcher, the listener takes each request from the SDA agent, communicates with the appropriate affinity table in the Dispatcher executor kernel, and prepares a response for the SDA agent.

For more information, refer to the files appearing in the Network Dispatcher's install directory:

- the API: `...nd/servers/samples/SDA/SDA_API.htm`
- the sample code for an SDA agent:
`...nd/servers/samples/SDA/SDA_SampleAgent.java`.

Cross port affinity

Cross port affinity only applies to the Dispatcher component.

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:

```
ndcontrol port set cluster:20 crossport 10
ndcontrol port set cluster:30 crossport 10
ndcontrol 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 “`ndcontrol port — configure ports`” on page 261, for detailed information on command syntax for the **crossport** option.

Affinity address mask

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 **ndcontrol port** command allows you to mask the common high-order bits of the 32-bit IP address. If this feature is enabled, 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.

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 Network Dispatcher. 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 172, for more information.

To enable affinity address mask, issue an `ndcontrol port` command similar to the following:

```
ndcontrol port set cluster:port 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 “`ndcontrol port — configure ports`” on page 261, for detailed information on command syntax for `stickymask` (affinity address mask feature).

Rule affinity override

With rule 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 don’t want the client to be sticky to that server. With rule 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 “`ndcontrol server — configure servers`” on page 274, for detailed information on command syntax for the rule affinity override, using the server `sticky` option.

Quiesce handling for sticky connections

Quiesce handling for sticky connections applies to the Dispatcher and CBR components.

To remove a server from the Network Dispatcher configuration for any reason (updates, upgrades, service, and so forth), you can use the `ndcontrol 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.

Only use quiesce “now” 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:

```
ndcontrol 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 Network Dispatcher.

Affinity option on the rule

You can specify the following types of affinity on the **ndcontrol rule** command:

- Active cookie — enables load-balancing Web traffic with affinity to the same server based upon cookies generated by Network Dispatcher.
- 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.
- URI — enables load-balancing Web traffic to caching-proxy servers in a manner that effectively increases the size of the cache.

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.

The active cookie affinity only applies to the CBR component. The passive cookie and URI affinity apply to the CBR component and to Dispatcher component’s cbr forwarding method.

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 “ndcontrol rule — configure rules” on page 267, for detailed information on command syntax.

Once a rule has been enabled for active cookie affinity, new client requests will be load-balanced using standard CBR algorithms, while succeeding requests from the same client will be sent to the initially chosen server. The chosen server is s 1 Tf3(be)-333ttenabsts

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 client's browser. The cookie contains the cluster:port 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. 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 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 greater than zero, and the expires timestamp is greater than now, then the server in the cookie is chosen to load balance to. If any of the preceding three conditions are not met, a server is chosen using the normal algorithm. Once a server has been chosen (using either of the two methods) a new cookie is constructed containing IBM CBR, cluster:port:server_chosen information, and a timestamp. The timestamp will be the time that affinity expires. The "cluster:port: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 two hours after the expires timestamp. This is so the client's cookie database isn't cluttered up.

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.

The active cookie affinity option, for the rule command, can only be set to activecookie if port stickytime is zero (not enabled). Once 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.

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 49 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. Once 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. Once the rule fires, if passive cookie affinity is enabled, Network Dispatcher will choose the server based on the cookie name in the HTTP header of the client request. Network Dispatcher will send new incoming requests to servers based on cookies that have been generated by servers during previous connections. If the cookie value in the client request is not found or does not match any of the servers' cookie values, the server will be chosen using 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 49.) This step is omitted for the CBR component.
- Set the **affinity** parameter to "passivecookie" on the **ndcontrol rule [add|set]** command. Also, the **cookieName** parameter must be set to the name of the cookie that Network Dispatcher should look for in the client HTTP header request.
- Set the **cookievalue** parameter, for each server in the rule's server set, on the **ndcontrol 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). Once 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 49 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 size of your site's cache by eliminating redundant caching of content on multiple machines. Configure URI affinity at the rule level. Once the rule fires, if URI affinity is enabled and the same set of servers are up and responding, then Network Dispatcher will forward new incoming client requests with the same URI to the same server.

Typically, Network Dispatcher can distribute requests to multiple servers that serve identical content. When using Network Dispatcher 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 Network Dispatcher distributed the request only to the caching server with that content.

With URI affinity, Network Dispatcher 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 will be 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 49.) This step is omitted for the CBR component.
- Set the **affinity** parameter to "uri" on the **ndcontrol 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). Once URI affinity is active on a rule then you cannot enable stickytime on the port.

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 via 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.

Network Dispatcher provides user exits that trigger scripts which you can customize that alert the Administrator to a possible denial of service attack. Dispatcher provides the following sample script files in the `...nd/servers/samples` directory:

- `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 `...nd/servers/bin` directory and remove the `".sample"` file extension.

To implement the DoS attack detection, set the `maxhalfopen` parameter on the `ndcontrol port` command as follows:

```
ndcontrol 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 (`ndcontrol 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% larger than the largest number of half-open connections that your site experiences.

In addition to statistical data reported, the `halfopenaddressreport` will also generate entries in the log (`./nd/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 connections.

Note: There is an SNMP trap corresponding to the `halfOpenAlert` and `halfOpenAlertDone` scripts. If the SNMP subagent is configured and running, the corresponding traps will be 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 190.

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 168 and “Use wildcard port to direct unconfigured port traffic” on page 170.

Using binary logging to analyze server statistics

Note: The binary logging feature does not apply to the Site Selector 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 **ndcontrol log** command set to configure binary logging.

- log start
- log stop
- log set interval <second>
- log set retention <hours>
- log 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 will be 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 will be 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 will be 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 `...nd/servers/samples/BinaryLog` directory. 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:

```
ndlogreport 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**. For Mailbox Locator, use **mllogreport**. For Cisco Consultant use **lbcllogreport**.)

Additional information on advanced Cisco Consultant functions

In Cisco Consultant, the Cisco CSS Switch performs the tasks done by the executor in the Dispatcher component. Along with the current weight for each server and some other information required for its calculations, the manager gets the active and new connections values from the Cisco CSS Switch. These values are based on information that is generated and stored internally in the Cisco CSS Switch.

Cisco Consultant queries the Cisco CSS Switch management information base (MIB) to obtain active and new connection information and receives the following:

- **For active connections**, Cisco Consultant gets `apSvcConnections` from the `svcExtMIB`. This variable is indexed by `serviceName` and maps directly to active connections as recorded in the manager. Following is the `apSvcConnections` MIB entry.

```
apSvcConnections OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "The current number of TCP connections to this service"
DEFVAL { 0 }
--DEFAULT ap-display-name Service Connections
::= {apSvcEntry 20}
```

The `apSvcConnections` object identifier is:

```
1.3.6.1.4.1.2467.1.15.2.1.20
```

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 provided by the load-balanced server machines. If the client connections are fast (such as small Web pages served using HTTP GET), the number of active connections is fairly low. If the client connections are slower (such as a database query), then the number of active connections is higher.

- **For new connections**, Cisco Consultant sets the `apCntsvcHits` MIB variable in the Cisco CSS Switch `cntSvcExtMib`. For each service, Cisco Consultant:
 - Computes the sum of all `apCntsvcHits` that have that service in the index
 - Keeps a record of the total `apCntsvcHits` received
 - Calculates the delta

The index for this variable is:

```
INDEX { apCntsvcOwnName, apCntsvcCntName, apCntsvcSvcName }
```

Following is the MIB entry.

```

apCntsvcHits OBJECT-TYPE
SYNTAX Integer32
MAX-ACCESS read-only
STATUS current
DESCRIPTION
    "Total number of flows placed onto this service for this content rule."
DEFVAL { 0 }
--DEFAULT ap-display-name Hits
--DEFAULT apjam-popup-ref apCntSvcInst, Statistics
--DEFAULT apjam-chart-def cntSvcHitsChart, pie, apCntInst, "Hit Information Per Service:"
--DEFAULT apjam-chart-item cntSvcHitsChart, getnext, apCntsvcSvcName
 ::= {apSvcEntry 20}

```

The apCntsvcHits object identifier is:

1.3.6.1.4.1.2467.1.18.2.1.4

Cisco Consultant weights

The Cisco CSS Switch must be configured to use weighted round-robin load balancing. Refer to "Configuring Weight" in the *Content Services Switch Basic Configuration Guide* for information on how to do this.

Weights are set by the manager function based on internal counters in the Cisco CSS Switch and feedback from the advisors and Metric Server. If you want to set weights manually while running the manager, specify the **fixedweight** option on the **lbcontrol server** command

If all servers are down, all weights are zero. In a case such as this, when no servers are processing requests because all weights are zero, weights are set to one-half the weightbound to allow an equal chance of request processing from any capable server. The monitor shows the true weight values of zero; however, Cisco Consultant displays a weight of one-half weightbound in all other places.

Weights are sent to the Cisco CSS Switch using SNMP. Cisco Consultant sets apSvcWeight in svcExt.mib. Following is the apSvcWeight entry.

```

apSvcWeight OBJECT-TYPE
SYNTAX Integer 32(1..10)
MAX-ACCESS read-create
STATUS current
DESCRIPTION
    "The service weight which is used in conjunction with load metrics when
    making load allocation decisions. The weight may be used to bias flows
    towards the specified service."
DEFVAL { 1 }
--DEFAULT ap-display-name Service Weight
--DEFAULT apjam-popup-ref apServicesGroupInst, Properties, Advanced
--DEFAULT apjam-wizard-field 2, normal
 ::= {apSvcEntry 16}

```

The apSvcWeight object identifier is:

1.3.6.1.4.1.2467.1.15.2.1.12

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 **lbcontrol port set weightbound** command. This command specifies the differences in the number of requests each server gets. If you set the maximum weight to 1, all the servers can have a weight of 1, 0 if suspended, or -1 if marked down. As you increase this number, the difference in how servers are weighted is increased. At a maximum weight of 2, one server can get twice as many requests as another.

When a server is offline...

If an advisor learns that a server is offline, it tells the manager, and the manager sets the weight for the server to zero. When the weight of a server is greater than zero, the weight is sent to the Cisco CSS Switch, and the server becomes active; however, if the server weight is less than or equal to zero, the server is suspended. Activating and suspending a service is accomplished by setting the apSvcEnable MIB variable in the Cisco CSS Switch svcExt.mib. Following is the apSvcEnable MIB entry.

```
apSvcEnable OBJECT-TYPE
SYNTAX Integer
        disable(0)
        enable(1)
MAX-ACCESS read-create
STATUS current
DESCRIPTION
    "The state of the service, either enabled or disabled."
DEFVAL { disable }
--DEFAULT ap-display-name Status
--DEFAULT apjam-popup-ref apServicesGroupInst, Properties
--DEFAULT apjam-wizard-field 2, normal
 ::= {apSvcEntry 12}
```

The apSvcEnable object identifier is:

1.3.6.1.4.1.2467.1.15.2.1.16

Chapter 15. Operating and managing Network Dispatcher

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 "ndcontrol" and "ndserver" with the following:

- For CBR, use **cbrcontrol** and **cbrserver**
- For Mailbox Locator, use **mlcontrol** and **mlserver**
- For Site Selector, use **cbrcontrol** and **ssserver**
- For Cisco Consultant, use **lbcontrol** and **lbserver**

This chapter explains how to operate and manage Network Dispatcher and includes the following sections:

- "Remote Authenticated Administration"
- "Using Network Dispatcher logs" on page 187
- "Using the Dispatcher component" on page 188
 - "Using Simple Network Management Protocol with the Dispatcher component" on page 190
- "Using the Content Based Routing component" on page 195
- "Using the Mailbox Locator component" on page 196
- "Using the Site Selector component" on page 197
- "Using the Cisco Consultant component" on page 197

Remote Authenticated Administration

Network Dispatcher provides an option to run its configuration programs on a machine other than the one running the Network Dispatcher servers.

Communication between the configuration programs (ndcontrol, cbrcontrol, mlcontrol, sscontrol, lbcontrol, ndwizard, cbrwizard, mlwizard, sswizard, ndadmin) is performed using Java Remote Method Invocation (RMI) calls. The command to connect to a Network Dispatcher machine for remote administration is **ndcontrol 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 will be 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:

ndkeys [create | delete]

This command runs only on the same machine as the Network Dispatcher.

Using the **create** option creates a public key in the servers key directory (**...nd/servers/key/**) and creates private keys in the administration keys directory (**...nd/admin/keys/**) for each of the Network Dispatcher components. The file name for the private key is: *component-ServerAddress-RMIport*. These private keys must then be transported to the remote clients and placed in the administration keys directory.

For a Network Dispatcher machine with hostname address 10.0.0.25 using the default RMI port for each component, the **ndkeys create** command generates the following files:

- The public key: **.../nd/servers/key/authorization.key**
- The private keys:
 - **.../nd/admin/keys/dispatcher-10.0.0.25-10099.key**
 - **.../nd/admin/keys/cbr-10.0.0.25-11099.key**
 - **.../nd/admin/keys/m1-10.0.0.25-13099.key**
 - **.../nd/admin/keys/ss-10.0.0.25-12099.key**
 - **.../nd/admin/keys/lbc-10.0.0.25-14099.key**

The administration fileset has been installed on another machine. The private key files must be placed in **.../nd/admin/keys** directory on the remote client machine.

The remote client will now be authorized to configure Network Dispatcher on 10.0.0.25.

These same keys must be used on all remote clients that you want to authorize to configure Network Dispatcher on 10.0.0.25.

If you were to run the **ndkeys 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 **ndkeys delete** command deletes the public and private keys on the server machine. If these keys are deleted, no remote clients will be authorized to connect to the servers.

For both `ndkeys create` and `ndkeys 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.

Using Network Dispatcher logs

Network Dispatcher 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.

You can set the logging level to define the expansiveness of the messages written to the log. At level 0, errors are logged and Network Dispatcher 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 server log is 0. The default for the manager, advisor, and 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 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.

- To configure the logging level or maximum log size for a server log, use the **ndcontrol set** command.
- To configure the logging level or maximum log size for a manager log, use the **ndcontrol manager** command. This command also controls the log level of the metric monitor log that logs communication with Metric Server agents.
- To configure the logging level or maximum log size for an advisor log, use the **ndcontrol advisor** command.
- To configure the logging level or maximum log size for a subagent log, use the **ndcontrol subagent** command. (Only the Dispatcher component uses the SNMP subagent.)

Changing the log file paths

By default, the logs generated by the Network Dispatcher will be stored in the logs directory of the Network Dispatcher installation. To change this path, set the *nd_logdir* variable in the ndserver script.

AIX, Linux, and Solaris: The ndserver script is found in /usr/bin directory. In this script, the variable *nd_logdir* is set to the default directory. You can modify this variable to specify your log directory. Example:

```
ND_LOGDIR=/path/to/my/logs/
```

Windows 2000: The ndserver file is found in the Windows 2000 system directory, typically C:\WINNT\SYSTEM32. In the ndserver file, the variable *nd_logdir* is set to the default directory. You can modify this variable to specify your log directory. Example:

```
set ND_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 Network Dispatcher uses the same log directory as the other log files. See “Using binary logging to analyze server statistics” on page 180.

Using the Dispatcher component

This section explains how to operate and manage the Dispatcher component.

Starting and Stopping Dispatcher

- Type **ndserver** on a command line to start Dispatcher.
- Type **ndserver stop** on a command line to stop Dispatcher.

Using stale timeout value

For Network Dispatcher, 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, Network Dispatcher will remove that connection record from its tables, and subsequent traffic for that connection will be discarded.

At the port level, for example, you can specify the stale timeout value on the **ndcontrol 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 32,000,000 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 Network Dispatcher's stale timeout value is smaller than the service's timeout value. In the case of LDAP, the Network Dispatcher `staletimeout` value defaults to 300 seconds. If there is no activity on the connection for 300 seconds, Network Dispatcher 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 Network Dispatcher. This will cause 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 Network Dispatcher `staletimeout` value.

Using FIN count to control garbage collection

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 by the garbage collector built into the executor.

You can use the FIN timeout and count to set how often the executor will perform garbage collection and how much it will perform. The executor periodically checks the list of connections it has allocated. When the number of connections in the FIN state is greater than or equal to the FIN count, the executor will attempt to free the memory used to hold this connection information. You can change the FIN count by entering the **`ndcontrol executor set fincount`** command.

The garbage collector frees the memory for any connection that is in the FIN state and is older than the number of seconds specified in the FIN timeout. You can change the FIN timeout by entering the **`ndcontrol executor set fintimeout`** command.

Stale timeout value is the number of seconds during which there can be no activity on a connection before that connection is removed. See "Using stale

timeout value” on page 188, for more information. The FIN count also affects how often “stale” connections are removed. If you have little memory on your Dispatcher machine, you should set the FIN count lower. If you have a busy Web site, you should set the FIN count higher.

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

Note: For Linux, SNMP is not supported on Network Dispatcher.

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 `..nd/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 Protocol Interface (DPI) capability. DPI is an interface between an SNMP agent and its subagents.

Enabling SNMP on AIX and Solaris

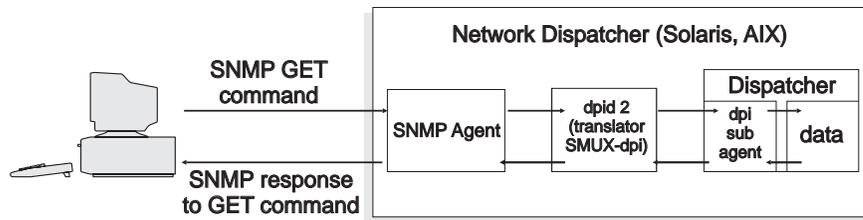


Figure 28. SNMP commands for AIX and Solaris

AIX 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 Solaris, you must obtain an SNMP agent that is SMUX-enabled since Solaris does not provide one. Network Dispatcher provides DPID2 for Solaris in the `/opt/nd/servers/samples/SNMP` directory.

The DPI agent must run as a root user. Before you execute the DPID2 daemon, update the `/etc/snmpd.peers` file and the `/etc/snmpd.conf` file as follows:

- 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
```

Refresh snmpd so that it will reread the /etc/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 Windows 2000

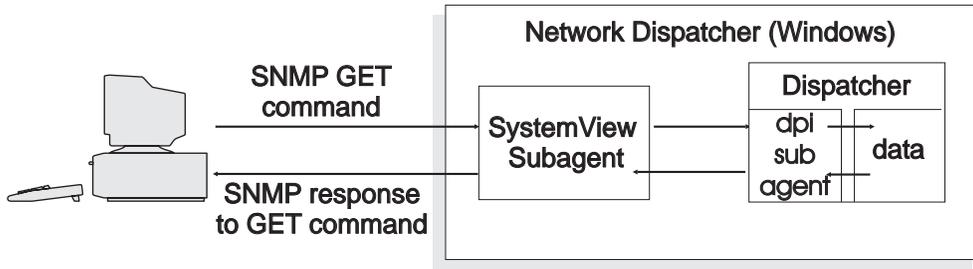


Figure 29. SNMP commands for Windows 2000

To get a DPI-capable SNMP agent for Windows 2000, install the Windows NT version of the IBM SystemView Agent toolkit from <http://www.tivoli.com/support/sva>.

Before you start the SystemView SNMPD process, you must disable the Microsoft Windows SNMP support. The SystemView snmpd supports DPI subagents and Microsoft-compliant agents.

To disable the Windows SNMP support:

1. Click Start->Programs->Administrative Tools->Services.
2. Right click **SNMP**, then select **Properties**.
3. Change **Startup type**: to "Disabled"

Note: If you do not disable the Microsoft Windows SNMP support, the Dispatcher SNMP subagent will not be able to connect to the snmpd agent.

To configure the SystemView SNMP agent, follow the instructions in "Providing a community name for SNMP".

Providing a community name for SNMP

You should configure the SNMP community name. The default SNMP community name is `public`. In UNIX systems, this is set up in a file named `/etc/snmpd.conf`.

On all systems, the community name must be configured and used consistently. That is, if the community name for Network Dispatcher is set to "OurCommunity" in the SNMP agent configuration, it must also be set to "OurCommunity" in the subagent configuration.

For Windows 2000, before creating community name, configure the IBM SystemView SNMP agent.

1. From your desktop, click the **IBM SystemView Agent** icon.
2. Click **snmpcfg**.
3. In the SNMP Configuration dialog box, add the community name. For testing purposes, enter **public** as the community name.
This step allows any host in any network to access the SNMP MIB variables. After you have verified that these values work, you can change them according to your requirements.
4. Check the `\sva\dmi\bin\svastart.bat` file and ensure that the **-dpi** option is specified.
5. Start the SNMP daemon by using `svastart.bat` from the `\sva\dmi\bin` subdirectory.

With the executor running, use the **ndcontrol subagent start [communityname]** command to define the community name used between the Dispatcher DPI subagent and the SNMP agent. The default for community name is `public`. If you change this value, you must also add the new community name to the SystemView Agent using `snmpcfg` as above.

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 csAddr, psNum, ssAddr 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 maxhalfopen threshold for the port specified by the csAddr, psNum portion of the Object Identifier. The number of servers configured on the port is sent in the trap. This trap indicates that Network Dispatcher 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 csAddr, psNum portion of the Object Identifier. The number of servers configured on the port is sent in the trap. When Network Dispatcher determines that the possible Denial of Service attack is over, this trap will be sent after an indDOSAttack trap is sent.

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 will be 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 `ndcontrol` command

The `ndcontrol subagent start` command turns the SNMP support on. The `ndcontrol subagent stop` command turns the SNMP support off.

For more information about the `ndcontrol` command, see “`ndcontrol subagent — configure SNMP subagent`” on page 282.

Using `ipchains` or `iptables` to reject all traffic to (harden) the Network Dispatcher box (on Linux)

Built into the Linux kernel is a firewall facility called `ipchains`. When Network Dispatcher and `ipchains` run concurrently, Network Dispatcher sees packets first, followed by `ipchains`. This allows the use of `ipchains` to harden a Linux Network Dispatcher box, which could be, for example, a Network Dispatcher box that is used to load balance firewalls.

When `ipchains` or `iptables` is configured as completely restricted (no inbound or outbound traffic permitted), the packet-forwarding portion of Network Dispatcher 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 Network Dispatcher to function properly. Some examples of this communication are:

- Advisors communicate between the Network Dispatcher box and the back-end servers.
- Network Dispatcher pings back-end servers, reach targets, and high availability partner Network Dispatcher boxes.
- User interfaces (graphical user interface, command line, and wizards) use RMI.
- Back-end servers must respond to pings from the Network Dispatcher box.

In general, an appropriate `ipchains` strategy for the Network Dispatcher boxes is to disallow all traffic, except that which is to or from the back-end servers, the partner high availability Network Dispatcher, any reach targets, or any configuration hosts.

Using the Content Based Routing component

This section explains how to operate and manage the CBR component of Network Dispatcher.

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 via the Caching Proxy plugin 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 85.

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 “Appendix B. Command reference for Dispatcher, CBR, and Mailbox Locator” on page 225. Some example uses are listed here.
- Configure CBR using the graphical user interface (GUI). Type **ndadmin** on the command line to open the GUI. See “GUI” on page 78 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 Network Dispatcher logs” on page 187.

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 188.

Using the Mailbox Locator component

Starting and stopping Mailbox Locator

- Type **mlserver** on a command line to start Mailbox Locator.
- Type **mlserver stop** on a command line to stop Mailbox Locator.

Controlling Mailbox Locator

After starting Mailbox Locator, you can control it using either of the following methods:

- Configure Mailbox Locator through the **mlcontrol** command. The complete syntax of this command is described in “Appendix B. Command reference for Dispatcher, CBR, and Mailbox Locator” on page 225. Some example uses are listed here.

- Configure Mailbox Locator using the graphical user interface (GUI). Type **ndadmin** on the command line to open the GUI. See “GUI” on page 93 for more information on how to configure Mailbox Locator using the GUI.

Using Mailbox Locator logs

The logs used by Mailbox Locator are similar to those used in Dispatcher. For more description, see “Using Network Dispatcher logs” on page 187.

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 “Appendix D. Command reference for Site Selector” on page 289. Some example uses are listed here.
- Configure Site Selector using the graphical user interface (GUI). Type **ndadmin** on the command line to open the GUI. See “GUI” on page 105 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 Network Dispatcher logs” on page 187.

Using the Cisco Consultant component

Starting and stopping Cisco Consultant

1. Type **lbcserver** on a command line to start Cisco Consultant.
2. Type **lbcserver stop** on a command line to stop Cisco Consultant.

Controlling Cisco Consultant

After starting Cisco Consultant, you can control it using either of the following methods:

- Configure Cisco Consultant through the **lbccontrol** command. The complete syntax of this command is described in “Appendix E. Command reference for Consultant for Cisco CSS Switches” on page 315. Some example uses are listed here.
- Configure Cisco Consultant using the graphical user interface (GUI). Type **ndadmin** on the command line to open the GUI. See “GUI” on page 105 for more information on how to configure Cisco Consultant using the GUI.

Using Cisco Consultant logs

The logs used by Cisco Consultant are similar to those used in Dispatcher. For more description, see “Using Network Dispatcher logs” on page 187.

Using the Metric Server component

Starting and stopping Metric Server

Metric Server provides server load information to the Network Dispatcher. Metric Server resides on each of the servers that are being load balanced.

- 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.

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 Network Dispatcher logs. This will generate an agent log in the **...ms/logs** directory.

Chapter 16. Troubleshooting

This chapter helps you detect and resolve problems associated with Network Dispatcher. Find the symptom you are experiencing in “Troubleshooting tables”.

Troubleshooting tables

These are the troubleshooting tables for Dispatcher, CBR, Mailbox Locator, Site Selector, and Consultant for Cisco CSS Switches.

Table 14. Dispatcher troubleshooting table

Symptom	Possible Cause	Go to...
Dispatcher not running correctly	Conflicting port numbers	“Checking Dispatcher port numbers” on page 204
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 207
Connections from client machines not being served or connections timing out	<ul style="list-style-type: none">• 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• Servers are down or set to a weight of zero	“Problem: Dispatcher requests are not being balanced” on page 207
Client machines are not being served or are timing out	High availability not working	“Problem: Dispatcher high-availability function is not working” on page 208
Unable to add heartbeat (Windows 2000)	Source address is not configured on an adapter	“Problem: Unable to add heartbeat (Windows 2000)” on page 208
Server not serving requests (Window)	An extra route has been created in the routing table	“Problem: Extra routes (Windows 2000)” on page 208

Table 14. Dispatcher troubleshooting table (continued)

Symptom	Possible Cause	Go to...
Advisors not working correctly with wide area	Advisors are not running on remote machines	"Problem: Advisors not working correctly" on page 208
SNMPD will not start or will not continue to run (Windows 2000)	The community name passed in the SNMP commands does not agree with the community name with which the subagent was started	"Problem: SNMPD does not run correctly (Windows 2000)" on page 208
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 2000)" on page 208
Connection to remote machine refused	Older version of the keys is still being used	"Problem: Dispatcher connection to a remote machine" on page 209
The ndcontrol or ndadmin command fails with 'Server not responding' or 'unable to access RMI server' message	<ol style="list-style-type: none"> 1. Commands fail due to socksified stack. Or commands fail due to not starting ndserver 2. RMI ports are not set correctly 	"Problem: ndcontrol or ndadmin command fails" on page 209
"Cannot Find the File..." error message, when running Netscape as default browser to view online help (Windows 2000)	Incorrect setting for HTML file association	"Problem: "Cannot find the file..." error message when trying to view online Help (Windows 2000)" on page 210
"stty: : No such device or address" error message, when starting ndserver on Solaris 2.7.	Please disregard this error message. This is not a problem. Ndservr will run correctly	"Problem: Spurious error message when starting ndserver on Solaris 2.7" on page 210
Graphical user interface does not start correctly	Insufficient paging space	"Problem: Graphical user interface (GUI) does not start correctly" on page 210
Error running Dispatcher with Caching Proxy installed	Caching Proxy file dependency	"Problem: Error running Dispatcher with Caching Proxy installed" on page 210

Table 14. Dispatcher troubleshooting table (continued)

Symptom	Possible Cause	Go to...
Graphical user interface does not display correctly.	Resolution is incorrect.	"Problem: Graphical user interface (GUI) does not display correctly" on page 210
Help panels sometimes disappear behind other windows	Java limitation	"Problem: On Windows 2000, help windows sometimes disappear behind other open windows" on page 211
Network Dispatcher cannot process and forward a frame	Need a unique MAC address for each NIC	"Problem: Network Dispatcher cannot process and forward a frame" on page 211
Blue screen appears	No installed and configured network card	"Problem: A blue screen displays when you start the Network Dispatcher executor" on page 211
Path to Discovery prevents return traffic	The cluster is aliased on the loopback	"Problem: Path to Discovery prevents return traffic with Network Dispatcher" on page 211
Advisors show that all servers are down	TCP checksum is not computed correctly	"Problem: Advisors show that all servers are down" on page 212
High availability in the Wide Area mode of Network Dispatcher 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 Network Dispatcher does not work" on page 213
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 213

Table 15. CBR Troubleshooting table

Symptom	Possible Cause	Go to...
CBR not running correctly	Conflicting port numbers	"Checking CBR port numbers" on page 205

Table 15. CBR Troubleshooting table (continued)

The cbrcontrol or ndadmin 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 ndadmin command fails" on page 214
Requests are not being load balanced	Caching Proxy was started before the executor was started	"Problem: Requests not being load balanced" on page 214
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	"Problem: On Solaris, cbrcontrol executor start command fails" on page 215
URL rule doesn't work	Syntactical or configuration error	"Problem: Syntactical or configuration error" on page 215

Table 16. Mailbox Locator Troubleshooting table

Symptom	Possible Cause	Go to...
Mailbox Locator not running correctly	Conflicting port numbers	"Checking Mailbox Locator port numbers" on page 205
The mlserver command returns a "java.rmi.RMI Security Exception: security.fd.read" exception	The system's limit on file descriptors is too small for the number of requests that mlserver is trying to service	"Problem: The mlserver command is stopped" on page 215
The mlcontrol or ndadmin 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 mlserver.	"Problem: mlcontrol or ndadmin command fails" on page 216
Unable to add a port	Another application is already listening to that port	"Problem: Unable to add a port" on page 216
Receive proxy error when trying to add a port	The cluster address was not configured on a NIC before the proxy was started. Or, another application is running on that port.	"Problem: Receive proxy error when trying to add a port" on page 216

Table 17. Site Selector troubleshooting table

Symptom	Possible Cause	Go to...
Site Selector not running correctly	Conflicting port number	"Checking Site Selector port numbers" on page 206
Site Selector does not round-robin incoming requests from Solaris client	Solaris systems run a "name service cache daemon"	"Problem: Site Selector doesn't round-robin traffic from Solaris clients" on page 217
The sscontrol or ndadmin 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 ndadmin command fails" on page 217
sssserver fails to start on Windows 2000	Windows does not require the host name to be in the DNS.	"Problem: sssserver is failing to start on Windows 2000" on page 217
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 217

Table 18. Consultant for Cisco CSS Switches troubleshooting table

Symptom	Possible Cause	Go to...
lbserver will not start	Conflicting port numbers	"Checking Cisco Consultant port numbers" on page 206
The lbcontrol or ndadmin 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 lbserver.	"Problem: lbcontrol or ndadmin command fails" on page 218
receive error: Cannot create registry on port 14099	Expired product license	"Problem: Cannot create registry on port 14099" on page 218

Table 19. Metric Server troubleshooting table

Symptom	Possible Cause	Go to...
Metric Server IOException on Windows 2000 running .bat or .cmd user metric files	Full metric name is required	"Problem: Metric Server IOException on Windows 2000 running .bat or .cmd user metric files" on page 218

Table 19. Metric Server troubleshooting table (continued)

Symptom	Possible Cause	Go to...
Metric Server not reporting the load information to the Network Dispatcher machine	Possible causes include: <ul style="list-style-type: none"> no key files on Metric Server machine host name of Metric Server machine not registered with local nameserver 	“Problem: Metric Server not reporting loads to Network Dispatcher machine” on page 219
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 219

Checking Dispatcher port numbers

If you are experiencing problems running the 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 ndcontrol
- 10004 to send metric queries to Metric Server
- 10005 to receive information from an SDA agent

If another application is using one of the Dispatcher port numbers, you can change the Dispatcher’s port number by doing the following:

- To change the port used to receive commands
 - Modify the ND_RMIPORT variable at the top of the ndserver 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 metricsserver 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 **ndcontrol manager start** command syntax “ndcontrol manager — control the manager” on page 254
- To change the port used to receive SDA information, change the ND_AFFINITY_PORT variable in the ndserver file to the port Dispatcher should use to receive SDA information.

Note: For Windows 2000, ndserver and metricsserver files are in the c:\winnt\system32 directory. For other platforms, these file are in the /usr/bin/ directory.

Checking CBR port numbers

If you are experiencing problems running the 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

If another application is using one of the CBR port numbers, you can change the CBR's port number by doing the following:

- To change the port used to receive commands
 - Modify the `ND_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 `metricsserver` 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 “`ndcontrol manager — control the manager`” on page 254

Note: For Windows 2000, `cbrserver` and `metricsserver` files are in the `c:\winnt\system32` directory. For other platforms, these file are in the `/usr/bin/` directory.

Checking Mailbox Locator port numbers

If you are experiencing problems running the Mailbox Locator, it may be that one of your applications is using a port number that Mailbox Locator normally uses. Be aware that Mailbox Locator uses the following port numbers:

- 13099 to receive commands from mlcontrol
- 10004 to send metric queries to Metric Server

If another application is using one of the Mailbox Locator port numbers, you can change the Mailbox Locator's port number by doing the following:

- To change the port used to receive commands
 - Modify the `ND_RMIPORT` variable at the top of the `mlserver` file to the port Mailbox Locator that you want Mailbox Locator to receive commands.
- To change the port used to receive metric reports from Metric Server
 - Modify the `RMI_PORT` variable in the `metricsserver` file to the port that you want Mailbox Locator to communicate with Metric Server.

- Provide the `metric_port` argument when the manager is started. See the description of the **manager start** command syntax “`ndcontrol manager — control the manager`” on page 254

Note: For Windows 2000, `mlserver` and `metricsserver` files are in the `c:\winnt\system32` 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

If another application is using one of the Site Selector port numbers, you can change the Site Selector’s port number by doing the following:

- To change the port used to receive commands,
 - Modify the `ND_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 `metricsserver` 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 298

Note: For Windows 2000, `ssserver` and `metricsserver` files are in the `c:\winnt\system32` directory. For other platforms, these file are in the `/usr/bin/` directory.

Checking Cisco Consultant port numbers

If you are experiencing problems running the Cisco Consultant component, it may be that another application is using one of the port numbers used by Cisco Consultant’s `lbserver`. Be aware that Cisco Consultant uses the following port numbers:

- 14099 to receive commands from `lbcontrol`
- 10004 to send metric queries to Metric Server

If another application is using one of the Consultant port numbers, you can change the port numbers for Consultant by doing the following:

- To change the port used to receive commands from lbcontrol, modify the ND_RMIPORT variable in the lbserver file. Change from 14099 to the port on which you want Consultant to receive lbcontrol 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 Consultant to communicate with Metric Server.
 2. Provide the metric_port argument when you start the manager. Refer to “lbcontrol manager — control the manager” on page 330 for a description of the lbcontrol manager start command syntax.

Note: For Windows 2000, lbserver and metricserver files are in the c:\winnt\system32 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 204.

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.

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? Use netstat -ni to check.
3. Does the loopback device on each server have the alias set to the cluster address? Use netstat -ni to check.
4. Is the extra route deleted? Use netstat -nr to check.
5. Use the **ndcontrol 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 **ndcontrol server report::** command to make sure that your servers are neither down nor set to a weight of zero.

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 155
- For **AIX**, **Linux**, and **Solaris**, make sure the goActive, goStandby, and goInOp scripts have execute permission set.
- For **Windows 2000**, be sure to configure the nonforwarding address.

Problem: Unable to add heartbeat (Windows 2000)

This Windows 2000 error occurs when the source address is not configured on an adapter. Check the following to correct or diagnose the problem.

- For **Windows 2000** be sure to configure the nonforwarding address using either the token-ring or ethernet interface and issuing either of the following commands:

```
ndconfig tr0 <ip address> netmask <netmask> or  
ndcontrol cluster configure
```

Problem: Extra routes (Windows 2000)

After setting up server machines, you may find that you have inadvertently created one or more extra routes. If not removed, these extra routes will prevent the Dispatcher from operating. To check for and delete them, see “Setting up server machines for load balancing” on page 62.

Problem: Advisors not working correctly

If you are using wide area support, and your advisors do not seem to be working correctly, make sure that they are started on both the local and the remote Dispatchers. See “Using remote advisors with wide area support” on page 144.

Problem: SNMPD does not run correctly (Windows 2000)

When using SNMP subagents, if the SystemView Agent SNMP daemon does not start and stay up, be sure that you have configured your SNMP community using the snmpcfg program. To access SNMP data from the Dispatcher subagent, the community name passed in the SNMP commands must agree with the community name with which the subagent was started.

Problem: Dispatcher, Microsoft IIS, and SSL do not work (Windows 2000)

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 Information and Planning Guide*, which comes with Windows 2000. The local URL for the document, which is viewed by a

web browser, is:

file:///C:/WINNT/system32/inetsrv/iisadmin/htmldocs/inetdocs.htm.

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: ndcontrol or ndadmin command fails

1. The ndcontrol command returns: **Error: Server not responding**. Or, the ndadmin 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 jre
EXCLUDE-MODULE jrew
EXCLUDE-MODULE javaw
```

2. The administration consoles for Network Dispatcher interfaces, (command line, graphical user interface, and wizards) communicate with ndserver using remote method invocation (RMI). The default communication uses two ports; one port is set in the ndserver start script and the other port is random.

The random port can cause problems when one of the administration consoles runs on the same machine as a firewall or through a firewall. For example, when Network Dispatcher runs on the same machine as a firewall, and you issue ndcontrol commands, you might see errors such as **Error: Server not responding**.

To avoid this problem, edit the ndserver script file (located in the PATH) to set the random port used by RMI. Include `-DND_RMI_SERVER_PORT=yourPort` within the `END_ACCESS` string, where *yourPort* is the port you specify.

For example:

```
END_ACCESS='-DND_CLIENT_KEYS_DIRECTORY=/usr/lpp/nd/admin/keys/dispatcher
-DND_SERVER_KEYS_DIRECTORY=/usr/lpp/nd/dispatcher/key
-DND_RMI_SERVER_PORT=10100'
ND_RMIPORT=10099
```

Once complete, restart ndserver and open traffic for ports 10099 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 **ndserver**.

Problem: “Cannot find the file...” error message when trying to view online Help (Windows 2000)

For Windows 2000 when using Netscape as your default browser, the error message which results with this problem is: “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: Spurious error message when starting ndserver on Solaris 2.7

When starting ndserver on Solaris 2.7 platforms, the following spurious error message appears: “stty: : No such device or address.” Please disregard this error message. Ndservr will run correctly.

Problem: Graphical user interface (GUI) does not start correctly

The graphical user interface (GUI), which is ndadmin, 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 Network Dispatcher 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 Network Dispatcher.
3. Reinstall both Network Dispatcher and Caching Proxy.

Problem: Graphical user interface (GUI) does not display correctly

If you experience a problem with the appearance of the Network Dispatcher 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 2000, help windows sometimes disappear behind other open windows

When you first open help windows on Windows 2000, they sometimes disappear into the background behind existing windows. If this occurs, click on the window to bring it forward again.

Problem: Network Dispatcher 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 Network Dispatcher as configured in `ibmnd.conf`, and if the NIC that is not defined in `ibmnd.conf` receives a frame, Network Dispatcher 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 hme0 ether 01:02:03:04:05:06
```

Problem: A blue screen displays when you start the Network Dispatcher executor

On Windows 2000, you must have a network card installed and configured before starting the executor.

Problem: Path to Discovery prevents return traffic with Network Dispatcher

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 creates 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 uses the loopback for the route. This causes the server's responses to never leave the box, 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.

There are two solutions to this problem.

1. Disable the path MTU discovery so that AIX is not dynamically adding routes. Use the following commands.

```
no -a lists AIX networking settings
```

```
no -o option=value
sets TCP parameters on AIX
```

2. Alias the cluster IP on the loopback with a 255.255.255.255 netmask. This means that the aliased subnet is only the cluster IP. When AIX creates the dynamic routes, the target gateway IP does not match that subnet, resulting in a route using the correct network interface. Then delete the new lo0 route that was created during the aliasing step. To do this, find the route on the loopback with a network destination of the cluster IP and delete that route. This must be done every time the cluster is aliased.

Notes:

1. The path MTU discovery is disabled by default in AIX 4.3.2 and lower; however, in AIX 4.3.3 and higher, it is enabled by default.
2. The following commands turn off path MTU discovery and must be performed at every boot of the system. Add these commands to the /etc/rc.net file.
 - -o udp_pmtu_discover=0
 - -o tcp_pmtu_discover=0

Problem: Advisors show that all servers are down

Windows 2000 has a new feature called Task Offload that allows the TCP checksum to be calculated by the adapter card rather than the operating system. This improves performance on the system. If Task Offload is enabled, Network Dispatcher advisors report that servers are down when they are not.

The problem is that the TCP checksum is not computed correctly for packets coming from the cluster address, which is what happens with advisor traffic.

To avoid this problem, go to the adapter card settings and disable Task Offload.

This problem was first observed with Adaptec's ANA62044 QuadPort Adapter. This adapter card refers to the feature as the Transmit Checksum offload. Disable Transmit Checksum offload to avoid the problem.

Problem: High availability in the Wide Area mode of Network Dispatcher does not work

When you set up a Wide Area Network Dispatcher, 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 goStandby 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 trying 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 change to the GUI.

There is an option on the runtime environment that can be specified to increase the memory allocation pool available to Java.

The option is `-Xmxn` 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 64MB. Solaris 7 and Solaris 8 SPARC platforms have a maximum value of 4000m; Solaris 2.6 and x86 platforms have a maximum value of 2000m.

To add this option, modify the ndadmin script file as follows:

- **Windows NT or 2000**

```
START jrew -mx64m %END_ACCESS% %CONFIG_DIR%  
-DEND_INSTALL_PATH=%IBMNDPATH% -cp %NDCLASSPATH%  
com.ibm.internet.nd.framework.FWK_Framework  
com.ibm.internet.nd.gui.GUI_eNDRootNode1
```

- **Solaris**

```
$JREDIR/$JRE -mx64m $END_ACCESS $CONFIG_DIR  
-DEND_INSTALL_PATH=/opt/&BASEDIR -cp $NDCLASSPATH  
com.ibm.internet.nd.framework.FWK_Framework  
com.ibm.internet.nd.gui.GUI_eNDRootNode &1
```

- **Linux**

```
re -mx64m $END_ACCESS $CONFIG_DIR $NDLOCALE  
-DEND_INSTALL_PATH=/opt/nd -classpath $NDCLASSPATH  
com.ibm.internet.nd.framework.FWK_Framework  
com.ibm.internet.nd.gui.GUI_eNDRootNode 1>/dev/null 2>&1 &1
```

- **AIX**

```
ava -mx64m $END_ACCESS $CONFIG_DIR $NDLOCALE  
-DEND_INSTALL_PATH=/usr/lpp/&BASEDIR -classpath $NDCLASSPATH  
com.ibm.internet.nd.framework.FWK_Framework  
com.ibm.internet.nd.gui.GUI_eNDRootNode 1>/dev/null 2>&1 &
```

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.

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 205.

Problem: cbrcontrol or ndadmin command fails

The cbrcontrol command returns: “Error: Server not responding.” Or, the ndadmin 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 jre  
EXCLUDE-MODULE jrew  
EXCLUDE-MODULE javaw
```

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, cbrcontrol executor start command fails

On Solaris, 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 80.

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 C. Content rule (pattern) syntax" on page 285, 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.

Solving common problems—Mailbox Locator

Problem: Mailbox Locator will not run

This problem can occur when another application is using one of the ports used by Mailbox Locator. For more information, go to "Checking Mailbox Locator port numbers" on page 205.

Problem: The mlserver command is stopped

On a UNIX platform, this problem occurs when **mlserver** is used to load balance a large number of IMAP/POP3 client requests and the system's limit on file descriptors is too small for the number of requests that **mlserver** is trying to service. The **mlserver** produces the following exception and then is stopped:

```
java.rmi.RMIException: security.fd.read
```

The protocol specific proxy log file reports:

```
SocketException=java.net.SocketException: Socket closed
```

The solution is to modify the **nfiles** (AIX, Linux) or the **open files** (Solaris) limit in the shell where **mlserver** is started. Increase the **nfiles** limit to a

reasonable number larger than the current nofiles limit. Use `ulimit -a` to display the current nofiles limit, and use `ulimit -n value` to increase the value.

Problem: mlcontrol or ndadmin command fails

The `mlcontrol` command returns: "Error: Server not responding." Or, the `ndadmin` 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 jre
EXCLUDE-MODULE jrew
EXCLUDE-MODULE javaw
```

These errors can also occur if you have not already started `mlserver`.

Problem: Unable to add a port

When attempting to add a port to a configuration, you might receive this error message: **Error: Unable to add port**. It is possible that another application is already listening on that port. Mailbox Locator tries to start a proxy that binds to the cluster IP on the specified port in the command. If another application is binding to that IP or listening to all IPs on that port, the proxy startup fails. To use Mailbox Locator on that port, you must stop the conflicting application.

Note that on the Linux platform, the `xinetd` daemon can start a listener without running, for example, a POP3 program. It is, therefore, important to check `netstat -a` to determine if any application is listening on the intended port.

Problem: Receive proxy error when trying to add a port

For Mailbox Locator, the `mlcontrol port add` command produces the following error message: "The proxy on cluster <cluster>, port <port> did not start." The solution is to configure the cluster address on a NIC before the proxy can be started. Also, verify that no other application is running on that port listening for the cluster address (including a general listen-on-everything application).

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 206.

Problem: Site Selector doesn't 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 will be answered from this cache instead of querying Site Selector.

Solution: Turn off the name service cache daemon on the Solaris machine.

Problem: ssscontrol or ndadmin command fails

The ssscontrol command returns: "Error: Server not responding." Or, the ndadmin 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 jre
EXCLUDE-MODULE jrew
EXCLUDE-MODULE javaw
```

These errors can also occur if you have not already started sssserver.

Problem: sssserver is failing to start on Windows 2000

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 does not always require the configured host name to be in the DNS. Site Selector requires that its host name be defined in the DNS to start properly.

Verify this host is defined in the DNS. Edit the sssserver.cmd file and remove the "w" from "javaw". This should provide more 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.

Solving common problems—Consultant for Cisco CSS Switches

Problem: lbcserver will not start

This problem can occur when another application is using one of the ports used by the Consultant's lbcserver. For more information, see "Checking Cisco Consultant port numbers" on page 206.

Problem: lbcontrol or ndadmin command fails

The lbcontrol command returns: "Error: Server not responding." Or, the ndadmin 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 jre
EXCLUDE-MODULE jrew
EXCLUDE-MODULE javaw
```

These errors can also occur if you have not already started **lbcserver**.

Problem: Cannot create registry on port 14099

This problem can occur when a valid product license is missing. When you attempt to start lbcserver, 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 lbcserver, type **lbcserver stop**.
2. Copy your valid license to the **...nd/servers/conf** directory.
3. Type **lbcserver** to start the server.

Solving common problems—Metric Server

Problem: Metric Server IOException on Windows 2000 running .bat or .cmd user metric files

You must use the full metric name for user-written metrics on Windows 2000 Metric Servers. For example, you must specify **usermetric.bat** instead of **usermetric**. The name **usermetric** is valid on the command line, but will not work when executed 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 Network Dispatcher machine

There can be several reasons why Metric Server is not reporting load information to Network Dispatcher. 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.
- Restart with a higher loglevel and look for errors.
- On the Network Dispatcher machine, increase the manager loglevel. Search for errors in the Metric Monitor log.

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.

Appendix A. 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.

▶▶—*variable*—▶▶

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*.

▶▶—*cluster:port*—▶▶

Appendix B. Command reference for Dispatcher, CBR, and Mailbox Locator

This appendix describes how to use the Dispatcher **ndcontrol** commands. It is also a command reference for CBR and Mailbox Locator. CBR and Mailbox Locator use a subset of the Dispatcher commands. See “Configuration differences between CBR, Mailbox Locator, and Dispatcher” on page 226 for more information.

Note: When using these syntax diagram —

- For CBR, substitute **cbrcontrol** for **ndcontrol**
- For Mailbox Locator, substitute **mlcontrol** for **ndcontrol**

Below is a list of commands in this appendix:

- “**ndcontrol** advisor — control the advisor” on page 228
- “**ndcontrol** cluster — configure clusters” on page 234
- “**ndcontrol** executor — control the executor” on page 239
- “**ndcontrol** file — manage configuration files” on page 244
- “**ndcontrol** help — display or print help for this command” on page 246
- “**ndcontrol** highavailability — control high availability” on page 248
- “**ndcontrol** host — configure a remote machine” on page 252
- “**ndcontrol** log — control the binary log file” on page 253
- “**ndcontrol** manager — control the manager” on page 254
- “**ndcontrol** metric — configure system metrics” on page 260
- “**ndcontrol** port — configure ports” on page 261
- “**ndcontrol** rule — configure rules” on page 267
- “**ndcontrol** server — configure servers” on page 274
- “**ndcontrol** set — configure server log” on page 280
- “**ndcontrol** status — display whether the manager and advisors are running” on page 281
- “**ndcontrol** subagent — configure SNMP subagent” on page 282

You can enter a minimized version of the **ndcontrol** 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 **ndcontrol he f** instead of **ndcontrol help file**.

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

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

Note: 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, Mailbox Locator, and Dispatcher

The CBR and Mailbox Locator command line interface is for the most part a subset of the command line interface of Dispatcher. Use the **cbrcontrol** command (for the CBR component) or use the **mlcontrol** command (for the Mailbox Locator component) instead of ndcontrol to configure the component.

Some of the commands that are *omitted* in CBR are listed below.

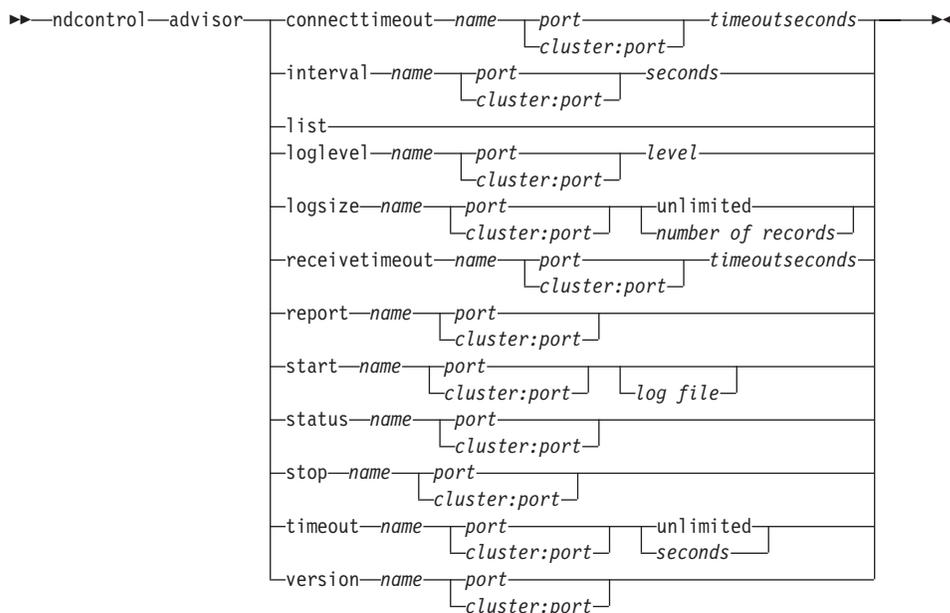
1. highavailability
2. subagent
3. executor
 - report
 - set nfa <value>
 - set fincount <value>
 - set fintimeout <value>
 - set porttype <value>
4. cluster
 - report {c}
 - set {c} porttype
5. port add {c:p} porttype
6. port set {c:p} porttype
7. rule add {c:p:r} type port
8. server add {c:p:s} router
9. server set {c:p:s} router

Some of the commands that are *omitted* in Mailbox Locator are listed below.

1. highavailability
2. rule
3. subagent
4. executor
 - start

- stop
 - report
 - set nfa <value>
 - set fincount <value>
 - set fintimeout <value>
 - set porttype <value>
5. cluster
 - report {c}
 - set {c} porttype
 6. port [add | set] {c:p} porttype
 7. server [add | set] {c:p:s} router

ndcontrol 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 129.

name

The name of the advisor. Possible values include **connect**, **db2**, **dns**, **ftp**, **http**, **ibmproxy (Caching Proxy)**, **imap**, **nntp**, **ping**, **pop3**, **self**, **smtp**, **ssl**, **ssl2http**, **telnet**, and **wlm**.

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 131 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 127 for more information.

The cluster is the address in dotted-decimal 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 will be 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 server fails. For more information, see “Advisor connect timeout and receive timeout for servers” on page 129.

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.

start

Start the advisor. There are advisors for each protocol. The default ports are as follows:

Advisor Name	Protocol	Port
connect	ICMP	12345
db2	private	50000
dns	DNS	53
ftp	FTP	21
http	HTTP	80
ibmproxy	HTTP (via Caching Proxy)	80
imap	IMAP	143
nntp	NNTP	119
ping	PING	0
pop3	POP3	110
self	private	12345
smtp	SMTP	25
ssl	HTTP	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 will be time-stamped.

The default file is *advisorname_port.log*, for example, **http_80.log**. To change the directory where the log files will be kept, see “Changing the log file paths” on page 188. 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:
`ndcontrol advisor start http 127.40.50.1:80`
- To start the http advisor on port 88 for all clusters:
`ndcontrol advisor start http 88`
- To stop the http advisor at port 80 for cluster 127.40.50.1:
`ndcontrol 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:
`ndcontrol 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:
`ndcontrol advisor connecttimeout http 127.40.50.1:80 20`
- To set the interval for the FTP advisor (for port 21) to 6 seconds:
`ndcontrol advisor interval ftp 21 6`
- To display the list of advisors currently providing information to the manager:
`ndcontrol 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:
ndcontrol advisor loglevel http 80 0
- To change the ftp advisor log size for port 21 to 5000 bytes:
ndcontrol 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:
ndcontrol advisor receivetimeout http 80 60
- To display a report on the state of the ftp advisor (for port 21):
ndcontrol 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:
ndcontrol 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
```

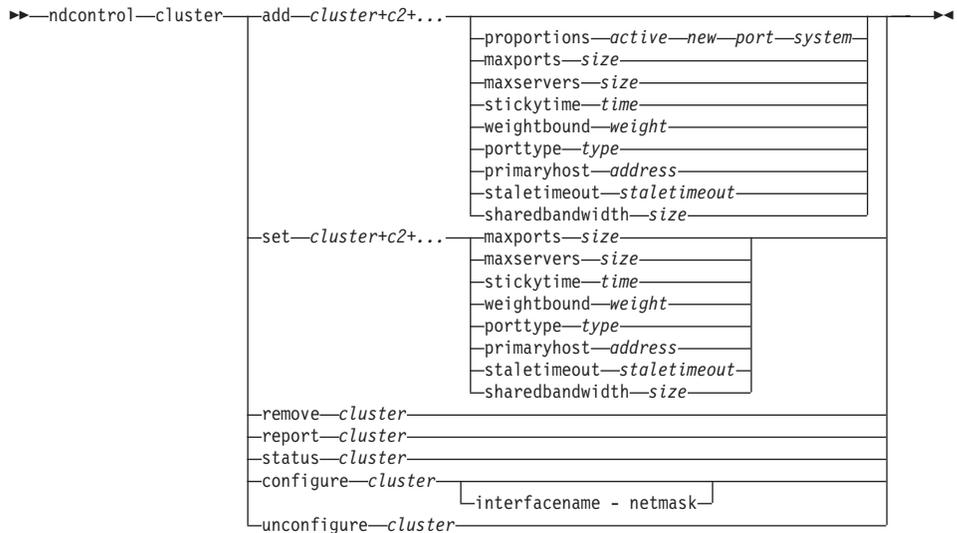
- To set the timeout value for the ftp advisor information on port 21 to 5 seconds:
ndcontrol advisor timeout ftp 21 5
- To display the current version number of the ssl advisor for port 443:

```
ndcontrol advisor version ssl 443
```

This command produces output similar to the following:

```
Version: 04.00.00.00 - 07/12/2001-10:09:56-EDT
```

ndcontrol cluster — configure clusters



add

Add this cluster. You must define at least one cluster.

cluster

The address of the cluster as either a symbolic name or in dotted-decimal format. A cluster address value of 0.0.0.0 can be used to specify a wildcard cluster. See “Use wildcard cluster to combine server configurations” on page 168 for more information.

With the exception of the ndcontrol cluster add command, you can use a colon (:) to act as a wild card. For example, the following command, `ndcontrol cluster set : weightbound 80`, will result in setting a weightbound of 80 to all clusters.

Note: Additional clusters are separated by a plus sign (+).

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 122.

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, Network Dispatcher 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:

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**.

Note: Porttype applies to Dispatcher.

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 once 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 32,000,000. 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 188, for more information.

Note: For Mailbox Locator, staletimeout corresponds to the inactivity autologout timer for these protocols. Staletimeout, for Mailbox Locator defaults to 60 seconds, which overrides the inactivity timeouts for POP3 and IMAP. For more information on staletimeout for Mailbox Locator, see “Overriding the POP3/IMAP inactivity timer” on page 89.

staletimeout

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 161 and “Shared bandwidth rule” on page 162.

Note: Shared bandwidth does not apply to CBR or Mailbox Locator.

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 does not apply to CBR or Mailbox Locator.

status

Show current status of a specific cluster.

configure

Configures a cluster alias to the network interface card.

Note: Configure does not apply to CBR or Mailbox Locator.

interfacename netmask

It is required if it is a different alias from what Dispatcher first finds.

unconfigure

Deletes the cluster alias from the network interface card.

Note: Unconfigure does not apply to CBR or Mailbox Locator.

Examples

- To add cluster address 130.40.52.153:
ndcontrol cluster add 130.40.52.153
- To remove cluster address 130.40.52.153:
ndcontrol 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:
ndcontrol cluster set 9.6.54.12 proportions 60 35 5 0
- To add a wildcard cluster:
ndcontrol 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:
`ndcontrol cluster set 9.6.54.12 primaryhost 9.65.70.19`
- To show the status for cluster address 9.67.131.167:
`ndcontrol cluster status 9.67.131.167`

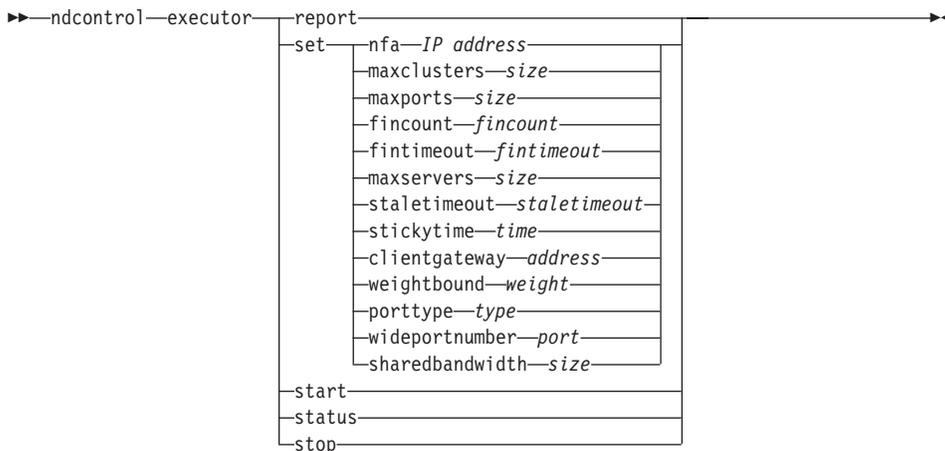
This command produces output similar to:

```

Cluster Status:
-----
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

```

ndcontrol executor — control the executor



report

Display a statistics snapshot report. For example: total packets received, packets discarded, packets forwarded with errors, etc.

Note: Report does not apply to CBR or Mailbox Locator.

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 does not apply to CBR or Mailbox Locator.

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.

fincount

The number of connections that must be in FIN state before garbage collection of connections will be initiated. The default value of fincount is 4000.

fincount

The fincount value.

Note: Fincount does not apply to CBR or Mailbox Locator.

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 60.

fintimeout

The fintimeout value.

Note: Fintimeout does not apply to CBR or Mailbox Locator.

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.

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 32,000,000. 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 188, for more information.

Note: For Mailbox Locator, staletimeout corresponds to the inactivity autologout timer for these protocols. Staletimeout, for Mailbox Locator, defaults to 60 seconds, which overrides the inactivity timeouts for POP3 and IMAP. For more information on staletimeout for Mailbox Locator, see "Overriding the POP3/IMAP inactivity timer" on page 89.

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 Network Dispatcher 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 47 and "Dispatcher's content-based routing (cbr forwarding method)" on page 49 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 does not apply to CBR or Mailbox Locator.

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 does not apply to CBR or Mailbox Locator.

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 161 and “Shared bandwidth rule” on page 162.

Note: Shared bandwidth does not apply to CBR or Mailbox Locator.

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.

start

Start the executor.

Note: Start does not apply to Mailbox Locator.

status

Display the current status of the values in the executor that can be set and their defaults.

stop

Stop the executor. For Dispatcher, stop is *not* a valid parameter on Windows 2000.

Note: Stop applies to Dispatcher and CBR.

Examples

- To display the internal counters for Dispatcher:

```
ndcontrol executor status
```

```
Executor Status:
```

```
-----
```

```
Nonforwarding address ..... 9.67.131.151
Client gateway address ..... 0.0.0.0
Fin count ..... 4,000
Fin timeout ..... 60
Wide area network port number ..... 2,001
Shared bandwidth (Kbytes) ..... 0
Default maximum ports per cluster ... 8
Maximum number of clusters ..... 100
Default maximum servers per port ... 32
Port stale timeout ..... 300
Port sticky time ..... 0
Port weight bound ..... 20
Maximum number of clusters ..... 100
```

- To set the nonforwarding address to 130.40.52.167:

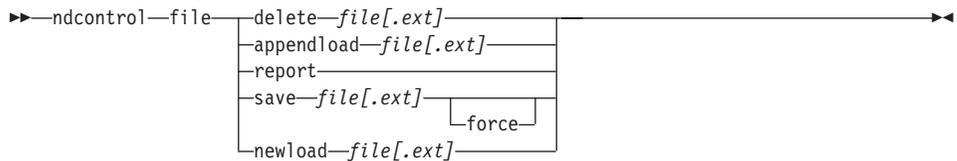
```
ndcontrol executor set nfa 130.40.52.167
```

- To set the maximum number of clusters:

```
ndcontrol executor set maxclusters 4096
```

- To start the executor:
ndcontrol executor start
- To stop the executor (**AIX, Linux, and Solaris only**):
ndcontrol executor stop

ndcontrol file — manage configuration files



delete

Delete the file.

file[.ext]

A configuration file consisting of ndcontrol 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 Network Dispatcher to the file.

Note: Files are saved into and loaded from the following directories, where *component* is either dispatcher, cbr, or ml (Mailbox Locator):

- AIX: */usr/lpp/nd/servers/configurations/component*
- Linux: */opt/nd/servers/configurations/component*
- Solaris: */opt/nd/servers/configurations/component*
- Windows 2000:

Common install directory path — *c:\Program*

Files\ibm\edge\nd\servers\configurations\component

Native install directory path — *c:\Program*

Files\ibm\nd\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 Network Dispatcher. The new configuration file replaces the current configuration.

Examples

- To delete a file:

```
ndcontrol file delete file3
```

File (file3) was deleted.
- To load a new configuration file to replace the current configuration:

```
ndcontrol file newload file1.sv
```

File (file1.sv) was loaded into the Dispatcher.
- To append a configuration file to the current configuration and load:

```
ndcontrol 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):

```
ndcontrol file report
```

FILE REPORT:
file1.save
file2.sv
file3
- To save your configuration into a file named file3:

```
ndcontrol file save file3
```

The configuration was saved into file (file3).

ndcontrol help — display or print help for this command

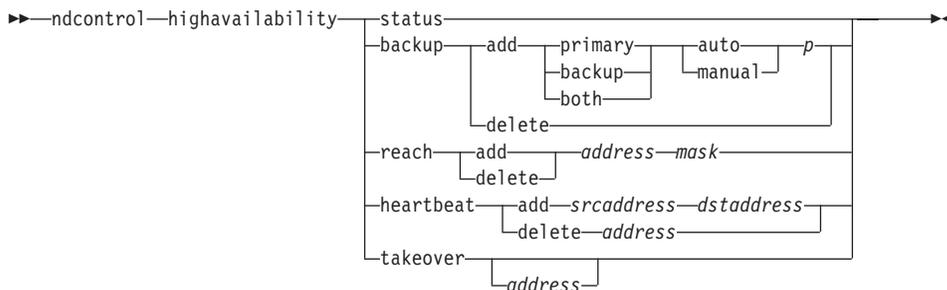
```
▶▶ndcontrol—help—help—▶▶
|host—
|executor—
|manager—
|advisor—
|cluster—
|port—
|rule—
|server—
|subagent—
|highavailability—
|file—
|set—
|status—
|log—
```

Examples

```
fintimeout <cluster address>|all <time>  
-Change FIN timeout  
(Use 'all' to change all clusters)
```

ndcontrol highavailability — control high availability

Note: The ndcontrol high availability syntax diagram does not apply to CBR or Mailbox Locator.



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[port]

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 (dotted-decimal 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 155.

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:

```
ndcontrol highavailability status
```

Output:

```
High Availability Status:
```

```
-----
```

```
Role .....primary
Recovery Strategy ..... manual
State ..... Active
Sub-state..... Synchronized
Primary host..... 9.67.131.151
Port .....12,345
Preferred Target..... 9.67.134.223
```

```
Heartbeat Status:
```

```
-----
```

```
Count ..... 1
```

```
Reachability Status:
```

```
-----
```

```
Count ..... 1
```

- To add the backup information to the primary machine using the automatic recovery strategy and port 80:

```
ndcontrol highavailability backup add primary auto 80
```

- To add an address that the Dispatcher must be able to reach:

```
ndcontrol 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:

```
ndcontrol highavailability takeover
```

ndcontrol host — configure a remote machine

▶▶ndcontrol—host:—*remote_host*◀◀

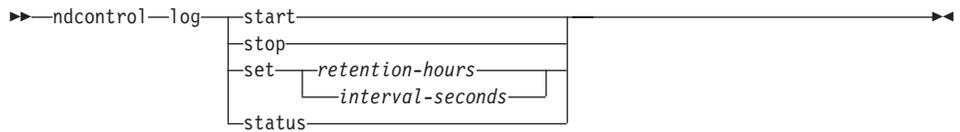
remote_host

The name of the remote Network Dispatcher machine being configured. When typing this command, make sure there is no space between **host:** and *remote_host*, for example:

```
ndcontrol host:remote_host
```

After this command has been issued on the command prompt, enter any valid ndcontrol command you want issued to the remote Network Dispatcher machine.

ndcontrol log — 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 180.

retention

The number of hours that binary log files will be kept. The default value for retention is 24.

hours

The number of hours.

intervals

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.

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.

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 handling for sticky connections” on page 174.

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 138 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 handling for sticky connections” on page 174.

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 dotted-decimal 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 will be time stamped.

The default file will be installed in the **logs** directory. See “Appendix F. Sample configuration files” on page 345. To change the directory where the log files will be kept, see “Changing the log file paths” on page 188.

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:
`ndcontrol manager interval 5`
- To set the level of logging to 0 for better performance:
`ndcontrol manager loglevel 0`
- To set the manager log size to 1,000,000 bytes:
`ndcontrol manager logsize 1000000`
- To specify that no more connections be sent to the server at 130.40.52.153:
`ndcontrol manager quiesce 130.40.52.153`
- To set the number of updating intervals before the weights will be refreshed to 3:
`ndcontrol manager refresh 3`
- To get a statistics snapshot of the manager:
`ndcontrol manager report`

This command produces output similar to:

HOST TABLE LIST	STATUS
9.67.129.221	ACTIVE
9.67.129.213	ACTIVE
9.67.134.223	ACTIVE

9.67.131.18	WEIGHT	ACTIVE % 48	NEW % 48	PORT % 4	SYSTEM % 0					
PORT: 80	NOW	NEW	WT	CONN	WT	CONN	WT	LOAD	WT	LOAD
9.67.129.221	8	8	10	0	10	0	7	29	0	0
9.67.134.223	11	11	10	0	10	0	12	17	0	0
PORT TOTALS:	19	19		0		0		46		0

9.67.131.18	WEIGHT	ACTIVE % 48	NEW % 48	PORT % 4	SYSTEM % 0					
PORT: 23	NOW	NEW	WT	CONN	WT	CONN	WT	LOAD	WT	LOAD
9.67.129.213	10	10	10	0	10	0	10	71	0	0
9.67.134.223	0	0	10	0	10	0	-9999	-1	0	0
PORT TOTALS:	10	10		0		0		70		0

ADVISOR	PORT	TIMEOUT
reach	0	unlimited
http	80	unlimited
ftp	21	unlimited

- To restart all the servers to normalized weights and write a message to the manager log file:
`ndcontrol 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:
`ndcontrol manager sensitivity 10`
- To set the smoothing index to 2.0:
`ndcontrol manager smoothing 2.0`
- To start the manager and specify the log file named `ndmgr.log` (paths cannot be set)
`ndcontrol manager start ndmgr.log`

- To display the current status of the values associated with the manager:
ndcontrol manager status

This command produces output similar to the following example.

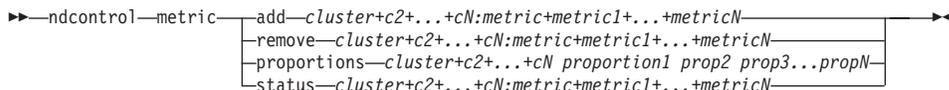
```

Manager status:
=====
Metric port..... 10,004
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

```

- To stop the manager:
ndcontrol 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.):
ndcontrol 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 will be sent to this server until stickytime expires.):
ndcontrol 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:
ndcontrol manager unquiesce 130.40.52.153
- To display the current version number of the manager:
ndcontrol manager version

ndcontrol 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 dotted-decimal IP address. Additional clusters are separated by a plus sign (+).

Note: For Cisco Consultant, the cluster address corresponds to the virtual IP (VIP) address of the content rule of the owner in the Cisco CSS Switch configuration.

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:

```
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:

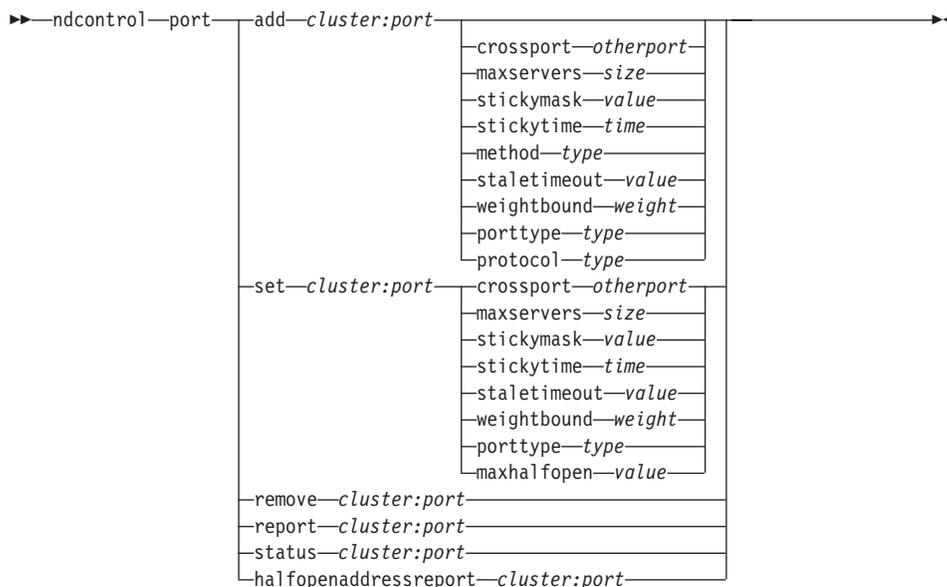
```
sscontrol metric status site1:metric1
```

This command produces output similar to the following:

Metric Status:

```
Cluster ..... 10.10.10.20
Metric name ..... metric1
Metric proportion ..... 50
  Server ..... plm3
  Metric data ..... -1
```

ndcontrol 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 will be processed locally. You can add more than one port at one time using this command.

Note: For the Mailbox Locator component of Network Dispatcher, you must have the cluster IP aliased on the machine before you attempt to add a port. The **add port** command attempts to start a Java proxy that binds to the cluster; therefore, the IP must exist in the IP stack.

On Windows, this means it must be in the Windows Networking setup. The **cluster configure** command is not enough, because it only simulates IP aliasing, and the proxy cannot bind to this fake IP. For all other operating systems, the **cluster configure** command is appropriate because it uses `ifconfig` to alias the IP.

cluster

The address of the cluster as either a symbolic name or in dotted-decimal format. You can use a colon (:) to act as a wild card. For instance, the following command, `ndcontrol 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 172.

Note: Crossport only applies to the Dispatcher component.

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) will be directed to the same server. See “Affinity address mask” on page 173, 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
 - 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) cannot co-exist when stickytime is set on the port.
 - Because setting a stickytime value enables SSL ID affinity, you cannot add a content rule to 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 stickytime value enables IP address affinity.
- The stickytime should be set to 1 if using the Server Directed Affinity API.

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/NAPT forwarding, or Content-based routing forwarding. You may *not* add a forwarding method of NAT/NAPT or content-based routing unless you first specify a nonzero IP address in the clientgateway parameter of the ndcontrol executor command. See "Dispatcher's NAT/NAPT (nat forwarding method)" on page 47 and "Dispatcher's content-based routing (cbr forwarding method)" on page 49 for more information.

Note: If the backend server is on the same subnet as the return address, and if you are using the content-based routing forwarding method or the NAT/NAPT forwarding method, you must define the router address to be the backend server address.

type

The forwarding method type. Possible values are: mac, nat, or cbr. The default is mac (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 or CBR component, the default value is 900 for port 21 (FTP) and 32,000,000 for port 23 (Telnet). For all other ports, the default is 300. Staletimeout can also be set at the executor or cluster level. See “Using stale timeout value” on page 188, for more information.

Note: For Mailbox Locator, staletimeout corresponds to the inactivity autologout timer for these protocols. Staletimeout, for Mailbox Locator, defaults to 60 seconds, which overrides the inactivity timeouts for POP3 and IMAP. For more information on staletimeout for Mailbox Locator, see “Overriding the POP3/IMAP inactivity timer” on page 89.

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 proxy protocol type (POP3 or IMAP). The protocol parameter is required when adding a port for Mailbox Locator.

Note: Protocol only applies to Mailbox Locator.

type

Possible values are **POP3** or **IMAP**.

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 will be 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 178 for more information.

Note: maxhalfopen only applies to Dispatcher.

value

The value of maxhalfopen. The default is zero (no checking will be made).

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. Don't 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 connections. Also, statistical data will be 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 178 for more information.

Examples

- To add port 80 and 23 to a cluster address 130.40.52.153:
ndcontrol port add 130.40.52.153:80+23
- To add a wildcard port to a cluster address of 130.40.52.153:
ndcontrol port set 130.40.52.153:0
- For Mailbox Locator, to add port 20 for POP3 protocol to a cluster address of 9.37.60.91:
mlcontrol port add 9.37.60.91:20 protocol pop3
- To set the maximum weight of 10 to port 80 at a cluster address of 130.40.52.153:

```
ndcontrol 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:

```
ndcontrol 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:

```
ndcontrol port set 130.40.52.153:80 crossport 23
```

- To remove port 23 from a cluster address of 130.40.52.153:

```
ndcontrol port remove 130.40.52.153:23
```

- To get the status of port 80 at a cluster address of 9.67.131.153:

```
ndcontrol port status 9.67.131.153:80
```

This command produces output similar to:

```
Port Status:
```

```
-----
```

```
Port number ..... 80
Cluster address ..... 9.67.131.153
Number of servers ..... 2
Stale timeout ..... 30
Weight bound ..... 20
Maximum number of servers ..... 32
Sticky time ..... 0
Port type ..... tcp/udp
Forwarding method ..... MAC Based Forwarding
Sticky mask bits ..... 32
Cross Port Affinity ..... 80
Max Half Open Connections ..... 0
```

- To get the half open address report for port 80 at a cluster address of 9.67.127.121:

```
ndcontrol port halfopenaddressreport 9.67.127.121:80
```

This command produces output similar to:

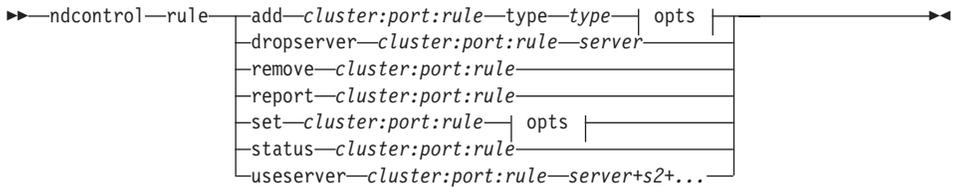
```
Half open connection report successfully created:
```

```
-----
```

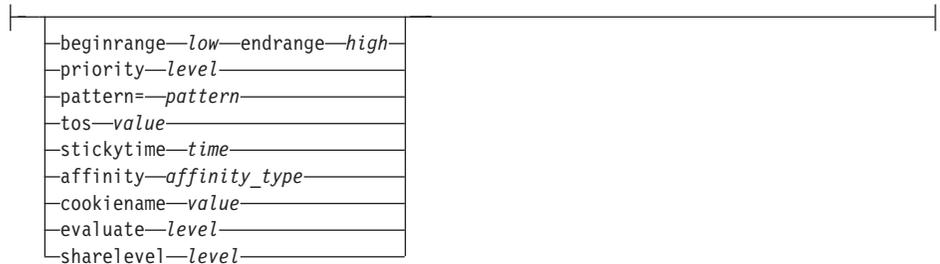
```
Half Open Address Report for cluster:port = 9.67.127.121:80
Total addresses with half open connections reported ... 0
Total number of half open connections reported ..... 0
Largest number of half open connections reported ..... 0
Average number of half open connections reported ..... 0
Average half open connection time (seconds) reported .. 0
Total half open connections received ..... 0
```

ndcontrol rule — configure rules

Note: The rule command syntax diagrams do not apply to Mailbox Locator.



opts:



add

Add this rule to a port.

cluster

The address of the cluster as either a symbolic name or in dotted-decimal format. You can use a colon (:) to act as a wild card. For instance, the following command, `ndcontrol 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, `ndcontrol 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 does not apply to CBR.

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 161 and “Reserved bandwidth rule” on page 162.

Note: Reservedbandwidth only applies to the Dispatcher component.

sharedbandwidth

This rule is based on the amount of bandwidth (kilobytes per second) that will be shared at the executor or cluster level. For more information, see “Using rules based on reserved bandwidth and shared bandwidth” on page 161 and “Shared bandwidth rule” on page 162.

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 which 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 dotted-decimal 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 dotted-decimal 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 will be 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 C. Content rule (pattern) syntax" on page 285.

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 (don't care). For example: 0xx1010x. For more information, see "Using rules based on type of service (TOS)" on page 161.

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 175 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 Network Dispatcher.

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 175, "Passive cookie affinity" on page 177, and "URI affinity" on page 177 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 Network Dispatcher. It is the name that Network Dispatcher should look for in the client HTTP header request. The cookie name, along with the cookie value, acts as an identifier to Network Dispatcher allowing Network Dispatcher 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 177 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 165.

level

Possible values are port or rule. The default is port.

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 162.

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 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 138 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:

```
ndcontrol 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:”

```
ndcontrol 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.:

```
ndcontrol rule add cluster1:80:timerule type time beginrange 11 endrange 14  
ndcontrol rule useserver cluster1:80:timerule server05
```

- To create a rule based on the content of the TOS byte field in the IP header:

```
ndcontrol 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:

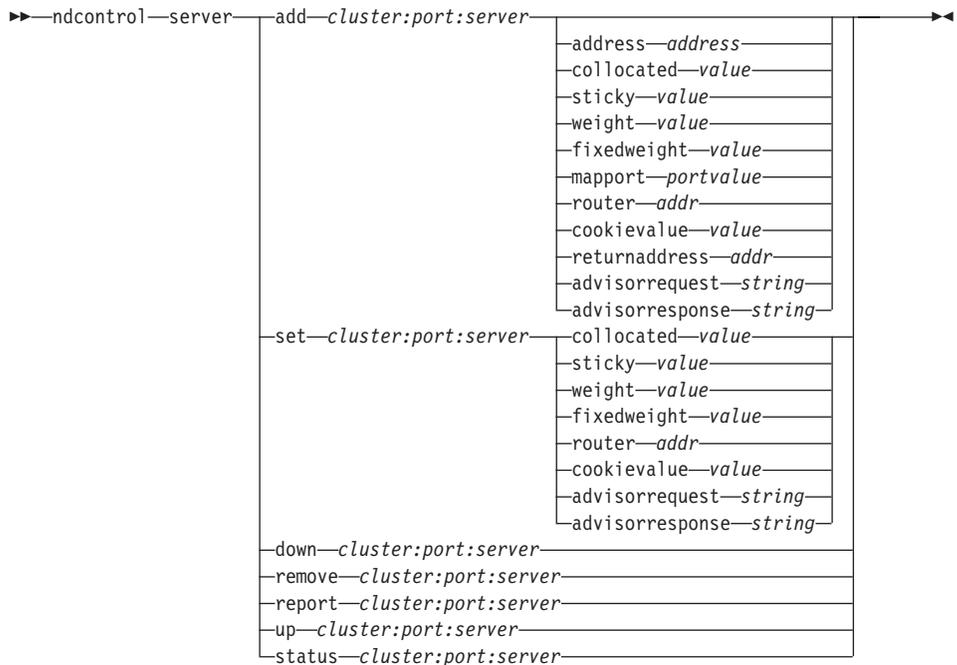
```
ndcontrol rule add 9.67.131.153:80:rbwrule type reservedbandwidth  
beginrange 0 endrange 100 evaluate rule
```

- 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 `ndcontrol cluster` command):

```
ndcontrol cluster set 9.67.131.153 sharedbandwidth 200
```

```
ndcontrol rule add 9.67.131.153:80:shbwrule type sharedbandwidth  
sharelevel cluster
```

ndcontrol server — configure servers



add

Add this server.

cluster

The address of the cluster as either a symbolic name or in dotted-decimal format. You can use a colon (:) to act as a wild card. For instance, the following command, `ndcontrol 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, `ndcontrol 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 dotted-decimal 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 **ndcontrol server add** command. See “Server Partitioning: logical servers configured to one physical server (IP address)” on page 138 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 dotted-decimal 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 138 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. The collocated option does not apply to the Windows 2000 platform.

Note: Collocated parameter is only valid when using the Dispatcher’s mac or nat forwarding methods. Mailbox Locator, Site Selector, and Cisco Consultant can be collocated on all platforms but do not require this keyword. For more information, see “Using collocated servers” on page 140.

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 “Rule affinity override” on page 174.

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 124.

value

Value of fixedweight: yes or no. Default is no.

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 Network Dispatcher 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 47 and “Dispatcher’s content-based routing (cbr forwarding method)” on page 49 . For CBR, see “Load balancing client-to-proxy in SSL and proxy-to-server in HTTP” on page 74.

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 once 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 142 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.

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 Network Dispatcher to send subsequent client requests to the same server. See “Passive cookie affinity” on page 177 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.

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. If you are using nat or cbr forwarding methods, when you add a server to the configuration you must specify the returnaddress.

addr

Value of the return address.

advisorrequest

The HTTP 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 advisor. You must start the HTTP advisor in order for this value to be enabled. See “HTTP advisor request/response (URL) option” on page 140 for more information.

Note: Advisorrequest applies to Dispatcher and CBR components.

string

Value of the string used by the HTTP advisor. The default is HEAD / HTTP/1.0.

Note: If a blank is contained within the string —

- When issuing the command from the **ndcontrol>>** shell prompt, you must place quotes around the string. For example: **server set cluster:port:server advisorrequest "head / http/2.0"**
- When issuing the **ndcontrol** command from the operating system prompt, you must precede the text with **"\"** and follow the text with **\""**. For example: **ndcontrol server set cluster:port:server advisorrequest "\"head / http/2.0\""**

advisorresponse

The advisor response string that the HTTP advisor scans for in the HTTP response. It will only be valid for servers that are advised upon by the HTTP advisor. You must start the HTTP advisor in order for this value to be enabled. See “HTTP advisor request/response (URL) option” on page 140 for more information.

Note: Advisorresponse applies to Dispatcher and CBR components.

string

Value of the string used by the HTTP advisor. The default is null.

Note: If a blank is contained within the string —

- When issuing the command from the **ndcontrol>>** shell prompt, you must place quotes around the string.
- When issuing the **ndcontrol** 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.

remove

Remove this server.

report

Report on this server.

set

Set values for this server.

status

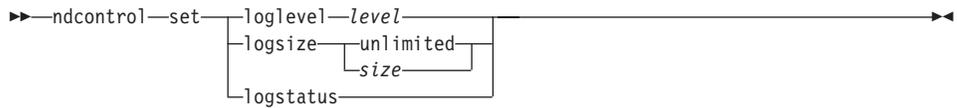
Show status of the servers.

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:
`ndcontrol server add 130.40.52.153:80:27.65.89.42`
- To set the server at 27.65.89.42 as nonsticky (rule affinity override feature):
`ndcontrol server set 130.40.52.153:80:27.65.89.42 sticky no`
- To mark the server at 27.65.89.42 as down:
`ndcontrol 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:
`ndcontrol 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 Network Dispatcher):
`ndcontrol 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:
`ndcontrol server set 130.40.52.153:80:27.65.89.42 weight 10`
- To mark the server at 27.65.89.42 as up:
`ndcontrol server up 130.40.52.153:80:27.65.89.42`
- To add a remote server:
`ndcontrol 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/2.0 for server 27.65.89.42 on HTTP port 80.:
`ndcontrol server set 130.40.52.153:80:27.65.89.42 advisorrequest "\"HEAD / HTTP/2.0\""`

ndcontrol set — configure server log



loglevel

The level at which the ndserver 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.

logstatus

Displays the server log settings (logging level and log size).

ndcontrol status — display whether the manager and advisors are running

▶▶—ndcontrol—status—▶▶

Examples

- To see what is running:

```
ndcontrol status
```

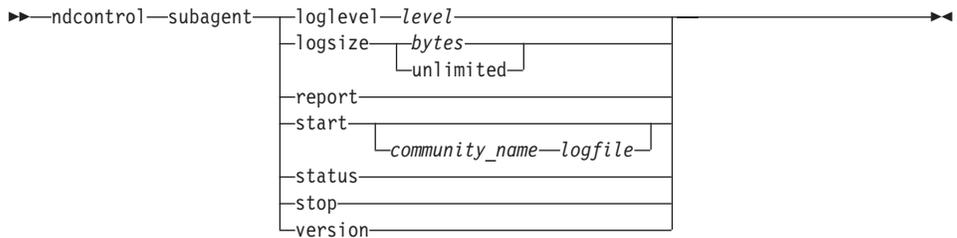
This command produces output similar to:

```
Executor has been started.  
Manager has been started.
```

ADVISOR	PORT	TIMEOUT
reach	0	unlimited
http	80	unlimited
ftp	21	unlimited

ndcontrol subagent — configure SNMP subagent

Note: Ndcontrol subagent command syntax diagrams do not apply to CBR or Mailbox Locator.



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 will be 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.

log file

File name to which the SNMP subagent data is logged. Each record in the log will be time stamped. The default is subagent.log. The default file will be installed in the **logs** directory. See “Appendix F. Sample configuration files” on page 345. To change the directory where the log files will be kept, see “Changing the log file paths” on page 188.

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:
`ndcontrol subagent start bigguy bigguy.log`

Appendix C. 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, using 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 paren used for logic grouping
 -) right paren 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

Version

specific version of request, either HTTP/1.0 or HTTP/1.1

Host value from the host: header.

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.

For example, the following command is valid only when using the **cbrcontrol>>** prompt.

```
rule add 10.1.203.4:80:cbr_prod_rule_ek type content
  pattern client=181.0.153.222&uri=http://10.1.203.4/nipoek/*
```

When using special characters, 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 "client=181.0.153.222&uri=http://10.1.203.4/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.

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 Network Dispatcher, 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 D. Command reference for Site Selector

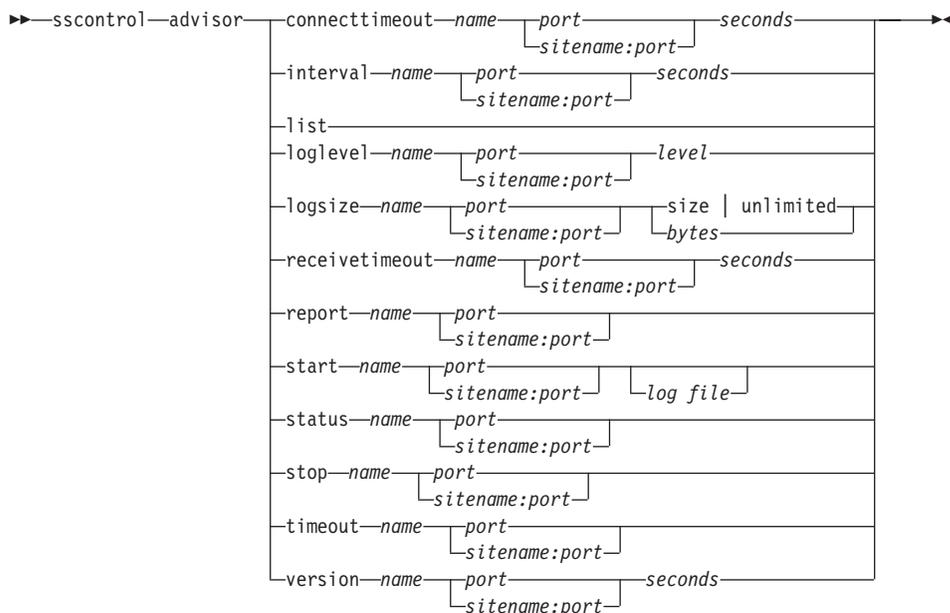
This appendix describes how to use the following Site Selector **sscontrol** commands:

- “sscontrol advisor — control the advisor” on page 290
- “sscontrol file — manage configuration files” on page 295
- “sscontrol help — display or print help for this command” on page 297
- “sscontrol manager — control the manager” on page 298
- “sscontrol metric — configure system metrics” on page 303
- “sscontrol nameserver — control the NameServer” on page 304
- “sscontrol rule — configure rules” on page 305
- “sscontrol server — configure servers” on page 308
- “sscontrol set — configure server log” on page 310
- “sscontrol sitename — configure a sitename” on page 311
- “sscontrol status — display whether the manager and advisors are running” on page 314

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 129.

name

The name of the advisor. Possible values include **http**, **ftp**, **ssl**, **smtp**, **imap**, **pop3**, **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 129.

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.

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
imap	IMAP	143
nntp	NNTP	119
PING	PING	0
pop3	POP3	110
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 188.

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

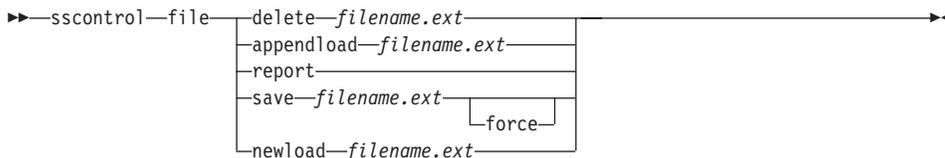
This command produces output similar to the following:

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

- 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: `/usr/lpp/nd/servers/configurations/ss`
- Linux: `/opt/nd/servers/configurations/ss`
- Solaris: `/opt/nd/servers/configurations/ss`
- Windows 2000:

Common install directory path — `c:\Program`

`Files\ibm\edge\nd\servers\configurations\component`

Native install directory path — `c:\Program`

`Files\ibm\nd\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
```

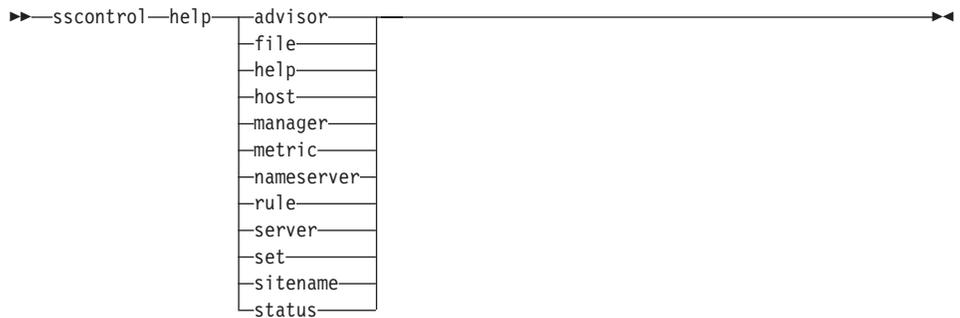
```
file3
```

- 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
```

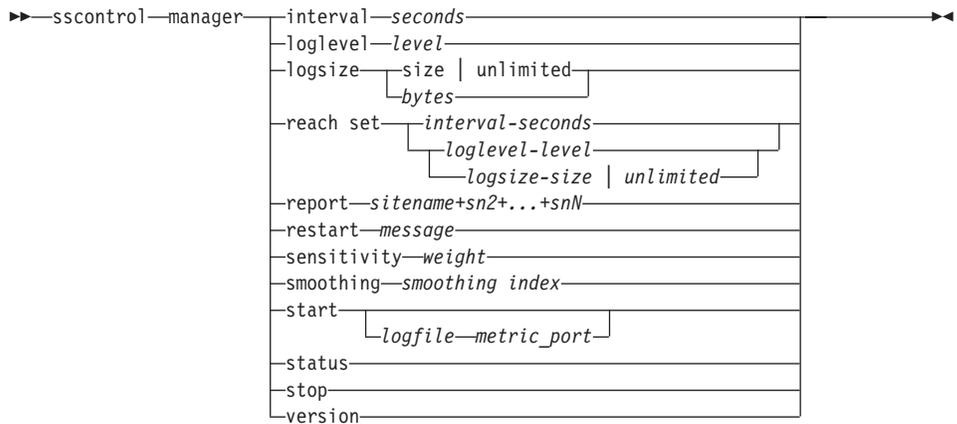
Parameters within < > are variables.

- 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 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 and the metric monitor 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.

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 sitemames 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 F. Sample configuration files” on page 345. To change the directory where the log files will be kept, see “Changing the log file paths” on page 188.

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.

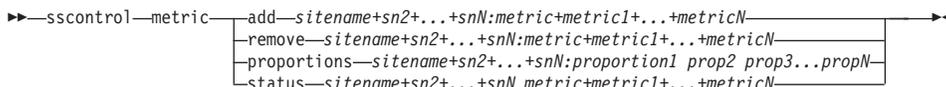
Manager status:

=====

```
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 sitemames 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.

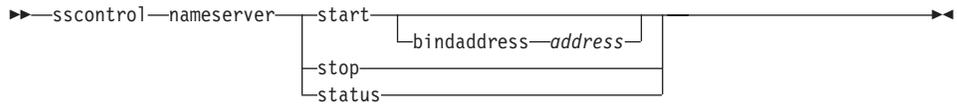
Examples

- To add a system metric:
`sscontrol metric add sitel:metric1`
- To set proportions for a sitename with two system metrics:
`sscontrol metric proportions sitel 0 100`
- To display the current status of values associated with the specified metric:
`sscontrol metric status sitel:metric1`

This command produces output similar to the following:

```
Metric Status:
-----
sitename ..... sitel
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 box.

stop

Stops the name server.

status

Displays the status of the name server.

sscontrol rule — configure rules

```
► sscontrol rule add —sitename+sn2+...+snN:rule+r2+...+rN —type—value— value | opts |  
dropserver—sitename+sn2+...+snN:rule+r2+...+rN—server+s2+...+snN—  
remove—sitename+sn2+...+snN:rule+r2+...+rN—  
set—sitename+sn2+...+snN:rule+r2+...+rN— value | opts |  
status—sitename+sn2+...+snN:rule+r2+...+rN—  
useserver—sitename+sn2+...+snN:rule+r2+...+rN—server+s2+...+snN—
```

opts:

```
|  
|—beginrange—low—endrange—high—|  
|—priority—value—|  
|—metricname—value—|  
|
```

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 dotted-decimal 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 dotted-decimal 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 will be 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 dotted-decimal 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 dotted-decimal 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 dotted-decimal 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

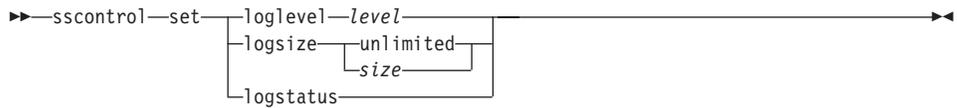
Show status of the servers.

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 sitemames:
`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.

logstatus

Displays the server log settings (logging level and log size).

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 100 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.

tfl 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 100 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

▶—sscontrol—status—▶

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 |  
-----
```

Appendix E. Command reference for Consultant for Cisco CSS Switches

This appendix describes how to use the following **lbcontrol** commands for Consultant for Cisco CSS Switches:

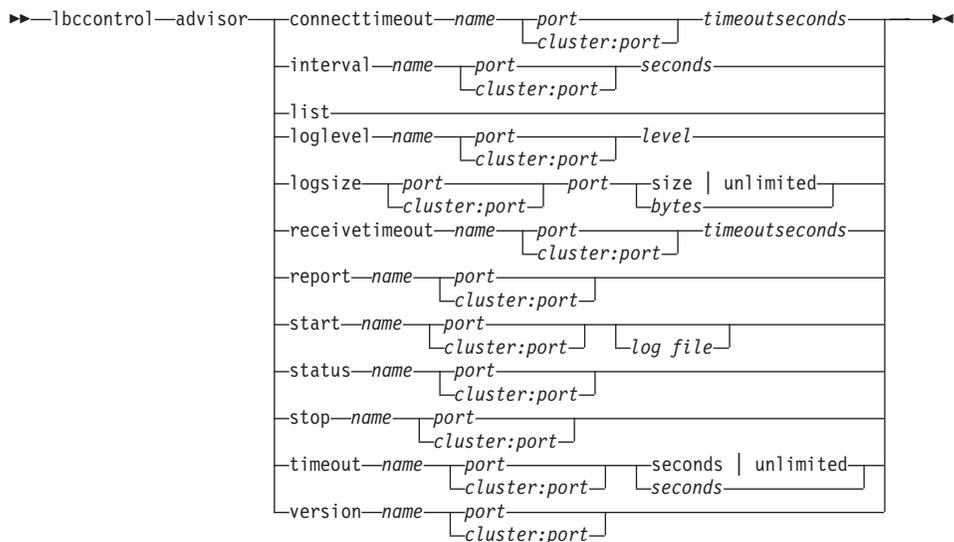
- “lbcontrol advisor — control the advisor” on page 316
- “lbcontrol cluster — configure clusters” on page 321
- “lbcontrol executor — control the executor” on page 323
- “lbcontrol file — manage configuration files” on page 325
- “lbcontrol help — display or print help for this command” on page 327
- “lbcontrol host — configure a remote machine” on page 328
- “lbcontrol log — control the binary log file” on page 329
- “lbcontrol manager — control the manager” on page 330
- “lbcontrol metric — configure system metrics” on page 336
- “lbcontrol port — configure ports” on page 338
- “lbcontrol server — configure servers” on page 340
- “lbcontrol set — configure server log” on page 342
- “lbcontrol status — display whether the manager and advisors are running” on page 343

You can enter a minimized version of the **lbcontrol** 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 **lbcontrol h e f** instead of **lbcontrol help file**.

The “lbc” prefix means load-balancing consultant.

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).

lbcontrol 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 129.

name

The name of the advisor. Possible values include **http**, **ftp**, **ssl**, **smtp**, **imap**, **pop3**, **nntp**, **telnet**, **connect**, **ping**, 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.

timeoutseconds

A positive integer representing the time in seconds that 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 queries 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 15.

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 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.

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 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 129.

timeoutseconds

A positive integer representing the time in seconds that 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.

start

Start the advisor. There are advisors for each protocol. The default ports are as follows:

Advisor Name	Protocol	Port
connect	ICMP	12345
db2	private	50000

Advisor Name	Protocol	Port
ftp	FTP	21
http	HTTP	80
ibmproxy	HTTP (via Caching Proxy)	80
imap	IMAP	143
nntp	NNTP	119
ping	PING	0
pop3	POP3	110
smtp	SMTP	25
ssl	SSL	443
telnet	Telnet	23
WLM	private	10007

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 will be time—stamped.

The default file is *advisorname_port.log*, for example, **http_80.log**. To change the directory where the log files will be kept, see “Changing the log file paths” on page 188.

Set the manager proportions to ensure that the advisor information is used.

status

Display the current status of all the advisor global values and their defaults.

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 time 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 informs 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 the word **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:

```
lbcontrol advisor connecttimeout http 80 30
```

- To set the interval for the FTP advisor (for port 21) to 6 seconds:

```
lbcontrol advisor interval ftp 21 6
```

- To display the list of advisors currently providing information to the manager:

```
lbcontrol advisor list
```

This command produces output similar to:

```
-----
```

ADVISOR	PORT	TIMEOUT
http	80	unlimited
ftp	21	unlimited

```
-----
```

- To change the log level of the advisor log to 0 for better performance:

```
lbcontrol advisor loglevel http 80 0
```

- To change the advisor log size to 5000 bytes:

```
lbcontrol 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:

```
lbcontrol advisor receivetimeout http 80 60
```

- To display a report on the state of the ftp advisor (for port 21):

```
lbcontrol 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 start the advisor with the ftpadv.log file:
lbccontrol advisor start ftp 21 ftpadv.log
- To display the current status of values associated with the http advisor:
lbccontrol advisor status http 80

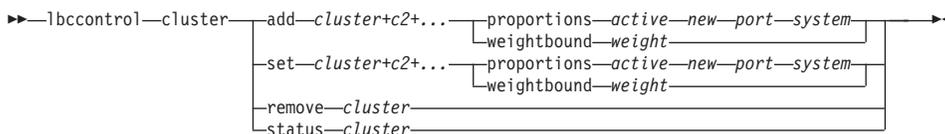
This command produces output similar to the following:

Advisor Status:

```
Interval (seconds) ..... 15
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
```

- To stop the http advisor at port 80:
lbccontrol advisor stop http 80
- To set the timeout value for advisor information to 5 seconds:
lbccontrol advisor timeout ftp 21 5
- To find out the current version number of the ssl advisor:
lbccontrol advisor version ssl 443

lbcontrol cluster — configure clusters



add

Add this cluster. You must define at least one cluster.

weightbound

Set the maximum weight for servers on this port. This affects how much difference there can be between the number of requests the Cisco CSS Switch will give each server. The default value is 10.

weight

The value of weightbound.

set

Set the properties of the cluster.

proportions

Set the proportion of importance for active connections (*active*), new connections (*new*), information from any advisors (*port*), and information from the Metric Server (*system*), which is 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 122.

active

A number from 0–100 representing the proportion of weight given to the active connections. The default is 50.

new

A number from 0–100 representing the proportion of weight given to the new connections. The default is 50.

port

A number from 0–100 representing the proportion of weight given to the information from advisors. The default is 0.

system

A number from 0–100 representing the proportion of weight given to the information from the system metrics. The default is 0.

remove

Remove this cluster.

status

Show current status of a specific cluster.

Examples

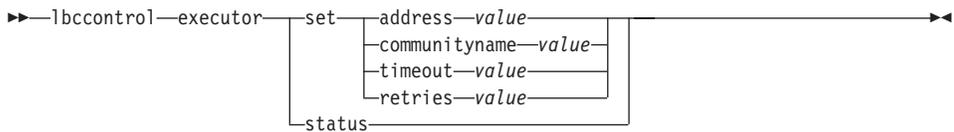
- To add cluster address 130.40.52.153:
`lbcontrol cluster add 130.40.52.153`
- To remove cluster address 130.40.52.153:
`lbcontrol cluster remove 130.40.52.153`
- To set the relative importance placed on input received by the manager:
`lbcontrol cluster proportions 60 35 5 0`
- To show the status for cluster address 9.67.131.167:
`lbcontrol cluster status 9.67.131.167`

This command produces output similar to

Cluster Status:

```
Address ..... 9.67.131.167
Number of target ports ..... 3
Default port weight bound ..... 10
Proportion given to active connections .. 49
Proportion given to new connections .... 49
Proportion given specific to the port ... 2
Proportion given to system metrics ..... 0
```

lbcontrol executor — control the executor



set

Set the fields of the executor.

address

The IP address or hostname used to contact the Cisco CSS Switch for administrative purposes. For more information, refer to the *Cisco Content Services Switch Basic Configuration Guide*.

value

Valid IP address or host name.

communityname

The SNMP community name used in SNMP communications with the Cisco CSS Switch. For more information, refer to the *Cisco Content Services Switch Basic Configuration Guide*.

value

Default is public with read-write access.

timeout

The number of seconds after which SNMP queries from Cisco Consultant to the Cisco CSS Switch will time out. Cisco Consultant uses SNMP to gather information from the Cisco CSS Switch. If manager.log messages indicate frequent timeouts, you can adjust this value to compensate.

value

Default is 3.

retries

The number of times Cisco Consultant retries an SNMP query issued to the Cisco CSS Switch. If manager.log messages indicate frequent SNMP query failures, you can adjust this value to compensate.

value

Default is 2.

status

Display the current status of the values in the executor that can be set and their defaults.

Examples

- To display the internal counters for Cisco Consultant:

```
lbcontrol executor status
```

```
Executor Status:
```

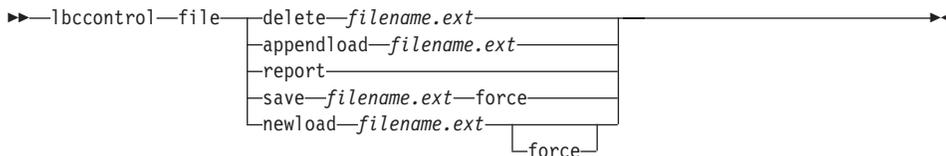
```
-----
```

```
address ..... 9.67.131.151  
community name ..... public  
timeout value ..... 3  
retires value ..... 2
```

- To set the address to 130.40.52.167:

```
lbcontrol executor set address 130.40.52.167
```

lbccontrol file — manage configuration files



delete

Delete the file.

filename.ext

A configuration file.

The file extension (*.ext*) can be anything you choose and is optional

appendload

Append a configuration file to the current configuration and load into Cisco Consultant.

report

Report on the available file or files.

save

Save the current configuration for Cisco Consultant to the file.

Note: Files are saved into and loaded from the following directories:

- AIX: `/usr/lpp/nd/servers/configurations/lbc`
- Linux: `/opt/nd/servers/configurations/lbc`
- Solaris: `/opt/nd/servers/configurations/lbc`
- Windows 2000:

Common install directory path — `c:\Program`

`Files\ibm\edge\nd\servers\configurations\component`

Native install directory path — `c:\Program`

`Files\ibm\nd\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 Cisco Consultant. The new configuration file replaces the current configuration.

Examples

- To delete a file:

```
lbccontrol file delete file3
```

File (file3) was deleted.

- To load a new configuration file to replace the current configuration:

```
lbccontrol file newload file1.sv
```

File (file1.sv) was loaded into the Dispatcher.

- To append a configuration file to the current configuration and load:

```
lbccontrol 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):

```
lbccontrol file report
```

```
FILE REPORT:
```

```
file1.save
```

```
file2.sv
```

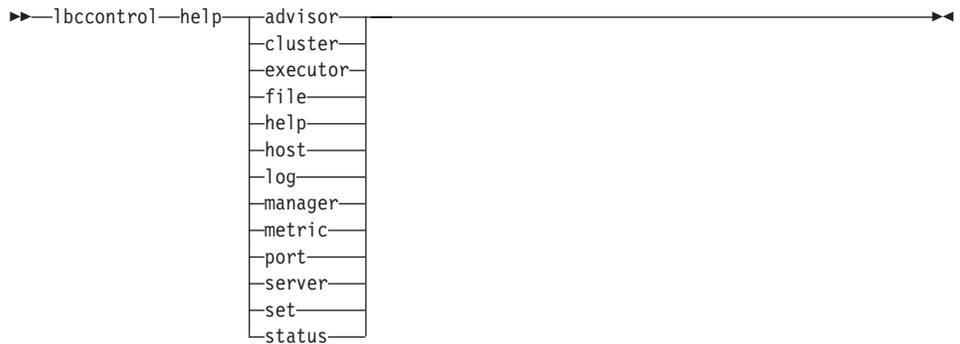
```
file3
```

- To save your configuration into a file named file3:

```
lbccontrol file save file3
```

The configuration was saved into file (file3).

lbcontrol help — display or print help for this command



Examples

- To get help on the lbcontrol command:
lbcontrol help

This command produces output similar to:

```
HELP COMMAND ARGUMENTS:
```

```
-----
```

```
Usage: help <help option>
```

```
Example: help cluster
```

```
executor      - help on executor command  
cluster      - help on cluster command  
port         - help on port command  
server       - help on server command  
manager      - help on manager command  
metric       - help on metric command  
advisor      - help on advisor command  
file         - help on file command  
host         - help on host command  
log          - help on log command  
set          - help on set command  
status       - help on status command  
help         - print complete help text
```

Parameters within < > are variables.

lbcontrol host — configure a remote machine

►►—lbcontrol—host:—*remote_host*—◄◄

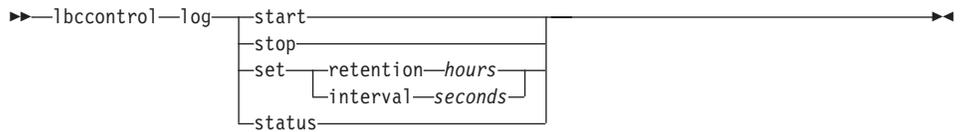
remote_host

The name of the remote Cisco Consultant machine being configured. When typing this command, make sure there is no space between **host:** and *remote_host*, for example:

```
lbcontrol host:remote_host
```

Issue this command at a command prompt, then type any valid lbcontrol command you want issued to the remote Cisco Consultant machine.

lbccontrol log — 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 180.

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.

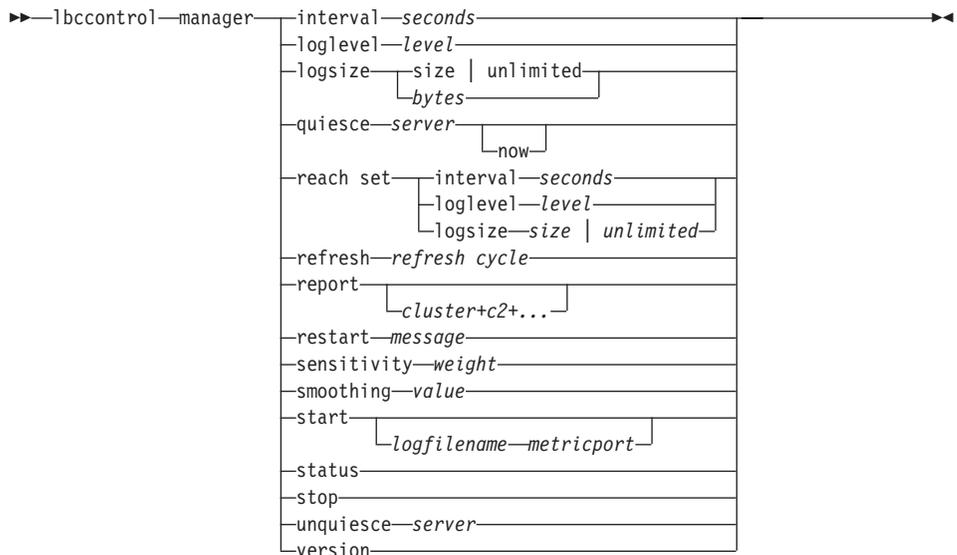
seconds

The number of seconds.

status

Shows the retention and intervals of the binary log.

lbcontrol manager — control the manager



interval

Set how often the manager updates the weights of the servers to the Cisco CSS Switch, updating the criteria that the Cisco CSS Switch uses to route client requests.

seconds

A positive number representing in seconds how often the manager updates weights to the Cisco CSS Switch. The default is 15, with a minimum interval of 10. If you attempt to set the manager interval to less than 10 seconds, the interval is set to 10 seconds. We recommend that you use the default manager interval of 15 seconds because the Cisco CSS Switch does not take advantage of faster updates.

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 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 to display 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 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.

quiesce

Specify that no more connections be sent to a server. The manager sets the weight for that server to 0 in every port to which it is defined, then sends a suspend command to the Cisco CSS Switch. Use this command if you want to quiesce a server for quick maintenance, then make it active again. If you delete a suspended server from the configuration and then add it back, it does not retain its status prior to being suspended.

server

The IP address of the server as either a symbolic name or in dotted-decimal format.

reach

Sets the interval, loglevel, and logsize for the reach advisor.

refresh

Set the number of intervals before querying the Cisco CSS Switch for a refresh of information about new and active connections.

refresh cycle

A positive number representing the number of intervals. The default is 1.

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 dotted-decimal 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 0 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 causes server weights to change less drastically as network conditions change. A lower index will causes server weights to change more drastically.

value

A positive floating point number. The default is 1.5.

start

Start the manager.

logfile

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 F. Sample configuration files” on page 345. See “Changing the log file paths” on page 188 for information on changing the directory where the log files are kept.

metricport

Port on which the metric server will communicate. 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 manager’s global values 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. The manager sends an active command to the Cisco CSS Switch.

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:
`lbcontrol manager interval 5`
- To set the level of logging to 0 for better performance:
`lbcontrol manager loglevel 0`
- To set the manager log size to 1,000,000 bytes:
`lbcontrol manager logsize 1000000`
- To specify that no more connections be sent to the server at 130.40.52.153:
`lbcontrol manager quiesce 130.40.52.153`
- To set the number of updating intervals to 3 before the weights are refreshed:
`lbcontrol manager refresh 3`
- To get a statistics snapshot of the manager:
`lbcontrol manager report`

This command produces output similar to:

```
lbccontrol>>manager report
```

HOST TABLE LIST	STATUS
server6	ACTIVE
server5	ACTIVE
server4	ACTIVE
server3	ACTIVE
server2	ACTIVE
server1	ACTIVE

9.67.154.35	WEIGHT	ACTIVE %	49	NEW %	50	PORT %	1	SYSTEM %	0	
PORT: 80	NOW	NEW	WT	CONNECT	WT	CONNECT	WT	LOAD	WT	LOAD
server1	4	4	5	0	5	0	3	301	-9999	-1
server2	5	5	5	0	5	0	6	160	-9999	-1
PORT TOTALS:	9	9		0		0		461		-2

9.67.154.35	WEIGHT	ACTIVE %	49	NEW %	50	PORT %	1	SYSTEM %	0	
PORT: 443	NOW	NEW	WT	CONNECT	WT	CONNECT	WT	LOAD	WT	LOAD
server3	4	4	5	0	5	0		0	-9999	-1
server4	5	5	5	0	5	0	0	0	-9999	-1
PORT TOTALS:	9	9		0		0		0		-2

9.67.154.34	WEIGHT	ACTIVE %	49	NEW %	50	PORT %	1	SYSTEM %	0	
PORT: 80	NOW	NEW	WT	CONNECT	WT	CONNECT	WT	LOAD	WT	LOAD
server5	5	5	5	0	5	0	5	160	-9999	-1
server6	0	0	5	0	5	0	-9999	-1	-9999	-1
PORT TOTALS:	5	5		0		0		159		-2

ADVISOR	PORT	TIMEOUT
http	80	unlimited

- To restart all the servers to normalized weights and write a message to the manager log file:

lbcontrol 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:
lbcontrol manager sensitivity 10
- To set the smoothing index to 2.0:
lbcontrol manager smoothing 2.0
- To start the manager and specify the log file named ndmgr.log (paths cannot be set)
lbcontrol manager start ndmgr.log
- To display the current status of the values associated with the manager:
lbcontrol manager status

This command produces output similar to the following example.

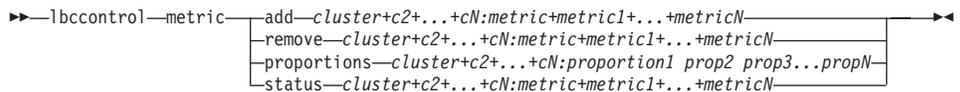
Manager status:

=====

```
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 ..... 1
Reach log level ..... 1
Maximum reach log size (bytes) ..... unlimited
Reach update interval (seconds) ..... 7
```

- To stop the manager:
lbcontrol manager stop
- To display the current version number of the manager:
lbcontrol manager version

lbcontrol metric — configure system metrics



add

Add a metric.

cluster

The address to which clients connect. The address can be either the host name of the machine, or the dotted-decimal IP address. Additional clusters are separated by a plus sign (+).

Note: For Cisco Consultant, the cluster address corresponds to the virtual IP (VIP) address of the content rule of the owner in the Cisco CSS Switch configuration.

metric

The system metric. Your choices for metric are:

- cpuload
- memload
- port
- system metrics

remove

Remove this metric.

proportions

Set the proportions for the metrics associated with this object.

status

Display the current values of this metric.

Examples

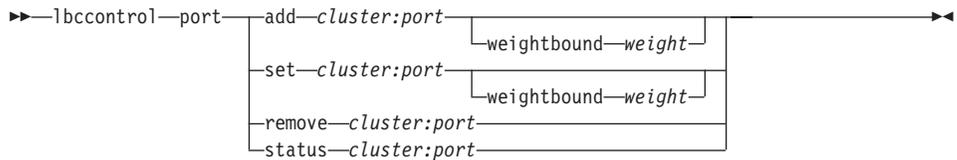
- To add a system metric:
lbcontrol metric add 10.10.10.20:metric1
- To set proportions for a cluster with two system metrics:
lbcontrol metric proportions 10.10.10.20 48 52
- To display the current status of values associated with the specified metric:
lbcontrol metric status 10.10.10.20:metric1

This command produces output similar to the following:

```
Metric Status:
-----
Cluster ..... 10.10.10.20
```

```
Metric name ..... metric1
Metric proportion ..... 52
  Server ..... 9.37.56.100
  Metric data .... -1
```

lbcontrol 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.

weightbound

Set the maximum weight for servers on this port. This affects how much difference there can be between the number of requests the Cisco CSS Switch will give each server. The default value is 10.

weight

A number from 1-10 representing the maximum weight bound.

set

Set the fields of a port.

remove

Remove 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; however, don't forget the colon.

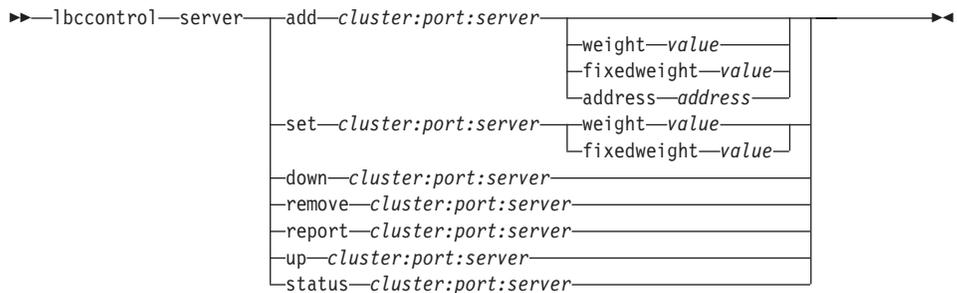
Examples

- To add port 80 and 23 to a cluster address 130.40.52.153:
`lbcontrol port add 130.40.52.153:80+23`
- To set the maximum weight of 10 to port 80 at a cluster address of 130.40.52.153:
`lbcontrol port set 130.40.52.153:80 weightbound 10`
- To remove port 23 from a cluster address of 130.40.52.153:
`lbcontrol port remove 130.40.52.153:23`
- To get the status of port 80 at a cluster address of 9.67.131.153:
`lbcontrol port status 9.67.131.153:80`

This command produces output similar to:

```
Port Status:
-----
Port number ..... 80
Cluster address ..... 9.67.131.153
Number of servers ..... 2
Weight bound ..... 10
```

lbcontrol server — configure servers



add

Add this server.

cluster

The address of the cluster as either a symbolic name or in dotted-decimal format.

Note: Additional clusters are separated by a plus sign (+).

port

The number of the port.

Note: Additional ports are separated by a plus sign (+).

server

The unique IP address of the TCP server machine as either a symbolic name or in dotted-decimal format. If you use a unique symbolic name that does not resolve to an IP address, you must provide the address attribute on the **lbcontrol server add** command.

weight

A number from 0-to-10 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 weight. If the manager is running and fixedweight is set to no, this setting will be quickly overwritten.

value

Value of the weight.

fixedweight

The fixedweight option allows you to specify if you want the manager to modify the server weight. 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 124.

value

Value of the fixed weight. The default is no.

address

The unique IP address of the TCP server machine as either a symbolic name or in dotted-decimal format. If the server name value is unresolved (for example, a logical server name), you must provide the address of the physical server machine.

value

The unique identifier of the server machine. If server is not resolvable, you must provide the address attribute.

down

Mark this server down. The Cisco CSS Switch will stop sending connections to this server.

remove

Remove this server.

report

Report on this server.

set

Set values for this server.

up Mark this server up. The Cisco CSS Switch will now send new connections to this server.

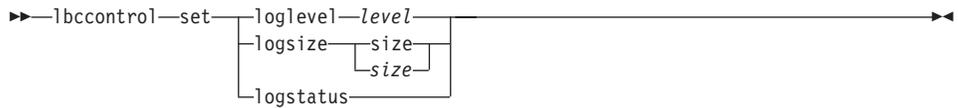
status

Show status of the servers.

Examples

- To add the server at 27.65.89.42 to port 80 on a cluster address 130.40.52.153:
`lbcontrol server add 130.40.52.153:80:27.65.89.42`
- To remove the server at 27.65.89.42 on all ports on all clusters:
`lbcontrol server remove ::27.65.89.42`
- To set the weight to 10 for server 27.65.89.42 at port 80 on cluster address 130.40.52.153:
`lbcontrol server set 130.40.52.153:80:27.65.89.42 weight 10`

lbcontrol set — configure server log



loglevel

The level at which the lbserver logs its activities.

level

The default value of **loglevel** is 1. The range is 0-to-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 logged in the log file.

size

The default value of logsize is 1 MB.

logstatus

Displays the server log settings (logging level and log size).

lbcontrol status — display whether the manager and advisors are running

►►—lbcontrol—status—◀◀

Examples

- To see what is running:

```
lbcontrol status
```

This command produces output similar to:

Manager has been started.

ADVISOR	PORT	TIMEOUT
http	80	unlimited
ftp	21	unlimited

Appendix F. Sample configuration files

This appendix contains sample configuration files for the Dispatcher component of Network Dispatcher.

Sample Network Dispatcher configuration files

Sample files are located in the `.../nd/servers/samples/` directory.

Dispatcher Configuration file—AIX, Red Hat Linux, and Solaris

```
#!/bin/ksh
#
# 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
#
# ndserver start
# sleep 5

#
# Then start the executor
#
# ndcontrol executor start

#
# The Dispatcher can be removed at any time using the
# "ndcontrol executor stop" and "ndserver 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"
# ndcontrol 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 "
# ndcontrol 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 forwarded to the corresponding port of one of the server
# machines.
#

# echo "Creating ports for CLUSTER: $CLUSTER"

# ndcontrol 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"
# ndcontrol 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. IfMae:M885wheovNP85is(n(e:M885
advisors arisorsNNTPrisorsareandowrol85is(:areand# W#:M8MW<ingcomponenechestS"
```

```

# 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.
# ndcontrol cluster 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.
# ndcontrol cluster 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.
# ndcontrol manager loglevel 1
# ndcontrol manager logsize 1048576
# ndcontrol manager sensitivity 5.000000
# ndcontrol manager interval 2
# ndcontrol manager refresh 2
#
# ndcontrol advisor interval ftp 21 5
# ndcontrol advisor loglevel ftp 21 1
# ndcontrol advisor logsize ftp 21 1048576
# ndcontrol advisor timeout ftp 21 unlimited
# ndcontrol advisor interval telnet 23 5
# ndcontrol advisor loglevel telnet 23 1
# ndcontrol advisor logsize telnet 23 1048576
# ndcontrol advisor timeout telnet 23 unlimited
# ndcontrol advisor interval smtp 25 5
# ndcontrol advisor loglevel smtp 25 1
# ndcontrol advisor logsize smtp 25 1048576
# ndcontrol advisor timeout smtp 25 unlimited
# ndcontrol advisor interval http 80 5
# ndcontrol advisor loglevel http 80 1
# ndcontrol advisor logsize http 80 1048576
# ndcontrol advisor timeout http 80 unlimited
# ndcontrol advisor interval pop3 110 5
# ndcontrol advisor loglevel pop3 110 1
# ndcontrol advisor logsize pop3 110 1048576
# ndcontrol advisor timeout pop3 110 unlimited
# ndcontrol advisor interval nntp 119 5
# ndcontrol advisor loglevel nntp 119 1
# ndcontrol advisor logsize nntp 119 1048576
# ndcontrol advisor timeout nntp 119 unlimited
# ndcontrol advisor interval ssl 443 5
# ndcontrol advisor loglevel ssl 443 1
# ndcontrol advisor logsize ssl 443 1048576
# ndcontrol advisor timeout ssl 443 unlimited
#

```

Dispatcher Configuration file—Windows

The following is a sample Network Dispatcher 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 ndserver must be started via Services

rem

rem
rem Then start the executor
rem
rem call ndcontrol 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 ndcontrol 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 ndcontrol executor set fintimeout 30
rem call ndcontrol executor set fincount 4000
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 ndcontrol 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 ndcontrol 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 ndcontrol 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 via
rem ndcontrol 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 ndcontrol manager start

rem echo "Starting the FTP advisor on port 21 ..."
rem call ndcontrol advisor start ftp 21
rem echo "Starting the HTTP advisor on port 80 ..."
rem call ndcontrol advisor start http 80
rem echo "Starting the Telnet advisor on port 23 ..."
rem call ndcontrol advisor start telnet 23
rem echo "Starting the SMTP advisor on port 25 ..."
rem call ndcontrol advisor start smtp 25
rem echo "Starting the POP3 advisor on port 110 ..."
rem call ndcontrol advisor start pop3 110
rem echo "Starting the NNTP advisor on port 119 ..."
rem call ndcontrol advisor start nntp 119
rem echo "Starting the SSL advisor on port 443 ..."
rem call ndcontrol advisor start ssl 443
rem

rem echo "Setting the cluster proportions..."
rem call ndcontrol 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 ndcontrol cluster 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 ndcontrol cluster 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 ndcontrol manager loglevel 1
rem call ndcontrol manager logsize 1048576
rem call ndcontrol manager sensitivity 5.000000
rem call ndcontrol manager interval 2
rem call ndcontrol manager refresh 2
rem
rem call ndcontrol advisor interval ftp 21 5
rem call ndcontrol advisor loglevel ftp 21 1
rem call ndcontrol advisor logsize ftp 21 1048576
rem call ndcontrol advisor timeout ftp 21 unlimited
rem call ndcontrol advisor interval telnet 23 5
rem call ndcontrol advisor loglevel telnet 23 1
rem call ndcontrol advisor logsize telnet 23 1048576
rem call ndcontrol advisor timeout telnet 23 unlimited
rem call ndcontrol advisor interval smtp 25 5
rem call ndcontrol advisor loglevel smtp 25 1
rem call ndcontrol advisor logsize smtp 25 1048576
rem call ndcontrol advisor timeout smtp 25 unlimited
rem call ndcontrol advisor interval http 80 5
rem call ndcontrol advisor loglevel http 80 1
rem call ndcontrol advisor logsize http 80 1048576
rem call ndcontrol advisor timeout http 80 unlimited
rem call ndcontrol advisor interval pop3 110 5
rem call ndcontrol advisor loglevel pop3 110 1
rem call ndcontrol advisor logsize pop3 110 1048576
rem call ndcontrol advisor timeout pop3 110 unlimited
rem call ndcontrol advisor interval nntp 119 5
rem call ndcontrol advisor loglevel nntp 119 1
rem call ndcontrol advisor logsize nntp 119 1048576
rem call ndcontrol advisor timeout nntp 119 unlimited
rem call ndcontrol advisor interval ssl 443 5
rem call ndcontrol advisor loglevel ssl 443 1
rem call ndcontrol advisor logsize ssl 443 1048576
rem call ndcontrol advisor timeout ssl 443 unlimited
rem

```

Sample advisor

The following is a sample advisor file called **ADV_sample**.

```

/**
 * ADV_sample: The Network Dispatcher HTTP advisor
 *
 *
 * This class defines a sample custom advisor for Network Dispatcher.
 * 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 Network Dispatcher for use in the Network Dispatcher'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 Network Dispatcher
*   Directories:
*
*   nd/servers/lib/CustomAdvisors/
*   (nd\servers\lib\CustomAdvisors on Windows 2000)
*
* - 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 Network Dispatcher, must be compiled with
* the prereq version of Java.
* To ensure access to Network Dispatcher classes, make sure that the
* ibmnd.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):
*
*   - Params
*     - 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 sDefaultLogFileName = 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.
*   - Parm
*     - String sDataString - The data to be sent is sent in the form of a
*       string
*   - Return
*     - int RC - Whether the data was successfully sent or not: zero
*               indicates data was sent; a negative integer indicates an
*               error.
*
* - receive - Receive information from the socket connection.
*   - Parm
*     - 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 Network Dispatcher 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_Network_Dispatcher_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 Network

```

```

*           Dispatcher-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 higher the weight will be within Network Dispatcher
*   regarding load balancing.
*
*   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.
*   Network Dispatcher will not attempt to load balance to a server that
*   is down. Network Dispatcher will resume load balancing to the server
*   when a positive value is received.
*
*   A value of zero is typically not returned; Network Dispatcher handles
*   a load of zero in a special way. Zero is assumed to indicate an
*   unknown status, and Network Dispatcher gives the server a high
*   weight in response.
*/
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);

        // If the receive is successful, a load of zero is returned.
        // This is because the "replace" flag is set to false,
        // indicating that the load built within the base advisor is
        // to be used.
        // Since nothing was done with the returned data, additional
        // load is not necessary.

        // Note: it is known that the advisor base load will not be
        // zero, therefore a zero load will
        // not be returned for use in calculating the weight.
        if (iRc >= 0)
        {
            iLoad = 0;
        }
    }
    return iLoad;
}

```

```
} // End - ADV_sample
```

Appendix G. 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 Network Dispatcher components (the Dispatcher component and the CBR component) along with Caching Proxy.

Server machine set up

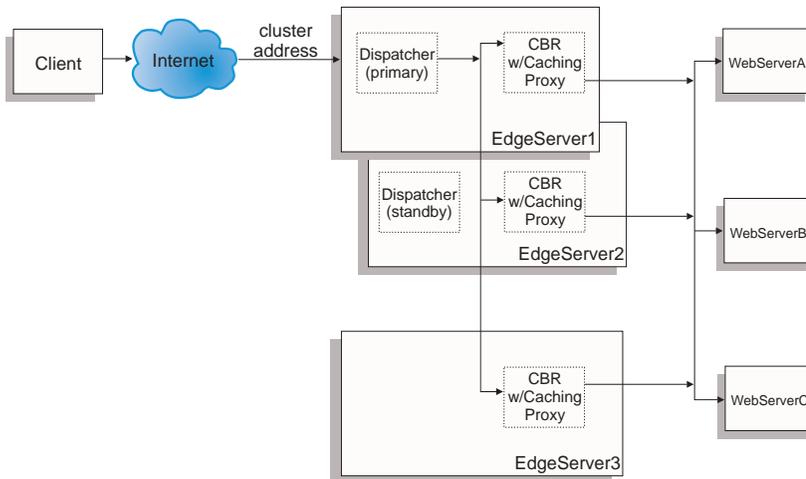


Figure 30. Example of a 2-tier, high availability configuration using Dispatcher, CBR, and Caching Proxy

The server machine set up for Figure 30 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 30 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 Edge

Servers. 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 Edge Servers. In order 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 Edge Servers to cache Web pages that are requested more frequently. For more information on setting up Caching Proxy, refer to the *IBM WebSphere Edge Server for Multiplatforms Administration Guide*.
- Define the cluster address and ports to be the same in both the CBR and Dispatcher components of Network Dispatcher.
- Configure the CBR component to be the same across all Edge Servers. 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 7. Configuring the Content Based Routing component” on page 75.
- Configure the Dispatcher component to be the same on Edge Servers 1 and 2. Define all the Edge Servers 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 5. Configuring the Dispatcher component” on page 53.
- Configure Edge Server 1 as the primary high availability machine and Edge Server 2 as the standby (backup) high availability machine. For more information, see “High availability” on page 150.

Note:

1. The hostname (i.e, www.company.com) associated with the cluster address will need to be updated in the Caching Proxy configuration file for the "Hostname" directive.
2. To avoid the backend server addresses displayed in the URL, you may need to set the "SendRevProxyName" directive to "yes" in the Caching Proxy configuration file.
3. In order 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.
4. In order to cache by the inbound URL name instead of IP address, add an additional line with the Proxy directive under the Mapping Rules section of the Caching Proxy configuration file.

Sample lines referred to in notes 1-4 (above):

```

Hostname          www.company.com
SendRevProxyName  yes
Caching           ON
CacheMemory       128000 K
Proxy             /* http://www.company.com/* www.company.com

```

5. 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 Edge Servers.
6. If using the Linux platform for the Edge Servers, you will need to install a patch to the linux kernel. For more information, see “Installing the Linux kernel patch (to suppress arp responses on the loopback interface)” on page 66.
7. 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 Server configuration as shown in Figure 30 on page 359. The sample configuration files represent the files for the Dispatcher and CBR components of Network Dispatcher. In the sample configuration, a single ethernet adapter is used for each of the Edge Server 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 Edge Server): 192.168.1.10
- EdgeServer2 (Backup high availability Edge Server): 192.168.1.20
- EdgeServer3 (Web caching Edge Server): 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 Edge Server:

```

ndcontrol executor start

ndcontrol cluster add 192.168.1.11 primaryhost 192.168.1.10

ndcontrol port add 192.168.1.11:80

ndcontrol server add 192.168.1.11:80:edgeserver1 address 192.168.1.10
ndcontrol server add 192.168.1.11:80:edgeserver2 address 192.168.1.20
ndcontrol server add 192.168.1.11:80:edgeserver3 address 192.168.1.30

ndcontrol manager start manager.log 10004

```

```
ndcontrol highavailability heartbeat add 192.168.1.10 192.168.1.20
ndcontrol highavailability backup add primary auto 4567
```

Sample Configuration file for CBR component on the Edge Servers:

```
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 H. Other resources

Command line access

In many circumstances, you can use keys or key combinations to perform operations that can also be done through mouse actions. Many menu actions can be initiated from the keyboard.

Consult the documentation for your operating system for instructions on using the keyboard.

Getting online help

Network Dispatcher includes an online help facility, which describes the tasks you will perform while installing, planning, configuring, and operating the product.

To get help for the current window, click the question mark (?) in the upper right corner. Choose from:

Field Help

Context sensitive help for the task you are performing.

How Do I

A list of tasks related to the current window.

Contents

A table of contents for all the help information.

Index An alphabetical index of the help topics.

Reference information

For additional information on using Network Dispatcher, refer to:

- The WebSphere Edge Server Web site at:
<http://www.ibm.com/software/webservers/edgeserver>
- The Network Dispatcher technote Web site at:
<http://www.ibm.com/software/webservers/edgeserver/support.html>
Click on **Search for Network Dispatcher hints and tips**.

Appendix I. Notices

References in this publication to IBM products, programs, or services do not imply that IBM intends to make them available in all countries in which IBM operates. Any reference to an IBM product, program, or service is not intended to state or imply that IBM product, program or service may be used. Subject to IBM's valid intellectual property or other legally protectable rights, any functionally equivalent product, program, or service may be used instead of the IBM product, program, or service. The evaluation and verification of operation in conjunction with other products, except those expressly designated by IBM, are the responsibility of the user.

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 the IBM Director of Licensing, IBM Corporation, North Castle Drive, Armonk, NY 10504-1785, U.S.A.

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:

Site Counsel
IBM Corporation
P.O. Box 12195
3039 Cornwallis Avenue
Research Triangle Park, NC 27709-2195
USA

The licensed program described in this document and all licensed material available for it are provided by IBM under terms of the IBM Customer Agreement.

This document is not intended for production use and is furnished as is without any warranty of any kind, and all warranties are hereby disclaimed including the warranties of merchantability and fitness for a particular purpose.

This product includes computer software created and made available by CERN. This acknowledgement shall be mentioned in full in any product which includes the CERN computer software included herein or parts thereof.

Trademarks

The following terms are registered trademarks or trademarks of IBM Corporation in the United States, other countries or both.

AIX

IBM

IBMLink

LoadLeveler

OS/2

NetView

WebSphere

Lotus is a registered trademark of Lotus Development Corporation in the United States, other countries or both.

Domino is a trademark of Lotus Development Corporation in the United States, other countries or both.

Tivoli is a registered trademark of Tivoli Systems, Inc in the United States, other countries, or both.

Java and all Java-based trademarks and logos are trademarks or registered trademarks of Sun Microsystems, Inc. in the United States, other countries or both.

Solaris is a trademark of Sun Microsystems, Inc. in the United States, other countries, or both.

Microsoft and Windows 2000 are trademarks or registered trademarks of the Microsoft Corporation in the United States, other countries, or both.

Cisco is a registered trademark of Cisco Systems, Inc. in the United States and certain other countries.

HP is a trademark of the Hewlett-Packard Company in the United States, other countries, or both.

Linux is a registered trademark of Linus Torvalds.

Red Hat is a registered trademark of Red Hat, Inc.

UNIX is a registered trademark of The Open Group in the United States and other countries.

Other company, product, and service names, which may be denoted by a double asterisk (**), 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 IP address is a 32-bit address field. This field contains two parts. The first part is the network address; the second part is the host number.

advisor. The advisors are a function of the Network Dispatcher. 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 Network Dispatcher. 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 Network Dispatcher.

cbrserver. In Content Based Router, handles the requests from the executor, manager and advisors.

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 Consultant. A component of IBM Network Dispatcher. Cisco Consultant uses Network Dispatcher 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. Network Dispatcher balances TCP or UDP traffic among these clustered servers.

collocate. When you don't have a dedicated machine, Dispatcher is installed on the same machine it is load balancing.

Note: Collocated only applies to the AIX, Red Hat Linux, and Solaris operating systems.

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 Network Dispatcher 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 IP addresses.

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 Dispatcher 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. In Cisco Consultant, the executor holds configuration information and contains the information required to connect to the Cisco CSS Switch.

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. Once a transaction is in FIN state, the Network Dispatcher 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 Dispatcher machines in high availability mode used by the standby Dispatcher to monitor the health of the active Dispatcher.

high availability. A Dispatcher feature in which one Dispatcher 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.

I

Internet. The worldwide collection of interconnected networks that use the Internet suite of protocols and permit public access.

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.

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 32-bit 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.

lbc. load-balancing Consultant

lbccontrol. In Cisco Consultant, provides the interface to the Cisco CSS Switch.

lbcserver. In Cisco Consultant, contains the configuration information and performs the commands.

loopback alias. An alternative IP address associated with the loopback interface. The alternative address has the useful side effect 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. A LAN or a LAN emulation concept.

Mailbox Locator. A component of Network Dispatcher. For IMAP or POP3 protocols, Mailbox Locator is a proxy that chooses an appropriate server based on user ID and password.

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 Network Dispatcher 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 Server. Formerly known as Server Monitor Agent (SMA). Metric server provides system specific metrics to the Network Dispatcher 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.

mlcontrol. Provides the interface to the Mailbox Locator component of Network Dispatcher.

mlserver. In Mailbox Locator, contains the configuration information and performs the commands.

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

ndcontrol. Provides the interface to the Dispatcher component of Network Dispatcher.

ndserver. In Dispatcher, handles the requests from the command line to the executor, manager, and advisors.

netmask. For Internet subnetworking, a 32-bit mask used to identify the subnetwork address bits in the host portion of an IP address.

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 executes 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.

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.

nonforwarding address (nfa). The primary IP address of the Network Dispatcher machine, used for administration and configuration.

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. A standard IP address is a 32-bit address field. The server address can be either the dotted decimal 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. A function provided by one or more nodes; for example, HTTP, FTP, Telnet.

shell. The software that accepts and processes command lines from a user's workstation. The Korn 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 Network Dispatcher. 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.

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).

sscontrol. Provides the interface to the Site Selector component of Network Dispatcher.

ssserver. In Site Selector, handles the requests from the command line to the site name, manager and advisors.

strategy. In high availability for the Dispatcher, a keyword for specifying how recovery takes place following the failure of the active machine.

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.

subnet mask. For Internet subnetworking, 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 Network Dispatcher links with other servers into a single, virtual server. Network Dispatcher 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

A

- accessibility 363
- active connections 182
- active cookie affinity 175, 271
- adding
 - cluster 237, 322
 - port to a cluster 60, 265, 338
 - server to a port 60, 279, 309, 341
- address mapping file
 - example of 168
- advisors
 - cbrcontrol 228
 - Cisco Consultant
 - display status 318, 320
 - interval for 316, 319
 - list of 317
 - name of 316
 - port for 316
 - report on the state of 319
 - report timeout 318, 320
 - server connect timeout 316, 319
 - server receive timeout 317, 319
 - starting 317, 320
 - stopping 318, 320
 - version of 319, 320
 - Dispatcher component 126
 - Caching Proxy advisor 130
 - customize 131
 - fast-failure detection 129
 - interval for 128, 231
 - list of 129, 231
 - name of 228
 - port for 235
 - report 232
 - report on the state of 232
 - report timeout 128, 231
 - self advisor 131, 149
 - server connect timeout 129, 228, 231
 - server receive timeout 129, 229, 232
 - ssl2http advisor 74, 130
 - starting 61, 231
 - starting/stopping 127
 - stopping 231
 - version of 232
 - advisors (*continued*)
 - HTTP advisor
 - request/response 140
 - lbcontrol 316
 - limitation on Linux 127
 - list of 230, 317
 - mlcontrol 228
 - ndcontrol 228
 - sample configuration files 352
 - Site Selector
 - interval 290
 - interval for 293
 - list 291
 - list of 292, 293
 - loglevel 291
 - name of 290
 - port for 228, 290
 - report on the state of 291, 293
 - report timeout 292, 294
 - server connect timeout 290, 293
 - server receive timeout 291, 293
 - starting 292, 294
 - stopping 292, 294
 - version of 293, 294
 - sscontrol 290, 297
 - starting 117
 - URL option, HTTP advisor 140
 - advisors, Network Dispatcher
 - component
 - list of 319
 - report on the state of 317
 - starting 61
 - affinity (sticky)
 - active cookie 175, 271
 - affinity address mask 173
 - cross port affinity 172, 173, 262
 - how it works 170
 - mailbox locator 89
 - passive cookie 175, 177, 271
 - quiesce now 174, 255, 259
 - rule affinity override 174
 - rule option 175
 - SDA (Server Directed Affinity) 171
 - SSL ID (cbr forwarding) 50, 51

- affinity (sticky) (*continued*)
 - sticky (rule affinity override) 174, 275
 - stickymask 172, 173, 262
 - stickytime 50, 51, 170, 172, 263, 270
 - URI 175, 177, 271
 - affinity address mask 173, 262
 - AIX
 - installing 13
 - requirements 12
 - alias
 - loopback device 62
 - Linux kernel patch 62, 66
 - the NIC 58, 83
 - apCntsvcHits 182
 - apSvcConnections 182
- ## B
- backup, high availability 45, 248
 - configuring 151
 - binary logging for server
 - statistics 180, 188
 - bind-specific servers 60, 61, 127, 145
- ## C
- Caching Proxy 73
 - configure for CBR 80
 - Caching Proxy advisor 130
 - CBR
 - alias the NIC 83
 - cbrcontrol fails 214
 - cbrcontrol fails on Solaris 215
 - configuration
 - overview of tasks 75
 - setting up the CBR machine 79
 - hardware and software requirements 71
 - ifconfig command 83
 - load-balancing settings 122
 - ndadmin fails 214
 - planning 71
 - requests not being load balanced 214
 - starting and stopping 196
 - syntactical or configuration error 215
 - troubleshooting table 201

- CBR *(continued)*
 - using Dispatcher component 49
 - will not run 214
 - with Caching Proxy
 - configuring 85
 - mapport keyword 74
 - overview 72
 - SSL connections 73
 - ssl2http advisor 74
- cbr forwarding method 49
 - stickytime 50, 51
- cbrcontrol command
 - advisor 228
 - cluster 234
 - executor 239
 - file 244
 - help 246
 - host 252
 - log 253
 - manager 254
 - metric 260
 - port 261
 - rule 267
 - server 274
 - set 280
 - status 281
- changing
 - FIN count 189
 - FIN time-out 189
 - stale timer 189
- checking for
 - extra route 64
- Cisco Consultant
 - cannot create registry on port 14099 218
 - commands 315
 - configuration
 - example 41
 - overview of tasks 113
 - setting up the CSS
 - machine 116
 - executor 109
 - hardware and software requirements 109
 - lbcontrol 110
 - lbcontrol fails 218
 - lbcsrvr 109
 - load-balancing setting
 - advisor server timeout 316
 - load-balancing settings
 - advisor report timeout 318, 320
 - advisor server timeout 317, 319
 - manager 109
- Cisco Consultant *(continued)*
 - ndadmin 110
 - ndadmin fails 218
 - planning 109
 - starting 197
 - starting and stopping 197
 - troubleshooting table 203
 - using 197
 - will not start 218
- cluster
 - adding 237, 322
 - cbrcontrol 234
 - changing the FIN count for 189
 - changing the FIN time-out for 189
 - configure the address 58
 - define 116
 - defining 58, 237, 322
 - displaying
 - status of this cluster 238, 322
 - lbcontrol 321
 - mlcontrol 234
 - ndcontrol 234
 - proportions 234
 - removing 237, 313, 321, 322
 - set proportions 61, 117
 - wildcard 58
- cluster-specific
 - proportions 311
- collocate, Network Dispatcher and server 56, 61, 140, 145, 275, 279
- collocated (keyword) 141, 279
- command line
 - access 363
 - configuration example 3
- command references
 - how to read 221
- commands
 - cbrcontrol
 - advisor 228
 - cluster 234
 - executor 239
 - file 244
 - help 246
 - host 252
 - log 253
 - manager 254
 - metric 260
 - port 261
 - rule 267
 - server 274
 - set 280
 - status 281
 - Cisco Consultant 315
 - ifconfig 60, 145
- commands *(continued)*
 - to alias the loopback device 62
- lbcontrol
 - advisor 316
 - cluster 321
 - executor 323
 - file 325
 - help 327
 - host 328
 - log 329
 - manager 330
 - metric 336
 - port 338
 - servers, configure 340
 - set 342
 - status 343
- mlcontrol
 - advisor 228
 - cluster 234
 - executor 239
 - file 244
 - help 246
 - host 252
 - log 253
 - manager 254
 - metric 260
 - port 261
 - server 274
 - set 280
 - status 281
- ndconfig 60, 145
- ndcontrol
 - advisor 228
 - cluster 234
 - executor 239
 - file 244
 - help 246
 - high availability, control 248
 - host 252
 - log 253
 - manager 254
 - metric 260
 - port 261
 - prompt 226
 - rule 267
 - server 274
 - set 280
 - status 281
 - subagent, configure
 - SNMP 282
 - to control the advisor 61
 - to control the manager 61
 - to define a port 60
 - to define a server 60

- commands (*continued*)
 - ndcontrol (*continued*)
 - to define the nonforwarding address 58, 242, 323, 324
 - netstat
 - to check IP addresses and aliases 64
 - route
 - to delete an extra route 64, 65
 - Site Selector 289
 - sscontrol
 - advisor 290
 - file 295
 - help 297
 - manager 298
 - metric 303
 - nameserver 304
 - rule 305
 - server 308
 - set 310
 - sitename 311
 - status 314
 - configuration
 - Cisco Consultant 113
 - Content Based Routing 75
 - define cluster 116
 - defining load-balanced servers 117
 - Dispatcher component 53
 - Mailbox Locator 91
 - mapping between Consultant and CSS 110
 - methods
 - command line (CBR) 76
 - command line (Cisco Consultant) 114
 - command line (Dispatcher) 53
 - command line (Mailbox Locator) 92
 - command line (Site Selector) 104
 - GUI (CBR) 78
 - GUI (Cisco Consultant) 115
 - GUI (Dispatcher) 54
 - GUI (Mailbox Locator) 93
 - GUI (Site Selector) 105
 - scripts (CBR) 77
 - scripts (Cisco Consultant) 114
 - scripts (Dispatcher) 54
 - scripts (Mailbox Locator) 92
 - scripts (Site Selector) 104
 - wizard (CBR) 79
 - configuration (*continued*)
 - methods (*continued*)
 - wizard (Dispatcher) 55
 - wizard (Mailbox Locator) 94
 - wizard (Site Selector) 106
 - Metric Server 118
 - port 116
 - sample files 345
 - set cluster proportions 117
 - Site Selector 103
 - starting the manager 117
 - tasks, advanced 119
 - testing 118
 - verify 65
 - wizard 4
 - connections, setting proportion of importance 123, 237
 - connecttimeout
 - Cisco Consultant 316
 - Site Selector 290
 - Content Based Routing 27
 - configuration
 - overview of tasks 75
 - setting up the CBR machine 79
 - hardware and software requirements 71
 - load-balancing settings 122
 - planning 71
 - troubleshooting table 201
 - using 195
 - using Dispatcher component 49
 - content rule 49, 165
 - cross port affinity 172, 262
 - custom (customizable) advisor 131
- D**
- DB2 advisor 131
 - default.cfg 58, 82, 95, 106
 - defining
 - cluster 237, 322
 - nonforwarding address 58, 242, 323, 324
 - port to a cluster 60, 265, 338
 - server to a port 60, 279, 309, 341
 - deleting
 - cluster 237, 313, 321, 322
 - extra route 65
 - port from a cluster 266, 338
 - server from a port 279, 308, 309, 341
 - Denial of service attack
 - detection 178
 - halfopenaddressreport 265
 - maxhalfopen 264
 - diagnosing problems
 - advisors not working 208
 - advisors show that all servers are down 212
 - blue screen displays when starting Network Dispatcher executor 211
 - cannot create registry on port 14099 218
 - CBR will not run 214
 - cbrcontrol fails on Solaris 215
 - cbrcontrol or ndadmin command fails 214
 - cbrserver command is stopped 215
 - common problems and solutions 207, 209, 214, 215, 216, 218
 - Dispatcher, Microsoft IIS, and SSL do not work 208
 - Dispatcher and server will not respond 207
 - Dispatcher high availability not working 208
 - Dispatcher requests not routed 207
 - Dispatcher will not run 207
 - error message when trying to view online Help 210
 - error running Dispatcher with Caching Proxy installed 210
 - extra routes 208
 - GUI does not display correctly 210
 - GUI does not start correctly 210
 - help panels disappear 211
 - high availability in the wide area mode of Network Dispatcher does not work 213
 - lbcontrol or ndadmin command fails 218
 - lbserver will not start 218
 - Mailbox Locator will not run 215
 - Metric Server IOException on Windows 2000 218
 - Metric Server log reports "Signature is necessary for access to agent" 219
 - Metric Server not reporting loads 219
 - mlcontrol or ndadmin command fails 216
 - ndcontrol or ndadmin command fails 209

- diagnosing problems (*continued*)
 - Network Dispatcher cannot process and forward a frame 211
 - Path to Discovery prevents return traffic with Network Dispatcher 211
 - port numbers used by CBR 205
 - port numbers used by Cisco Consultant 206
 - port numbers used by Mailbox Locator 205
 - port numbers used by Site Selector 206
 - port numbers used by the Dispatcher 204
 - receive Mailbox Locator error when trying to add a port 216
 - requests not being load balanced 214
 - Site Selector doesn't load-balance correctly 217
 - Site Selector doesn't round-robin (Solaris) 217
 - Site Selector will not run 216
 - SNMPD not working 208
 - spurious error message when starting ndserver on Solaris 2.7 210
 - sscontrol or ndadmin command fails 217
 - sssserver failing to start on Windows 2000 217
 - Syntactical or configuration error 215
 - unable to add a port 216
 - Unable to add heartbeat 208
 - unexpected behavior loading large configuration file 213
- disability 363
- Dispatcher
 - configuration
 - setting up the TCP server machines 62
- Dispatcher component
 - advisors not working 208
 - advisors show that all servers are down 212
 - blue screen displays when starting executor 211
 - cannot forward a frame 211
 - cannot open help window 210
 - configuration
 - overview of tasks 53
- Dispatcher component (*continued*)
 - configuration (*continued*)
 - setting up a private network 167
 - setting up the Network Dispatcher machine 56
 - connection to a remote machine 209
 - content-based routing 49
 - error starting ndserver on Solaris 2.7 210
 - error when caching proxy is installed 210
 - extra routes (Windows 2000) 208
 - GUI does not display correctly 210
 - GUI not starting correctly 210
 - hardware and software requirements 43
 - help windows disappear 211
 - high availability in the wide area mode of Network Dispatcher does not work 213
 - high availability is not working 208
 - load-balancing settings 122
 - advisor intervals 128
 - advisor report timeout 128
 - advisor server timeout 129
 - manager intervals 124
 - proportion of importance given to status information 122
 - sensitivity threshold 125
 - smoothing index 125
 - weights 123
 - MAC forwarding 47
 - MS IIS and SSL do not work 208
 - NAT/ NAPT 47
 - ndadmin fails 209
 - ndcontrol fails 209
 - Path to Discovery prevents return traffic with Network Dispatcher 211
 - planning 43
 - requests not being balanced 207
 - server will not respond 207
 - SNMPD not working 208
 - starting 188
 - troubleshooting table 199
 - unable to add heartbeat 208
 - unexpected behavior loading large configuration file 213
- Dispatcher component (*continued*)
 - using 188
 - will not run 207
- displaying
 - global values and their default settings
 - for an advisor 232, 292, 294
 - for the manager 259, 300, 301, 332, 335
 - internal counters 242, 323
 - list of
 - advisors currently providing metrics 231, 293, 319
 - report on the state of an advisor 232, 291, 293, 317
 - statistics report 257, 299, 300, 331, 333
 - status of
 - a cluster or all clusters 238, 322
 - servers on a port 266, 338
 - version number
 - of advisor 232, 293, 294
 - of manager 259, 300, 302, 332, 335
- down, marking a server as 279, 308, 309
- DPID2 191
- E**
- Ethernet NIC
 - ibmnd.conf
 - configuring for Solaris 56
- examples
 - managing local servers 33, 34, 36, 37, 39, 41
 - quick start 1
- executor
 - cbrcontrol 239
 - lbcccontrol 323
 - mlcontrol 239
 - ndcontrol 239
 - starting 243, 323
 - stopping 243
- explicit linking 166
- extra routes 64, 65
- F**
- field help 363
- file
 - cbrcontrol 244
 - lbcccontrol 325
 - mlcontrol 244
 - ndcontrol 244
 - sscontrol 295

- FIN count 189
- FIN count limit
 - changing 189
- FIN time out
 - changing 189
- Firewall 22
- forwarding method
 - cbr 49
 - mac 47, 48
 - mac, nat or cbr 263
 - MAC, NAT or cbr 50
 - NAT 47
- ftp advisor 228, 290

G

- garbage collection 189
- goActive 155
- goIdle 156
- goInOp 156
- goStandby 156
- graphical user interface (GUI) 5
- GRE (Generic Routing Encapsulation)
 - OS/390 148
 - wide area support 148
- GUI 5
 - resolution 210

H

- hardware requirements
 - CBR 71
 - Cisco Consultant 109
 - Dispatcher component 43
 - Mailbox Locator 87
 - Site Selector 97
- help
 - cbrcontrol 246
 - lbcontrol 327
 - mlcontrol 246
 - ndcontrol 246
- help, online 363
- high availability 27, 42, 45, 150
 - configuring 151
 - mutual 46, 152, 236, 238, 250
 - ndcontrol 248
 - primaryhost 236, 238
 - scripts 155
 - goActive 155
 - goIdle 156
 - goInOp 156
 - goStandby 156
 - highavailChange 156
- highavailChange 156
- host
 - cbrcontrol 252

- host (*continued*)
 - lbcontrol 328
 - mlcontrol 252
 - ndcontrol 252
- how do I 363
- http advisor 228, 290

I

- ibmnd.conf
 - configuring for Solaris 56
- ibmproxy 73, 80
 - advisor 130
- ifconfig command 60, 62, 83, 145
- imap
 - overriding 89
- installing
 - Network Dispatcher 11
 - on AIX 13
 - on Linux 17
 - on Solaris 19
 - on Windows 2000 22
- interval, setting how often
 - the advisor queries the servers 231, 293, 316, 319
 - the manager queries the executor 125, 257, 331, 333
 - the manager updates the weights to the executor 124, 257, 298, 300, 330, 333

J

- Java runtime environment (JRE) 13, 17, 19

K

- keyboard 363
- keys
 - ndkeys 136, 186

L

- lbcontrol command
 - advisor 316
 - cluster 321
 - executor 323
 - file 325
 - help 327
 - host 328
 - log 329
 - manager 330
 - metric 336
 - port 338
 - server 340
 - set 342
 - status 343
- lbserver
 - will not start 206, 218

- Linux
 - installing 17
 - kernel patch
 - versions 2.2.12, 2.2.13 68
 - versions 2.4.x 67
 - requirements 16
- load-balancing settings (optimizing) 122
- log
 - binary, for server statistics 180, 253, 329
 - cbrcontrol 253
 - file, setting the name of
 - for the advisor 292, 318
 - for the manager 300, 332
 - lbcontrol 329
 - level, setting
 - for the advisor 187, 232, 293, 317, 319
 - for the manager 187, 298, 330
 - for the server 187
 - for the subagent 187
 - mlcontrol 253
 - ndcontrol 253
 - size, setting
 - for the advisor 187, 232, 291, 293, 317, 319
 - for the manager 187, 257, 298, 300, 330, 333
 - for the server 187
 - for the subagent 187
 - using CBR logs 196
 - using Cisco Consultant logs 198
 - using Mailbox Locator logs 197
 - using Metric Server logs 198
 - using Network Dispatcher logs 187
 - using Site Selector logs 197
- logon/logoff 11

M

- mac forwarding method 47
- Mailbox Locator
 - configuration
 - overview of tasks 91
 - setting up the machine 94
 - hardware and software requirements 87
 - inactivity timeout 236, 240, 264
 - load-balancing settings 122
 - mlcontrol fails 216
 - mlserver 88
 - mlserver command is stopped 215

- Mailbox Locator *(continued)*
 - ndadmin fails 216
 - overview 88
 - planning 87
 - proxy error trying to add port 216
 - proxy protocol 264, 265
 - staletimeout 236, 240, 264
 - starting and stopping 196
 - troubleshooting table 202
 - unable to add a port 216
 - using 196
 - will not run 215
- manager
 - cbrcontrol 254
 - fixed weight 124
 - lbcontrol 330
 - mlcontrol 254
 - ndcontrol 254
 - proportions 122, 321
 - sscontrol 298
 - starting 61, 117, 258, 299, 301, 332, 335
 - stopping 259, 300, 302, 332, 335
 - version of 259, 300, 302, 332, 335
- managing Network Dispatcher 185
- mapping between Consultant and CSS 110
- marking a server as being
 - down 279, 308, 309
 - up 279, 309
- maximum weight, setting for servers on a specific port 124, 265, 338
- metric
 - cbrcontrol 260
 - lbcontrol 336
 - mlcontrol 260
 - ndcontrol 260
 - sscontrol 303
- Metric Server
 - Metric Server IOException on Windows 2000 218
 - Metric Server log reports
 - "Signature is necessary for access to agent" 219
 - Metric Server not reporting loads 219
 - overview 136
 - start 118
 - starting and stopping 198
 - troubleshooting table 203
 - using 198
- migrating 11
- mlcontrol command
 - advisor 228
 - cluster 234
 - executor 239
 - file 244
 - help 246
 - host 252
 - log 253
 - manager 254
 - metric 260
 - port 261
 - server 274
 - set 280
 - status 281
- Monitor menu option 190
- multiple address collocation 61
- mutual high availability 46, 151, 152
 - primaryhost 236, 238
 - scripts 155
 - takeover 155
- N**
- nameserver
 - sscontrol 304
- NAT forwarding method 47
- ndconfig 145
 - command 60
- ndcontrol command
 - advisor 61, 228
 - cluster 234
 - command prompt 226
 - executor 58, 239
 - file 244
 - help 246
 - highavailability 248
 - host 252
 - log 253
 - manager 61, 254
 - metric 260
 - minimize command
 - parameters 225
 - port 60, 261
 - rule 267
 - server 60, 274
 - set 280
 - status 281
 - subagent 282
- ndkeys 137, 186
- ndserver
 - starting 3
- netstat command 64
- network address port translation (NAPT) 47
- network address translation (NAT) 47
- Network Dispatcher
 - benefits 26
 - cnfiguring
 - CBR 75
 - Cisco Consultant 113
 - Mailbox Locator 91
 - configuration tasks, advanced 119
 - configuring
 - Dispatcher component 56, 79, 94, 106
 - Site Selector 103
 - functions 25, 32
 - hardware requirements 43, 71, 87, 97, 109
 - installing 11
 - operating and managing 185, 197
 - overview 25, 32
 - planning considerations 43, 97
 - quick start example 1
 - software requirements 43, 71, 87, 97, 109
 - troubleshooting 199
- network proximity 100
- new connections 182
- new connections, setting proportion of importance 122, 235, 321
- new features, V2.0
 - AIX v5.1 Support 27
 - CBR usability improvements 29
 - Cisco Consultant 28
 - Cluster Specific Proportions 30
 - DB2 Advisor 32
 - Denial of Service detection 31
 - Dispatcher's content-based routing 30
 - Enhanced User Exits 31
 - HTTP Advisor Req/Rsp 31
 - Linux and Solaris NLS 28
 - Mailbox Locator 29
 - Metric Server 29
 - NAT and NAPT 29
 - New Chinese NLS Standard Support 28
 - Passive cookie affinity 30
 - Red Hat Linux v7.1 Support 28
 - Server Partitioning 31
 - Site (Cluster) Specific Advisors 31
 - Site Selector 28
 - SuSE Linux v7.1 Support 28
 - URI affinity 30

- NIC
 - alias 58
 - ethernet (for Solaris) 56
 - mapping (for Windows 2000) 59
- nonforwarding address
 - defining 58
 - setting 242, 323, 324
- notices 365
- O**
- online help 363
- operating Network Dispatcher 185
- OS/390
 - GRE support 148
- overview
 - configuration of CBR 75
 - configuration of Cisco Consultant 113
 - configuration of Dispatcher component 53
 - configuration of Mailbox Locator 91
 - configuration of Site Selector 103
- P**
- passive cookie affinity 175, 177, 271
- planning
 - CBR 71
 - Cisco Consultant 109
 - Dispatcher component 43
 - Mailbox Locator 87
 - Site Selector 97
- planning for installation 25, 43, 97
- pop3
 - overriding 89
- port
 - cbrcontrol 261
 - configuration 116
 - lbcontrol 338
 - mlcontrol 261
 - ndcontrol 261
- ports
 - adding 265, 338
 - defining to a cluster 60, 265, 338
 - displaying
 - status of servers on this
 - port 266, 338
 - for advisors 228, 290
 - removing 266, 338
 - setting the maximum weight 124, 265, 338
 - wildcard 60
- primaryhost 152, 238
- private key
 - for remote authentication 185
- private network, using with Dispatcher 167
- product components 43
- proportion of importance for load balancing, setting 122, 237
- proportions 117
- proximity options 100
- public key
 - for remote authentication 185
- Q**
- quick start example 1
- quiescing a server 174, 255, 257, 259, 333
- R**
- remote administration 20, 185
- remove
 - cluster 237, 313, 321, 322
 - extra route 65
 - port from a cluster 266, 338
 - server from a port 279, 308, 309, 341
- requirements
 - AIX 12
 - Linux 16
 - Solaris 19
 - Windows 2000 21
- resolution, GUI 210
- resources 363
- restart all servers to normalized weights 258, 299, 301, 331, 334
- RMI (Remote Method Invocation) 185
- route command 64, 65
- routes, delete extra 65
- routes, extra 64
- rule
 - cbrcontrol 267
 - ndcontrol 267
 - sscontrol 305
- rule affinity override
 - server 174, 275, 279
- rules-based load balancing 157
 - active connections to port 160, 268
 - always true 164, 268, 272, 305, 307
 - choice of rules, by component 157
 - client IP address 159, 268, 273, 305, 307
 - client port 161, 268
- rules-based load balancing
 - (continued)*
 - connections per second 160, 268
 - content of request 49, 165, 269
 - evaluate option 165
 - metric all 163
 - metric average 164
 - metricall 305
 - metricavg 305
 - reserved bandwidth 161, 162, 268, 273
 - server evaluate option 165
 - shared bandwidth 161, 162, 268, 273
 - time of day 159, 268, 273, 305, 307
 - type of service (TOS) 161, 268, 273
- S**
- sample configuration files 345
 - advisor 352
 - Dispatcher component (AIX) 345
 - Dispatcher component (Windows) 349
- scripts 155
 - goActive 155
 - goIdle 156
 - goInOp 156
 - goStandby 156
 - highavailChange 156
 - user exit 126
- SDA (Server Directed Affinity) 139, 171
- Secure Sockets Layer 60
- sensitivity to weights update, setting 125, 258, 299, 301, 332, 335
- server
 - adding 279, 309, 341
 - address 275, 341
 - advisorrequest 277
 - advisorresponse 278
 - cbrcontrol 274
 - collocated 275, 279
 - cookievalue 277
 - defining to a port 60, 279, 309, 341
 - fixedweight 276
 - lbcontrol 340
 - logical 138
 - mapport 74, 276
 - marking as being down 279, 308, 309
 - marking as being up 279, 309

- server (*continued*)
 - mlcontrol 274
 - ndcontrol 274
 - nonsticky (rule affinity override) 275, 279
 - partitioning 138
 - physically 138
 - quiescing 174, 255, 257, 259, 333
 - removing 279, 308, 309, 341
 - restarting all to normalized weights 258, 299, 301, 331, 334
 - returnaddress 277
 - router 276
 - setting the weight 279, 309, 341
 - sscontrol 308
 - unquiescing 259, 332
 - weight 275
- Server Directed Affinity (SDA) 139, 171
- set
 - cbrcontrol 280
 - lbcontrol 342
 - mlcontrol 280
 - ndcontrol 280
 - sscontrol 310
- setting
 - cluster address 60
 - how often the manager should query the executor 125, 257, 331, 333
 - interval time
 - for the advisor to query the servers 231, 293, 316, 319
 - for the manager to update the executor 124, 257, 298, 300, 330, 333
 - logging level
 - for the advisor 187, 232, 293, 317, 319
 - for the manager 298, 330
 - maximum size of the log
 - for the advisor 187, 232, 291, 293, 317, 319
 - for the manager 257, 298, 300, 330, 333
 - maximum weight
 - for servers on a specific port 124, 265, 338
 - name of log file 292, 318
 - for the manager 300, 332
 - nonforwarding address 56
 - proportion of importance in load balancing 237
- setting (*continued*)
 - sensitivity to weights
 - update 125, 258, 299, 301, 332, 335
 - smoothing index 126, 258, 299, 301, 332, 335
 - weight for a server 257, 259, 279, 309, 332, 333, 341
- settings, displaying all global values
 - for an advisor 232, 292, 294
 - for the manager 259, 300, 301, 332, 335
- showing
 - global values and their default settings
 - for an advisor 232, 292, 294
 - for the manager 259, 300, 301, 332, 335
 - internal counters 242, 323
 - list of
 - advisors currently providing metrics 231, 293, 319
 - report on the state of an advisor 232, 291, 293, 317
 - statistics report 257, 299, 300, 331, 333
 - status of
 - a cluster or all clusters 238, 322
 - servers on a port 266, 338
 - version number
 - of advisor 232, 293, 294
 - of manager 259, 300, 302, 332, 335
- Simple Network Management Protocol (SNMP) 190
- Site Selector
 - commands 289
 - configuration
 - overview of tasks 103
 - setting up the machine 106
 - configuration example 39
 - hardware and software requirements 97
 - load-balancing HA Dispatchers 157
 - load-balancing settings 122
 - ndadmin fails 217
 - not load balancing correctly with duplicate routes 217
 - overview 38
 - planning 97
 - sscontrol fails 217
 - ssserver failing to start on Windows 2000 217
- Site Selector (*continued*)
 - starting and stopping 197
 - troubleshooting table 203
 - using 197
 - will not round-robin traffic from Solaris clients 217
 - will not run 216
 - sitename
 - sscontrol 311
 - smoothing index, setting 126, 258, 299, 301, 332, 335
 - SNMP 187, 190
 - software requirements
 - CBR 71
 - Cisco Consultant 109
 - Dispatcher component 43
 - Mailbox Locator 87
 - Site Selector 97
 - Solaris
 - apr publish command 60
 - installing 19
 - requirements 19
 - setting up Dispatcher machine 56
 - sscontrol command
 - advisor 290
 - file 295
 - help 297
 - manager 298
 - metric 303
 - nameserver 304
 - rule 305
 - server 308
 - set 310
 - sitename 311
 - status 314
 - SSL 60
 - SSL connections
 - advisor 130
 - configuring ibmproxy 73
 - for CBR 73, 74
 - problem with enabling 208
 - ssl2http advisor 74, 130
 - stale timeout 188, 236, 240, 264
 - starting
 - advisor 61, 231, 292, 294
 - Cisco Consultant 197
 - Dispatcher 3
 - executor 58, 243, 323
 - manager 61, 258, 299, 301, 332, 335
 - Metric Server 198
 - server 57, 58
 - Site Selector 197

- starting and stopping
 - CBR 196
 - Dispatcher 188
 - Mailbox Locator 196
- statistics snapshot report, displaying 257, 299, 300, 331, 333
- status
 - cbrcontrol 281
 - lbcontrol 343
 - mlcontrol 281
 - ndcontrol 281
- status, displaying
 - all cluster 322
 - one cluster 322
 - servers on a specific port 266, 338
- sticky (affinity)
 - active cookie 175, 271
 - affinity address mask 173
 - cross port affinity 172, 173, 262
 - how it works 170
 - passive cookie 175, 177, 271
 - quiesce now 174, 255, 259
 - rule affinity override 174
 - SDA (Server Directed Affinity) 171
 - sticky (rule affinity override) 174, 275
 - stickymask 172, 173, 262
 - stickytime 50, 51, 170, 172, 263, 270
 - URI 175, 271
- stopping
 - advisor 231, 292, 294
 - Cisco Consultant 197
 - executor 243
 - manager 259, 300, 302, 332, 335
- subagents 187, 190
- ndcontrol 282
- syntax diagrams
 - examples 222
 - parameters 221
 - punctuation 221
 - reading 221
 - symbols 221
- system metrics
 - configure 260, 303, 336
 - setting proportion of importance 123, 234, 235, 321

T

- testing
 - configuration 118
- trademarks 366
- troubleshooting 199

- troubleshooting (*continued*)
 - advisors not working 208
 - advisors show that all servers are down 212
 - blue screen displays when starting Network Dispatcher executor 211
 - cannot create registry on port 14099 218
 - CBR will not run 214
 - cbrcontrol fails on Solaris 215
 - cbrcontrol or ndadmin command fails 214
 - cbrserver command is stopped 215
 - common problems and solutions 207, 209, 214, 215, 216, 218
 - Dispatcher, Microsoft IIS, and SSL do not work 208
 - Dispatcher and server will not respond 207
 - Dispatcher high availability not working 208
 - Dispatcher requests not routed 207
 - Dispatcher will not run 207
 - error message when trying to view online Help 210
 - error running Dispatcher with Caching Proxy installed 210
 - extra routes 208
 - GUI does not display correctly 210
 - GUI does not start correctly 210
 - help panels disappear 211
 - high availability in the wide area mode of Network Dispatcher does not work 213
 - lbcontrol or ndadmin command fails 218
 - lbserver will not start 218
 - Mailbox Locator will not run 215
 - Metric Server IOException on Windows 2000 218
 - Metric Server log reports "Signature is necessary for access to agent" 219
 - Metric Server not reporting loads 219
 - mlcontrol or ndadmin command fails 216
 - ndcontrol or ndadmin command fails 209

- troubleshooting (*continued*)
 - Network Dispatcher cannot process and forward a frame 211
 - Path to Discovery prevents return traffic with Network Dispatcher 211
 - port numbers used by CBR 205
 - port numbers used by Cisco Consultant 206
 - port numbers used by Mailbox Locator 205
 - port numbers used by Site Selector 206
 - port numbers used by the Dispatcher 204
 - receive Mailbox Locator error when trying to add a port 216
 - requests not being load balanced 214
 - Site Selector doesn't load-balance correctly 217
 - Site Selector doesn't round-robin (Solaris) 217
 - Site Selector will not run 216
 - SNMPD not working 208
 - spurious error message when starting ndserver on Solaris 2.7 210
 - sscontrol or ndadmin command fails 217
 - sssserver failing to start on Windows 2000 217
 - Syntactical or configuration error 215
 - unable to add a port 216
 - Unable to add heartbeat 208
 - unexpected behavior loading large configuration file 213
- troubleshooting tables
 - CBR 201
 - Cisco Consultant 203
 - Dispatcher component 199
 - Mailbox Locator 202
 - Metric Server 203
 - Site Selector 203

U

- uninstall
 - on AIX 14
 - on Linux 17
 - on Solaris 20
 - on Windows 2000 22
- up, marking a server as 279, 309
- URI affinity 175, 177, 271

- user exit scripts 126
 - denial of service detection 179
 - managerAlert 126
 - managerClear 126
 - serverDown 126
 - serverUp 126

V

- version, displaying
 - advisor 232, 293, 294
 - manager 259, 300, 302, 332, 335

W

- WAS (WebSphere Application Server) advisor 132

- weight

- how the manager sets 124, 184
 - setting

- boundary for all servers on a port 124, 265, 338
 - for a server 279, 309, 341

- xml example 183

- wide area support 142

- configuration example 146

- using GRE 148

- using remote advisors 144

- using remote Dispatcher 143

- wildcard cluster 58, 237

- to combine server

- configurations 168

- to load balance firewalls 169

- with Caching Proxy for

- transparent proxy 170

- wildcard port 60, 265

- ping advisor 130

- to direct unconfigured port

- traffic 170

- Windows 2000

- cluster configure command 59

- installing 22

- ndconfig command 60

- requirements 21

- setting up Dispatcher

- machine 57

- wizard, configuring

- Dispatcher 4

- workload manager advisor

- (WLM) 135



Printed in the United States of America
on recycled paper containing 10%
recovered post-consumer fiber.

GC31-8496-06



Spine information:



WebSphere™ Edge Server for
Multiplatforms

Network Dispatcher Administration Guide

Version 2.0