



**DX Application Acceleration Platform**

**Installation and Administration Guide  
for DXOS**

*Release 5.2*

**Juniper Networks, Inc.**

1194 North Mathilda Avenue  
Sunnyvale, CA 94089

USA

408-745-2000

**[www.juniper.net](http://www.juniper.net)**

Part Number: 093-1827-000, Revision 2.0

This product includes the Envoy SNMP Engine, developed by Epilogue Technology, an Integrated Systems Company. Copyright © 1986-1997, Epilogue Technology Corporation. All rights reserved. This program and its documentation were developed at private expense, and no part of them is in the public domain.

This product includes memory allocation software developed by Mark Moraes, copyright © 1988, 1989, 1993, University of Toronto.

This product includes FreeBSD software developed by the University of California, Berkeley, and its contributors. All of the documentation and software included in the 4.4BSD and 4.4BSD-Lite Releases is copyrighted by the Regents of the University of California. Copyright © 1979, 1980, 1983, 1986, 1988, 1989, 1991, 1992, 1993, 1994. The Regents of the University of California. All rights reserved.

GateD software copyright © 1995, the Regents of the University. All rights reserved. Gate Daemon was originated and developed through release 3.0 by Cornell University and its collaborators. GateD is based on Kirton's EGP, UC Berkeley's routing daemon (routed), and DCN's HELLO routing protocol. Development of GateD has been supported in part by the National Science Foundation. Portions of the GateD software copyright © 1988, Regents of the University of California. All rights reserved. Portions of the GateD software copyright © 1991, D. L. S. Associates.

This product includes software developed by Maker Communications, Inc., Copyright © 1996, 1997, Maker Communications, Inc.

Juniper Networks, the Juniper Networks logo, NetScreen, NetScreen Technologies, the NetScreen logo, NetScreen-Global Pro, ScreenOS, and GigaScreen are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The following are trademarks of Juniper Networks, Inc.: ERX, E-series, ESP, Instant Virtual Extranet, Internet Processor, J2300, J4300, J6300, J-Protect, J-series, J-Web, JUNOS, JUNOScope, JUNOScript, JUNOSE, M5, M7i, M10, M10i, M20, M40, M40e, M160, M320, M-series, MMD, NetScreen-5GT, NetScreen-5XP, NetScreen-5XT, NetScreen-25, NetScreen-50, NetScreen-204, NetScreen-208, NetScreen-500, NetScreen-5200, NetScreen-5400, NetScreen-IDP 10, NetScreen-IDP 100, NetScreen-IDP 500, NetScreen-Remote Security Client, NetScreen-Remote VPN Client, NetScreen-SA 1000 Series, NetScreen-SA 3000 Series, NetScreen-SA 5000 Series, NetScreen-SA Central Manager, NetScreen Secure Access, NetScreen-SM 3000, NetScreen-Security Manager, NMC-RX, SDX, Stateful Signature, T320, T640, T-series, and TX Matrix. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners. All specifications are subject to change without notice.

*Installation and Administration Guide for DXOS Version 5.2*

Copyright © 2006, Juniper Networks, Inc.

All rights reserved. Printed in USA.

Revision History

17 Nov 06—Second Release

The information in this document is current as of the date listed in the revision history. Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer or otherwise revise this publication without notice.

Products made or sold by Juniper Networks (including the ERX-310, ERX-705, ERX-710, ERX-1410, ERX-1440, M5, M7i, M10, M10i, M20, M40, M40e, M160, M320, and T320 routers, T640 routing node, and the JUNOS and SDX-300 software) or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

YEAR 2000 NOTICE

Juniper Networks hardware and software products are Year 2000 compliant. The JUNOS software has no known time-related limitations through the year 2038; however, the NTP application is known to have some difficulty in the year 2036.

End User License Agreement

READ THIS END USER LICENSE AGREEMENT ("AGREEMENT") BEFORE DOWNLOADING, INSTALLING, OR USING THE SOFTWARE. BY DOWNLOADING, INSTALLING, OR USING THE SOFTWARE OR OTHERWISE EXPRESSING YOUR AGREEMENT TO THE TERMS CONTAINED HEREIN, YOU (AS CUSTOMER OR IF YOU ARE NOT THE CUSTOMER, AS A REPRESENTATIVE/AGENT AUTHORIZED TO BIND THE CUSTOMER) CONSENT TO BE BOUND BY THIS AGREEMENT. IF YOU DO NOT OR CANNOT AGREE TO THE TERMS CONTAINED HEREIN, THEN (A) DO NOT DOWNLOAD, INSTALL, OR USE THE SOFTWARE, AND (B) YOU MAY CONTACT JUNIPER NETWORKS REGARDING LICENSE TERMS.

1. The Parties. The parties to this Agreement are Juniper Networks, Inc. and its subsidiaries (collectively "Juniper"), and the person or organization that originally purchased from Juniper or an authorized Juniper reseller the applicable license(s) for use of the Software ("Customer") (collectively, the "Parties").
2. The Software. In this Agreement, "Software" means the program modules and features of the Juniper or Juniper-supplied software, and updates and releases of such software, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller.
3. License Grant. Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer non-exclusive and non-transferable license, without right to sublicense, to use the Software, in executable form only, subject to the following use restrictions:
  - a. Customer shall use the Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller, unless the applicable Juniper documentation expressly permits installation on non-Juniper equipment.
  - b. Customer shall use the Software on a single hardware chassis having a single processing unit, or as many chassis or processing units for which Customer has paid the applicable license fees.
  - c. Other Juniper documentation for the Software (such as product purchase documents, documents accompanying the product, the Software user manual(s), Juniper's Web site for the Software, or messages displayed by the Software) may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, concurrent users, sessions, subscribers, nodes, or transactions, or require the purchase of separate licenses to use particular features, functionalities, or capabilities, or provide temporal or geographical limits. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.

The foregoing license is not transferable or assignable by Customer. No license is granted herein to any user who did not originally purchase the applicable license(s) for the Software from Juniper or an authorized Juniper reseller.

4. Use Prohibitions. Notwithstanding the foregoing, the license provided herein does not permit the Customer to, and Customer agrees not to and shall not:
  - (a) modify, unbundle, reverse engineer, or create derivative works based on the Software;
  - (b) make unauthorized copies of the Software (except as necessary for backup purposes);
  - (c) rent, transfer, or grant any rights in and to any copy of the Software, in any form, to any third party;
  - (d) remove any proprietary notices, labels, or marks on or in any copy of the Software;
  - (e) distribute any copy of the Software to any third party, including as may be embedded in Juniper equipment sold in the secondhand market;
  - (f) use any 'locked' or key-restricted feature, function, or capability without first purchasing the

applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, or capability is enabled without a key; (g) distribute any key for the Software provided by Juniper to any third party; (h) use the Software in any manner that extends or is broader than the uses purchased by Customer from Juniper or an authorized Juniper reseller; (i) use the Software on non-Juniper equipment where the Juniper documentation does not expressly permit installation on non-Juniper equipment; (j) use the Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; or (k) use the Software in any manner other than as expressly provided herein.

5. Audit. Customer shall maintain accurate records as necessary to verify compliance with this Agreement. Upon request by Juniper, Customer shall furnish such records to Juniper and certify its compliance with this Agreement.

6. Confidentiality. The Parties agree that aspects of the Software and associated documentation are the confidential property of Juniper. As such, Customer shall exercise all reasonable commercial efforts to maintain the Software and associated documentation in confidence, which at a minimum includes restricting access to the Software to Customer employees and contractors having a need to use the Software.

7. Ownership. Juniper and Juniper's licensors, respectively, retain ownership of all right, title, and interest (including copyright) in and to the Software, associated documentation, and all copies of the Software. Nothing in this Agreement constitutes a transfer or conveyance of any right, title, or interest in the Software or associated documentation, or a sale of the Software, associated documentation, or copies of the Software.

8. Warranty, Limitation of Liability, Disclaimer of Warranty. If the Software is distributed on physical media (such as CD), Juniper warrants for 90 days from delivery that the media on which the Software is delivered will be free of defects in material and workmanship under normal use. This limited warranty extends only to the Customer. Except as may be expressly provided in separate documentation from Juniper, no other warranties apply to the Software, and the Software is otherwise provided AS IS. Customer assumes all risks arising from use of the Software. Customer's sole remedy and Juniper's entire liability under this limited warranty is that Juniper, at its option, will repair or replace the media containing the Software, or provide a refund, provided that Customer makes a proper warranty claim to Juniper, in writing, within the warranty period. Nothing in this Agreement shall give rise to any obligation to support the Software. Any such support shall be governed by a separate, written agreement. To the maximum extent permitted by law, Juniper shall not be liable for any liability for lost profits, loss of data or costs or procurement of substitute goods or services, or for any special, indirect, or consequential damages arising out of this Agreement, the Software, or any Juniper or Juniper-supplied software. In no event shall Juniper be liable for damages arising from unauthorized or improper use of any Juniper or Juniper-supplied software.

EXCEPT AS EXPRESSLY PROVIDED HEREIN OR IN SEPARATE DOCUMENTATION PROVIDED FROM JUNIPER AND TO THE EXTENT PERMITTED BYLAW, JUNIPER DISCLAIMS ANY AND ALL WARRANTIES IN AND TO THE SOFTWARE (WHETHER EXPRESS, IMPLIED, STATUTORY, OR OTHERWISE),INCLUDING ANY IMPLIED WARRANTY OF MERCHANTABILITY, FITNESS FOR A PARTICULAR PURPOSE, OR NONINFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR ORINTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK.

9. Termination. Any breach of this Agreement or failure by Customer to pay any applicable fees due shall result in automatic termination of the license granted herein. Upon such termination, Customer shall destroy or return to Juniper all copies of the Software and related documentation in Customer's possession or control.

10. Taxes. All license fees for the Software are exclusive of taxes, withholdings, duties, or levies (collectively "Taxes"). Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software.

11. Export. Customer agrees to comply with all applicable export laws and restrictions and regulations of any United States and any applicable foreign agency or authority, and not to export or re-export the Software or any direct product thereof in violation of any such restrictions, laws or regulations, or without all necessary approvals. Customer shall be liable for any such violations. The version of the Software supplied to you may contain encryption or other capabilities restricting your ability to export the Software without an export license.

12. Commercial Computer Software. The Software is "commercial computer software" and is provided with restricted rights. Use, duplication, or disclosure by the United States government is subject to restrictions set forth in this Agreement and as provided in DFARS 227.7201 through 227.7202-4,FAR 12.212, FAR 27.405(b)(2), FAR 52.227-19, or FAR 52.227-14(ALT III) as applicable.

13. Miscellaneous. This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. For any disputes arising under this Agreement, the Parties hereby consent to the personal and exclusive jurisdiction of, and venue in, the state and federal courts within Santa Clara County, California. This Agreement constitutes the entire and sole agreement between Juniper and the Customer with respect to the Software, and supersedes all prior and contemporaneous agreements relating to the Software, whether oral or written (including any inconsistent contained in a purchase order), except that the terms of a separate written agreement executed by an authorized Juniper representative and Customer shall govern to the extent such terms are inconsistent or conflict with terms contained herein. No modification to this Agreement nor any waiver of any rights hereunder shall be effective unless expressly assented to in writing by the party to be charged. If any portion of this Agreement is held invalid, the Parties agree that such invalidity shall not affect the validity of the remainder of this Agreement.

If you have any questions about this agreement, contact Juniper Networks at the following address:

Juniper Networks, Inc.  
1194 North Mathilda Avenue  
Sunnyvale, CA 94089  
USA  
Attn: Contracts Administrator



# Table of Contents

<b>About This Guide</b>	<b>XV</b>
Audience .....	XV
Conventions .....	XVI
Package Contents .....	XVI
Document Organization .....	XVII
Documentation Feedback .....	XVIII
Product Training .....	XVIII
Technical Support .....	XVIII

## Part 1

### Overview

<b>Chapter 1</b>	<b>Introduction</b>	<b>3</b>
	Accelerating the IP- and Web-based Applications in Your Data Center .....	3
	Server Load Balancing .....	4
	Securing Applications Through SSL .....	4
	Securing Your Data Center .....	6
	Optimizing Branch Office Internet Access .....	6
	Accelerating Web Applications .....	6
	Providing Authentication to Any Web Application .....	8
	Scaling Capacity and Providing High Availability .....	8
	Increasing Web Transaction Success Rate through Caching and Application Rules .....	9
	Balancing Server Load Globally .....	11
	Monitoring and Reporting .....	11
	DX Platform Services .....	12
	Network Topologies .....	13
	The DX Product Family .....	15
	Chassis Connections and Labels .....	16
	DX Product Licensing Options .....	17
	Overview .....	17
	DX Base License Features .....	19
	HTTP Acceleration License Features .....	21
	Advanced HTTP Acceleration License Features .....	22
	GSLB License Features .....	23
	New Features in this Release .....	23

<b>Chapter 2</b>	<b>Application and DX Product Concepts</b>	<b>25</b>
	Multi-Level Administrative Rights .....	26
	User Access Levels.....	26
	User Roles .....	27
	Valid User Names and Passwords.....	27
	Server Load Balancing.....	28
	SLB with the DX .....	28
	Load Balancing Policies .....	30
	SSL Termination.....	33
	Basic Conventions and Terms.....	34
	HTTP(S) Authentication.....	39
	Overview .....	40
	Authentication, Authorization, and Auditing (AAA) .....	40
	Authentication Methods.....	41
	Password Change Request.....	45
	Health Checking.....	46
	In-Band Layer 4 Health Checking.....	46
	Out-of-Band Layer 4 Health Checking.....	47
	Out-of-Band Layer 7 Health Checking.....	47
	Scriptable Health Checking.....	47
	Cluster Health Checking Policies.....	47
	Failover.....	48
	Failover Processing.....	48
	ActiveN .....	49
	Topologies .....	50
	Layer 4 Switching and ActiveN .....	52
	Client IP Sticky .....	59
	3G Caching.....	60
	The Juniper Solution .....	60
	Cache Usage Scenarios .....	61
	Caching Features .....	61
	Using Overdrive Application Rules .....	63
	Basic Application Rule Concepts.....	63
	Types of Application Rules.....	63
	Application Rule Grammar .....	65
	Global Server Load Balancing.....	66
	GSLB with the DX .....	66
	Forward Proxy Acceleration.....	74
<b>Chapter 3</b>	<b>Using the DX Administrative Interfaces</b>	<b>81</b>
	Overview .....	81
	Using the Command Line Interface.....	81
	Accessing the Command Line Interface.....	82
	Working in DXSHELL.....	83
	The Web User Interface (WebUI).....	86
	Enabling the WebUI Server.....	86
	Setting the WebUI Interface to Communicate over SSL .....	87
	Accessing the WebUI.....	87
	Working with the WebUI.....	88
	Making Changes with the WebUI.....	89
	On-Line Help in the WebUI.....	89
	Logging out of the WebUI .....	90
	SNMP Agent.....	90

**Part 2****Installation Information and Procedures**

<b>Chapter 4</b>	<b>Installing Your DX Appliance</b>	<b>95</b>
	Installation Overview .....	95
	Network Configuration Information Needed .....	96
	Installing Your DX Appliance .....	97
<b>Chapter 5</b>	<b>Performing Initial Configuration of the DX Appliance</b>	<b>99</b>
	Connecting to the DX Appliance with a Terminal or Terminal Emulator .....	99
	Logging-In for the First Time .....	102
	Read and Agree to the License Agreement .....	103
	Answer the Configuration Questions .....	103
	Changing the Default Administrator Account Password .....	105

**Part 3****Configuration Information and Procedures**

<b>Chapter 6</b>	<b>DX Appliance Configuration Flows</b>	<b>109</b>
	Setting up the DX Appliance for Server Load Balancing .....	109
	Securing Non-Web Applications through SSL .....	110
	Migrating Web Applications to Secure Web .....	111
<b>Chapter 7</b>	<b>Administering Your DX Platform</b>	<b>113</b>
	Managing Users .....	113
	Adding a User .....	114
	Changing the User's Password .....	115
	Clearing a User's Role .....	116
	Assigning Local or Remote Access Rights .....	116
	Deleting a User .....	117
	Viewing User Information .....	117
	Making Global Changes to User Accounts .....	117
	Exporting and Importing User Accounts .....	119
	Resetting the admin User Password .....	120
	Administrator Remote Authentication .....	121
	Obtaining a License Key .....	124
	Obtaining a Juniper Customer Support Center (CSC) User ID and Password .....	125
	Obtaining a Permanent License .....	126
	Installing the DX License Key .....	128
	Using the Administrator Audit Trail .....	128
	Syntax of the Log Entries .....	128
	Enabling and Disabling Logging of show Commands .....	129
	Configuring System Event Logging and Notification .....	129
	Viewing Event Logging Configuration .....	130
	Viewing the System Event Log .....	130
	Sample Configuration .....	131
	Managing Your DX Appliance Configuration .....	131
	Exporting a Configuration .....	131
	Importing a Configuration .....	132

- Viewing the Contents of a Configuration File ..... 133
- Editing a Configuration File ..... 133
- Restoring the Factory Default Configuration ..... 135
- Creating a DX Platform System Image..... 135
- Synchronizing Configurations Across Multiple DX Appliances ..... 138
- Configuring the Login Banner ..... 144
- Upgrading the DX Application Acceleration Platform Software ..... 146
  - Viewing Enabled DX Platform Features ..... 146
  - Upgrade Requirements ..... 147
  - Preserving Your Configuration and Choosing a .pac File..... 147
  - Upgrading Using the install Command ..... 148

**Chapter 8 Integrating the DX Appliance into Your Network 151**

- Overview ..... 152
- Cluster, Redirector, Forwarder, Cache, and ActiveN Group Naming
  - Conventions ..... 152
- Deploying the DX Appliance Behind an External Server Load
  - Balancer (SLB) ..... 153
- Integrating the DX Appliance into a Direct Server Return (DSR)
  - Environment ..... 154
  - Overview ..... 154
  - What is Direct Server Return (DSR)? ..... 154
  - Why use DSR? ..... 154
  - How Does DSR Work? ..... 154
  - Inserting the DX Appliance into a DSR Environment ..... 155
- Client IP Transparency ..... 155
  - Client IP Transparency Commands ..... 157
- Source Network Address Translation..... 157
  - SNAT Operation..... 158
  - SNAT Configuration Commands ..... 158
- Floating VIP ..... 160
- Connection Binding and Microsoft's NTLM Authentication Protocol..... 161
  - Configuring Connection Binding..... 161
- Connection Binding and Layer 7 Health Checking..... 162
- Reverse Route Return ..... 162
  - Behavior ..... 162
  - Reverse Route Return Commands ..... 163
- TCP Selective Acknowledgement ..... 164
- Configuring a Virtual LAN..... 164
  - Behavior ..... 165
  - VLAN Commands ..... 166
- Pausing a Target Host ..... 167
  - Target Host Pause Commands..... 168
- Using a Local IP for Target Host Communication ..... 169
  - Local IP Configuration Commands ..... 169
- Enabling Target Server Compression ..... 170
  - Target Server Compression Mode..... 170
  - Target Compression Encoding ..... 171
  - Configuring Target Server Compression with DXSHELL..... 171
  - Configuring Target Server Compression with the WebUI ..... 172
- Instant Redirect..... 173
- Configuring SNMP ..... 174
  - Configuring the SNMP Agent Parameters..... 174
  - Configuring the SNMP Agent for Sending Traps..... 174



Configuring Support for the Juniper Secure Access SSL VPN .....	175
Configuring SA Compatibility.....	176
Viewing the SA Compatibility Configuration .....	176
Modifying the SA Compatibility Configuration .....	176
<b>Chapter 9   Configuring Server Load Balancing</b> .....	<b>179</b>
Configuring a Basic Server Load Balancer .....	179
Customizing the SLB Service.....	180
Configuring Network Address Translation (NAT).....	180
Configuring the SLB Group Communication Protocol.....	181
Configuring the SLB Group Load-Balancing Policy .....	181
Configuring SLB Health Checking .....	181
Configuring Client to Server Sticky .....	182
Configuring Quality of Service (QoS).....	183
Configuring SLB Session Parameters .....	183
Pausing a Target Host .....	184
Deleting an SLB Group .....	185
Statistics.....	185
Overall Statistics .....	185
Group Statistics.....	188
Target Host Statistics .....	188
<b>Chapter 10   Setting Up the DX Appliance for SSL Traffic</b> .....	<b>189</b>
Before You Begin .....	190
Step-by-step Configuration Examples.....	190
Possible SSL Cluster Configurations with the DX Appliance .....	190
SSL Configuration Examples: Listen: Enabled and Target: Disabled .....	190
SSL Configuration Examples: Listen: Disabled and Target: Enabled .....	191
SSL Configuration Examples: Listen: Enabled and Target: Enabled.....	192
SSL Configuration Examples: Listen: Disabled and Target: Disabled .....	193
SSL Forwarder Configuration .....	193
Possible SSL Forwarder Configurations with the DX Appliance.....	194
SSL Configuration Examples: Listen: Enabled and Target: Disabled .....	194
SSL Configuration Examples: Listen: Disabled and Target: Enabled .....	195
SSL Configuration Example, Listen: Enabled, Target: Enabled .....	196
SSL Configuration Example, Listen: Disabled, Target: Disabled .....	197
Importing Existing Keys and Certificates.....	197
Importing from Apache mod_ssl .....	198
Importing from ApacheSSL.....	199
Importing from IIS 4 on Windows NT.....	200
Exporting Key and Certificate Files to the DX Appliance:.....	201
Importing from IIS 5 on Windows 2000 .....	202
Exporting Key and Certificate Files to the DX Appliance.....	204
Importing from iPlanet.....	205
Generating Keys and Certificates .....	206
GEN KEY .....	206
GEN CSR.....	206
GEN SSC .....	208
SSL Ciphersuite Details.....	209
Specifying Your Own List of SSL Ciphersuites.....	210
Forcing Clients to use HTTPS with Cluster Redirection (Auto SSL) .....	212
EXAMPLE: Configuring Cluster Redirection to Redirect HTTP Requests to HTTPS.....	212

	Configuring SSL Client Authentication .....	213
	Overview .....	213
	Certificate Authority (CA) Certificate Presentation .....	214
	Trusted Certificate Authority (CA) Certificate Storage.....	214
	Certificate Revocation List (CRL) .....	215
	Example of Chain Certificates and CRLs .....	215
	DXSHELL Commands for SSL Client Authentication .....	218
	Browsers that Poorly Support SSL Client Authentication .....	219
<b>Chapter 11</b>	<b>Configuring Health Checking</b>	<b>221</b>
	Configuring Layer 4 Health Check Settings.....	221
	Configuring Layer 7 Health Check Settings.....	222
	Viewing Health Check Configuration Settings.....	223
	Viewing All Settings .....	223
	Viewing a Particular Setting .....	224
	Layer 7 Health Logging System Log Messages.....	225
	Notes on Layer 7 Health Checking .....	226
	Customizing Layer 7 Health Checking for SLB Services .....	227
	One-to-one Cluster to Server Mapping .....	227
	Conserving IPs with One-to-One Mapping .....	228
	Scriptable Health Checking .....	229
	Expect/TCL Scripts.....	229
	Scriptable Health Checking TCL API.....	231
	The Expect/TCL Command Set .....	232
	Logging and Statistics .....	233
	TCL UDP Extension .....	234
	Scriptable Health Checking Commands .....	236
	Capture and Configuration Example.....	237
	Sample Scripts.....	237
	Health Check Settings.....	238
<b>Chapter 12</b>	<b>Configuring ActiveN</b>	<b>239</b>
	Overview .....	239
	Configuring the ActiveN Service.....	239
	Modifying Your ActiveN Configuration .....	242
	Sample ActiveN Configuration .....	243
	DX Appliance 1 Configuration .....	243
	DX Appliance 2 Configuration .....	244
	Configuring Advanced Settings for the ActiveN Service.....	244
	Set Commands .....	245
	Delete Commands.....	248
	Clear Commands.....	249
	Show Commands .....	249
<b>Chapter 13</b>	<b>Setting up the DX Appliance for “Sticky” Traffic</b>	<b>251</b>
	Overview .....	251
	Configuration Instructions for Cookie-Based Client Stickiness .....	251
	Configuration Instructions for Client IP-Based Stickiness.....	252

<b>Chapter 14</b>	<b>Configuring HTTP(S) Authentication</b>	<b>253</b>
	Authentication Commands.....	253
	Set Commands .....	253
	Show Commands .....	255
	Clear Commands.....	256
	Configuring Basic Authentication Parameters .....	256
	Configuring Authentication Using an LDAP Server .....	257
	Configuring Your DX Appliance for LDAP Authentication .....	258
	Configuring Your DX Appliance for LDAP Authentication over SSL.....	259
	Configuring Authentication Using Microsoft Active Directory as the LDAP Server .....	260
	Configuring Your DX Appliance for LDAP Authentication Using Active Directory.....	261
	Configuring Authentication Using RSA SecurID .....	261
	Configuring Your DX Appliance for Authentication Using SecurID .....	263
	Logging Into Multiple Web Applications within a Domain .....	263
	Sample Usage of the Single Sign-On Feature .....	264
	Configuring the DX Appliance for Single Sign-On .....	265
	Modifying Single Sign-On Configuration for a Cluster .....	267
	Viewing Single Sign-On Configuration for a Cluster .....	267
<b>Chapter 15</b>	<b>Configuring HTTP(S) Logging</b>	<b>269</b>
	Overview .....	269
	Compiling Log Information on a Master Logging Machine .....	270
	Logging Client IP on the Webserver with a Custom Header .....	273
	Configuring Logging with Apache .....	274
	Configuring Logging with IIS .....	275
	Configuring Logging with Resin.....	280
	Configuring Logging with iPlanet.....	280
	Configuring Logging with NetCache .....	281
<b>Chapter 16</b>	<b>Configuring the Forward Proxy Accelerator</b>	<b>285</b>
	Overview .....	285
	Command Line Interface Commands.....	285
	Forward Proxy Accelerator with the WebUI .....	287
<b>Chapter 17</b>	<b>Configuring the 3G Cache</b>	<b>289</b>
	3G Cache Commands.....	289
	AppRules.....	293
	Usage.....	294
	Case 1 .....	294
	Case 2 .....	294
	Case 3 .....	294

<b>Chapter 18</b>	<b>Configuring OverDrive Application Rules</b>	<b>297</b>
	Writing Application Rules.....	297
	Test Conditions.....	298
	Actions.....	304
	Combining Rules into Rule Sets.....	308
	Rule Execution Modes.....	309
	Action Execution Modes.....	309
	Application Rule Relationships.....	310
	Importing Rule Sets.....	318
	Binding Rule Sets to Clusters.....	319
	Enabling OverDrive.....	319
	Modifying Rules.....	320
	Viewing Application Rules on the DX Appliance.....	320
	Limitations When Applying Application Rules.....	321
	Application Rules and Latency.....	321
	Displaying Rules.....	322
	User Data Parsing.....	322
	Using the Forward Action with PTH Rules.....	322
	Using Prepend, Append, and Replace (PAR) Actions.....	323
	Source IP Filtering.....	324
	Logging.....	325
	Application Rule Scenarios.....	325
	Route Request Application Rules.....	325
	Request Retry, Alerting, and Log (Transaction Assurance) AppRules.....	326
	Request Routing Application Rules.....	327
	Request Sentry Examples.....	329
	Request Translator Examples.....	330
	Request Retry Examples.....	331
	Request Routing Examples.....	331
	Page Translator Examples.....	332
<b>Chapter 19</b>	<b>Configuring Global Server Load Balancing</b>	<b>335</b>
	GSLB Configuration Task Flow.....	336
	Configuring the GSLB Agent.....	336
	Defining the GSLB Remote Nodes.....	337
	Configuring a GSLB Resolver.....	338
	Configuring Resolver Basics.....	338
	Defining GSLB Groups.....	339
	Configuring the Local DNS Server.....	346
	Synchronizing Your GSLB Configuration.....	349
	Removing Configuration Information.....	349
	Show Configuration Commands.....	349
	Statistics Commands.....	350
	Deployment Scenarios.....	351
	Basic DNS, Resolver, and Group Configuration.....	351
	Simple Round Robin.....	352
	Weighted Round Robin.....	352
	Metric-based Load Balancing.....	352
	Adjusted Metric Load Balancing.....	354
	RTT-only Load Balancing.....	354
	GSLB Failover.....	355

<b>Chapter 20</b>	<b>Configuring Failover</b>	<b>357</b>
	Configuring Failover on Your DX Appliance .....	357
	Customizing the Failover Process .....	357
	Viewing Failover Configuration and Statistics.....	359
	Viewing the Current Failover Configuration .....	359
	Viewing Statistical Information .....	360
	Viewing Status Information about Services.....	363
	Migrating to the New Failover Method .....	365
	Migrating Server Failover Configurations .....	365
	Migrating SLB Failover Configurations .....	365
	Migrating ActiveN and ActiveOne Failover Configurations .....	366
	Initiating a Manual Server Failover .....	367
	Gateway Failure Detection .....	368
<b>Chapter 21</b>	<b>Tuning the DX Appliance for Enterprise Applications</b>	<b>373</b>
	Target Tuning Tool.....	373
	WebDAV .....	374
	Methods .....	375
	Compression of 401 Responses.....	375
	Compression of “text/x-component” MIME Type.....	375
	Integration with Application Rules.....	375
	Optimization .....	376
	New WebDAV and HTTP Extensions .....	376
	OWA Commands.....	377
<b>Part 4</b>	<b>Monitoring and Troubleshooting Information and Procedures</b>	
<b>Chapter 22</b>	<b>Performance Monitoring</b>	<b>381</b>
	View Juniper Server Statistics.....	382
	Capacity Planning .....	383
	Historical Rates and Statistics.....	383
	The Round Robin Database Mechanism .....	384
	Memory Considerations.....	384
	Description .....	384
	Statistical Data Items .....	385
	Enabling Historical Rates and Statistics .....	386
	Disabling Historical Statistics .....	386
	DXSHELL Output Example .....	390
	CSV Export Statistics .....	391
	Export CSV Statistics Commands.....	392
	Exporting CSV Statistics from the WebUI .....	392
	Advanced Statistics .....	393
	Overview .....	393
	I/O Listen Statistics .....	393
	I/O Target Host Statistics .....	394
	I/O Physical Target Statistics.....	395
	HTTP Listen Statistics: Requests from Clients .....	395
	HTTP Target Host Statistics.....	398

SSL Listen Statistics .....	401
SSL Target Host Statistics.....	401
DXSHELL Commands for Advanced Statistics .....	402
Clearing Cluster Statistics.....	403
Forwarder Statistics .....	403
Forwarder's Target Host Statistics .....	404
Clearing Forwarder Statistics .....	404
Redirector Statistics .....	404
Clearing Redirector Statistics .....	404
DX Appliance Server Statistics .....	405
Clearing DX Appliance Server Statistics .....	405
Web Log Configuration .....	405
Web Log Commands .....	407
Web Log Batch Mode.....	408

**Chapter 23 Troubleshooting 413**

---

Checking Settings.....	413
Troubleshooting .....	414
Slow or Degraded Performance .....	414
DX Appliance is Not Responding to Requests for Web Content .....	414
Cannot Access the WebUI with your Web Browser.....	417
Cannot Connect to the DXSHELL Command Line with SSH.....	417
Technical Service Dump.....	418
What Information is Collected .....	418
What Information is not Collected .....	418
Creating the Technical Service Dump .....	418
Using tcpdump to Generate a Detailed Report of Network Activity .....	419
Guidelines for Using tcpdump.....	420
Setting up Your DX for tcpdump.....	420
Running the tcpdump Utility.....	420
Managing tcpdump Files.....	421
Sample TCPdump Scenario .....	423

**Part 5 Reference Information**

---

**Appendix A Glossary 427**

---

**Appendix B List of Events 433**

---

**Appendix C Configuring Failover by Service 437**

---

Configuring Failover for the SLB Service .....	437
Configuring Server Failover for the Cluster, Forwarder, and Redirector Services .....	438
Initiating a Manual Server Failover .....	440
Configuring Failover for the ActiveN Service .....	441

**Index..... 443**

# About This Guide

This document describes how to install and configure the hardware and software of the DX Application Acceleration Platform Quick Start. It provides an overview of the applications and network topologies in which the DX Application Acceleration Platform Quick Start can be deployed. This document applies to all DX Application Acceleration Platform Quick Start product models.

This chapter contains the following topics:

- “Audience” on page XV
- “Conventions” on page XVI
- “Document Organization” on page XVII
- “Documentation Feedback” on page XVIII
- “Product Training” on page XVIII
- “Technical Support” on page XVIII

## Audience

---

This document assumes that the reader has knowledge of the network architecture or topology in which the DX will be installed. This documentation is intended for network engineers, Web operations engineers, IT professionals, and system administrators who have experience with the following:

- Installing, configuring, and administering network equipment
- Managing Web traffic and connectivity

## Conventions

Table 1 illustrates the text conventions that are used in this manual.

**Table 1: Notation Conventions**

Notation	Example	Meaning and Use
Courier typeface	.ini file	Code listings, names of files, symbols, and directories, are shown in courier typeface.
Bold Courier typeface	<b>install</b>	In the command line, keywords are shown in bold, non-italic, Courier typeface. Enter them exactly as shown.
Square brackets	[version]	You may, but need not, select one item enclosed within square brackets. Do not enter the brackets.
Angle brackets	<username>	You must provide the information enclosed within angle brackets. Do not enter the brackets.
Bar	les   les.out	You may select one (but not more than one) item from a list separated by bars. Do not enter the bar.

The following notes are used to convey information important for the user:



**NOTE:** Indicates information that may be helpful to the user.



**CAUTION:** Indicates that an action by the user may cause damage to equipment or a loss of data.



**WARNING:** Indicates that an action by the user may cause injury to themselves or others.

## Package Contents

The DX ships with the following items:

- One DX Web I/O Processor
- One AC Power Cord
- One Ethernet Cable
- One Null-Modem Cable
- One Rack Mount Kit (rack ears and screws)
- One *DX Application Acceleration Platform Quick Start*



- One CD-ROM containing the following manuals in Adobe PDF format:
  - *DX Application Acceleration Platform Quick Start*
  - *DX Application Acceleration Platform Installation and Administration Guide*
  - *DX Application Acceleration Platform Command Line Reference Guide*

If any of these items are missing or damaged, please contact a Juniper Networks Customer Service Representative to obtain a replacement.

## Document Organization

---

This guide is organized into multiple parts containing installation and configuration information followed by reference material. See Table 2.

**Table 2: Organization of DX Installation and Administration Guide**

Section
<b>Part 1 Overview</b>
Chapter 1, “Introduction”
Chapter 2, “Application and DX Product Concepts”
Chapter 3, “Using the DX Administrative Interfaces”
<b>Part 2 Installation Information and Procedures</b>
Chapter 4, “Installing Your DX Appliance”
Chapter 5, “Performing Initial Configuration of the DX Appliance”
<b>Part 3 Configuration Information and Procedures</b>
Chapter 6, “DX Appliance Configuration Flows”
Chapter 7, “Administering Your DX Platform”
Chapter 8, “Integrating the DX Appliance into Your Network”
Chapter 9, “Configuring Server Load Balancing”
Chapter 10, “Setting Up the DX Appliance for SSL Traffic”
Chapter 11, “Configuring Health Checking”
Chapter 12, “Configuring ActiveN”
Chapter 13, “Setting up the DX Appliance for “Sticky” Traffic”
Chapter 14, “Configuring HTTP(S) Authentication”
Chapter 15, “Configuring HTTP(S) Logging”
Chapter 16, “Configuring the Forward Proxy Accelerator”
Chapter 17, “Configuring the 3G Cache”
Chapter 18, “Configuring OverDrive Application Rules”
Chapter 19, “Configuring Global Server Load Balancing”
Chapter 20, “Configuring Failover”
Chapter 21, “Tuning the DX Appliance for Enterprise Applications”

**Table 2: Organization of DX Installation and Administration Guide (continued)**

Section
<b>Part 4 Monitoring and Troubleshooting Information and Procedures</b>
Chapter 22, "Performance Monitoring"
Chapter 23, "Troubleshooting"
<b>Part 5 Reference Information</b>
Appendix A, "Glossary"
Appendix B, "List of Events"
Appendix C, "Configuring Failover by Service"

## Documentation Feedback

---

We welcome your feedback on this guide. Send comments through U.S. post to:

Juniper Networks, Inc.  
 1194 North Mathilda Avenue  
 Sunnyvale, CA 94089  
 U.S.A.  
 Attention: DX Application Acceleration Platform Technical Documentation

## Product Training

---

Juniper Networks offers product training throughout the year. For information on available classes and registration information, go to <http://www.juniper.net/training/>.

## Technical Support

---

To contact Juniper Networks technical support, use one of the following methods:

- Go to <http://www.juniper.net/support>
- Call +1-888-314-JTAC (U.S, Canada, and Mexico) or +1-408-745-9500

## Part 1

# Overview

This part of the *Installation and Administration Guide for DXOS* provides an overview of the the DX Application Acceleration hardware and software, describes the various administrator interfaces, and presents terminology and concepts associated with the DX product and the technologies it uses.

These topics can be found in the following chapters:

- Chapter 1, “Introduction” on page 3
- Chapter 2, “Application and DX Product Concepts” on page 25
- Chapter 3, “Using the DX Administrative Interfaces” on page 81



## Chapter 1

# Introduction

This chapter provides an introduction to the DX Application Acceleration Platform. The software, hardware, and common applications are discussed, as well as the DX product licensing options.

This chapter includes the following topics:

- “Accelerating the IP- and Web-based Applications in Your Data Center” on page 3
- “DX Platform Services” on page 12
- “Network Topologies” on page 13
- “The DX Product Family” on page 15
- “DX Product Licensing Options” on page 17
- “New Features in this Release” on page 23

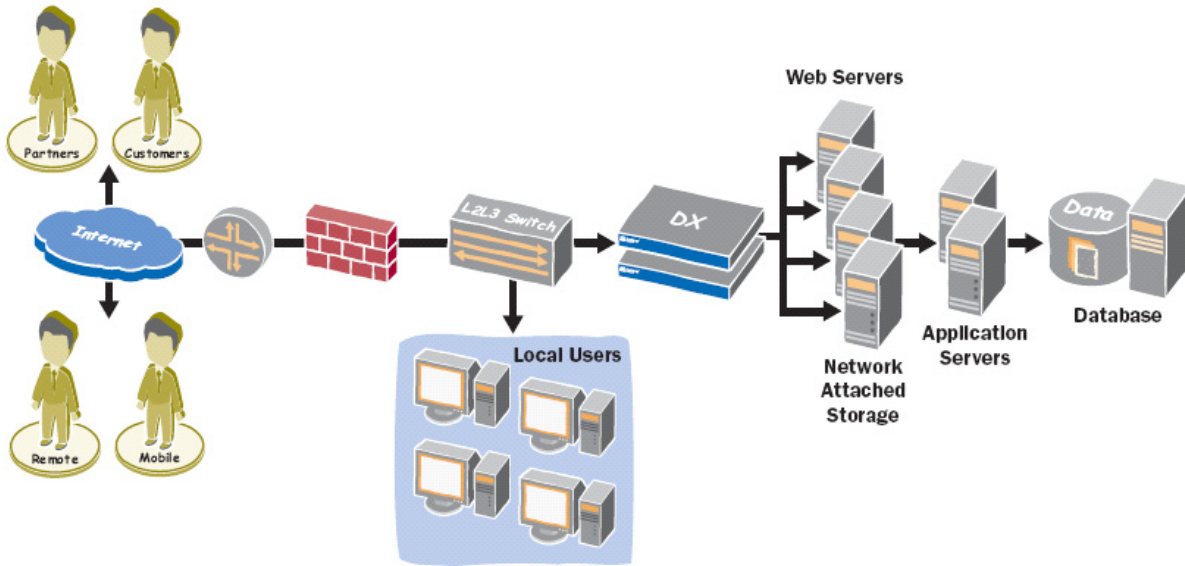
### **Accelerating the IP- and Web-based Applications in Your Data Center**


---

The DX application acceleration platform delivers a complete data center acceleration solution for Web-enabled and IP-based business applications. Through the unique DX Framework, the DX platform greatly improves the end-user experience by delivering content more quickly and meeting IT budget, high-availability and security requirements through a combination of centralized services—including server load balancing, global server load balancing, SSL encryption and termination, HTTP compression and application security—on a single device. The DX platform has scaling options in both functionality and performance for any business environment. Working with other Juniper solutions such as the WX/WXC application acceleration platforms and the Secure Access SSL VPN, the DX platform contributes to the industry's most complete, secure and assured application delivery solution for the distributed enterprise.

The DX application acceleration platform resides in the data center in front of content servers, where it serves as a full request/response-aware bi-directional HTTP proxy for processing incoming and outgoing requests. By offloading servers from CPU-intensive tasks, the DX platform makes servers significantly more efficient, accelerating the performance of Web-enabled applications and improving the productivity of remote, branch-office and mobile employees that access centralized business applications. See Figure 1.

**Figure 1: Speed delivery of Web-enabled applications from the data center to local, remote and mobile users.**



 All features described in this chapter are not available with every license. Please see “DX Product Licensing Options” on page 17 for details about which features are available for each license.

### **Server Load Balancing**

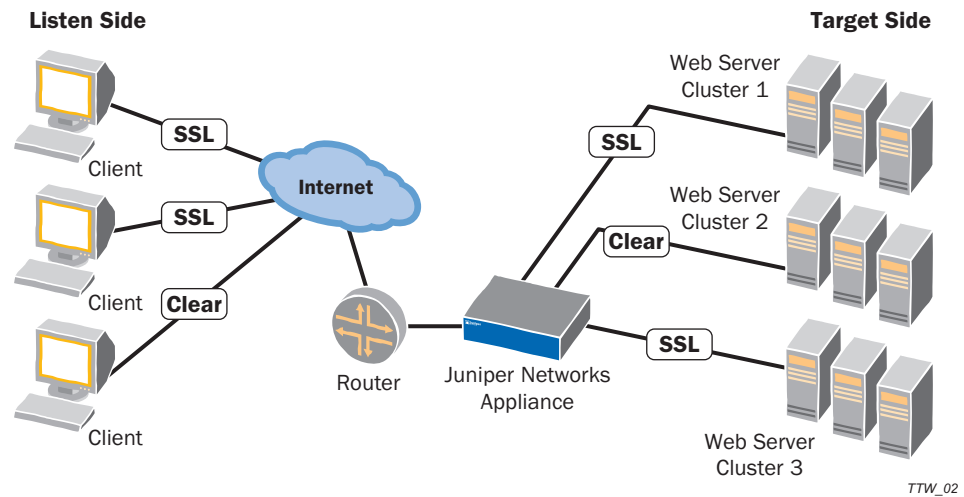
As a fundamental service, the DX platform delivers full Layer 4-7 server load balancer (SLB) functionality. Equipped with multiple load-balancing methods and sticky modes, the DX platform can load-balance any application that needs to be scaled or made highly available. The DX platform performs server health checks ranging from simple ICMP ping and Layer 7 HTTP content validation to scriptable health-check facilities to verify that applications, whether custom or off the shelf, are healthy before any requests are sent to the server. Full server and connection management features ensure that servers and services in a live network can be inserted or deactivated at any time without causing a service outage.

### **Securing Applications Through SSL**

The DX platform provides transport-level security by providing full Secure Socket Layer (SSL) session termination for any Web or IP application offloading this task from content servers and providing an extra layer of protection for critical data center resources. With options for either hardware-assisted SSL termination for high-capacity needs or FIPS L3 for maximum protection, the DX platform can secure all applications to the desired level.

Configuring the DX to serve data using SSL is easy. The DX sits in front of your server(s), holds your site certificates and keys, and processes the incoming and outgoing SSL transactions. Refer to Figure 2.

Figure 2: Listen and Target-Side Illustration



### Migrating Web Applications to Secure Web (HTTP to HTTPS)

The DX platform acts a reverse proxy between the clients and your Web servers allowing it to redirect requests from a browser to a new location or redirect requests using the same or a different protocol (HTTP or HTTPS) than the client. The following table illustrates these various options.

Protocol Usage between Client and DX	Protocol Usage between DX and Web Server
HTTP	HTTP
HTTP*	HTTPS*
HTTPS	HTTP
HTTPS	HTTPS

\* While it is possible to receive communication from a client using the HTTP protocol and then communicate with the Web servers using the HTTPS protocol, it is not reasonable for the DX to do so.

Using this capability, the DX platform can:

- Migrate clients over to using HTTPS—Configure the DX to redirect all requests coming in through HTTP on port 80 to the *same page* using HTTPS on port 443
- Migrate Web applications to secure Web—Configure the DX to redirect all requests coming in through HTTP on port 80 to a *new page* using HTTPS on port 443.

To redirect client requests, the DX responds with the HTTP 302 “temporarily moved” response code in compliance with RFC 2616. The response also contains the new location in an HTTP Location header which both HTTP 1.0 and HTTP 1.1-compliant clients recognize.

### **The DX Appliance as an SSL Forwarder**

The DX can also act as an SSL Forwarder. In Forwarder mode, the DX performs the SSL encryption or decryption, and then forwards the HTTP or non-HTTP traffic directly to the server or client. In the Forwarder mode, the client connection gets terminated at the DX, and the DX opens a new connection to the server. The DX then forwards HTTP and non-HTTP traffic transparently from the client to the server, which means it never initiates termination of a connection. That is done by either the client or the server.

### **Securing Your Data Center**

The DX platform also acts as an "internal firewall," protecting the Web tier and content servers from malicious TCP and HTTP/Web-based attacks by authenticating all users and HTTP sessions before allowing access. The DX platform can provide per-request authorization by leveraging the existing RADIUS and LDAP infrastructure, secure data and connections, protect servers from denial-of-service (DoS) attacks and SYN floods, and provide other security features based on native HTTP protocol communication.

### **Optimizing Branch Office Internet Access**

In a typical branch office, access to the internet and external Web sites is provided through one or more head quarters proxy servers. The proxy servers provide pages with or without compression to the client as each server is able.

With a DX platform serving as the provider of Web site content to the clients from the head quarters proxy server, the following benefits are realized:

- Compression is always performed on data that is sent to the client, reducing download time for HTTP traffic.
- Load balancing is provided for the head quarters proxy servers.
- Fewer proxy servers are needed because the DX becomes the contact point for all of the proxy servers, multiplexing the numerous HTTP connections required for the bursty traffic associated with the TCP protocol into a significantly smaller number of required connections.
- Clients experience improved performance because they are not sent to a proxy server that is no longer available.

### **Accelerating Web Applications**

Designed with HTTP in mind, the DX platform provides a full suite of services for Web-enabled applications that benefit both the end-user and the data center.

TCP/IP connection multiplexing enables the DX platform to reduce thousands of incoming client connections down to just a few, relieving the TCP/IP connection-management burden on back-end servers and allowing them to do what they do best: serve content. By assuming responsibility for resource-intensive tasks such as session set-up and tear-down, the DX platform frees up CPU cycles on the servers, allowing them to process up to four times the number of normal requests.



A full understanding of the requesting and sending browser, as well as content and network conditions, enables the DX platform to speed content to the user in record time without requiring a proprietary client to the browser. The DX platform employs standards-based Deflate and GZIP algorithms to compress all application flows, from standard HTTP objects to Microsoft Office documents and XML content, to accelerate Web-enabled applications for all local, remote and mobile users. To provide the best possible performance, the DX platform imposes no size restriction for documents being compressed, and support for mechanisms such as chunking allow content to be displayed as soon as it's available to accelerate page loads.

A major cause of Web traffic delay is the TCP connection set-up and bandwidth tuning process called slow-start, which can take multiple round trips before everything is optimized. Servers and clients automatically close these connections after a brief period of inactivity, even within an active browser session, forcing users to incur the start-up penalty each time they initiate an action. To accelerate performance, the DX platform keeps all client TCP connections open for as long as capacity allows, ensuring an immediate response the next time the client browser retrieves content from the server, even if hours or days later.

### **Accelerating Microsoft's Outlook Web Application**

As many applications move from being stand-alone enterprise applications to Web-based applications (such as Outlook Web Access [OWA]), the Internet Engineering Task Force (IETF) has begun and continues to develop Web-based Distributed Authoring and Versioning (WebDAV), a set of extensions to the HTTP protocol that allows users to collaboratively manage and edit files on remote Web servers.

OWA uses the WebDAV extensions to the HTTP protocol to provide increased functionality. The DX supports WebDAV extensions to HTTP protocol and allows users to accelerate OWA provided you have a valid HTTP Acceleration license (see "DX Product Licensing Options" on page 17). You can set the WebDAV option for the desired cluster on the DX platform to enable the following:

- WebDAV methods
  - Standard—GET, HEAD, POST, PUT
  - Extended—DELETE, TRACE, OPTIONS, CONNECT
- Connection binding
- Compression of unauthorized responses (HTTP 401 responses)—The 401 response, containing a relatively large amount of HTML content, is sent when a user is not properly authenticated for OWA.
- Compression of XML and X-component MIME types—OWA delivers content of this type at the beginning of a session.

More recent HTTP request methods, headers, and response codes have been added as options to the Application Rules to allow full control of the HTTP traffic by the end user.

### Providing Authentication to Any Web Application

Enterprise customers increasingly look to Juniper Networks for user authentication functionality to provide secure access to Enterprise applications and HTTP(S) content. One of the challenges of, and barriers to, migrating from client/server applications to Web-based Enterprise applications is security. Authenticating users prior to allowing them access to proprietary HTTP or HTTPS applications is essential.

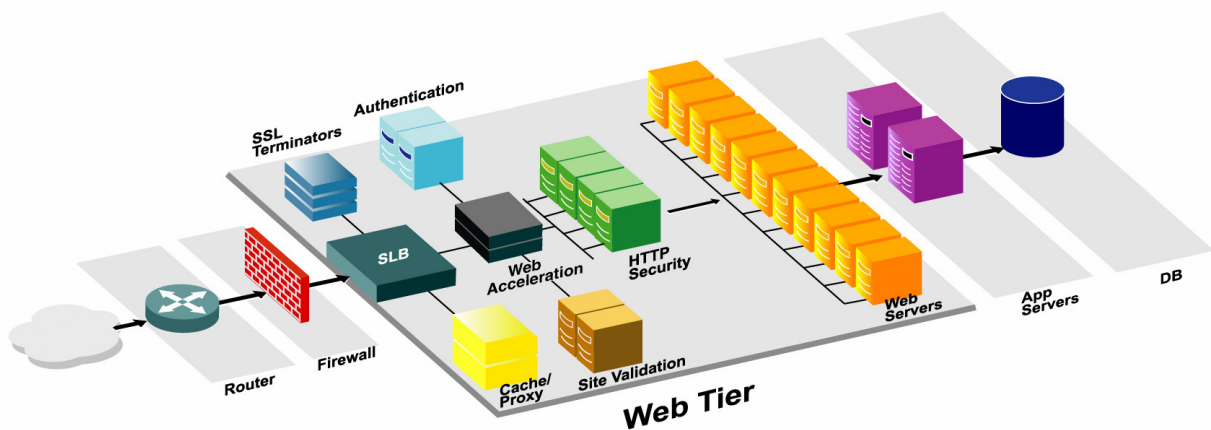
Because the DX handles all the connections to users, and delivers all of the HTTP and HTTPS traffic to users, it is logical that the DX support user authentication. The authentication methods that Juniper supports are RADIUS, LDAP, and LDAPS.

RADIUS can act as a proxy for several other authentication methods. Some commercial and non-proprietary RADIUS server software packages have the ability to query an external authentication source like an LDAP server or an RSA SecurID server. This gives the DX the ability to move into environments that use other methods of authentication while not having native support for them.

### Scaling Capacity and Providing High Availability

As enterprises grow, the requirements placed on their data centers increase. Adding capacity to current network configurations can become extremely complex and prohibitively expensive. For example, in a typical data center network (Figure 3), to add double your capacity you must add double the number of Web servers containing the content. In turn, you must double or triple the number of SSL terminators, authentication servers, Web accelerators, cache and proxy servers, site validation servers, and so forth, to support the additional Web servers. And while you have increased your overall capacity, the end user may not see any improvement in performance when accessing your Web site.

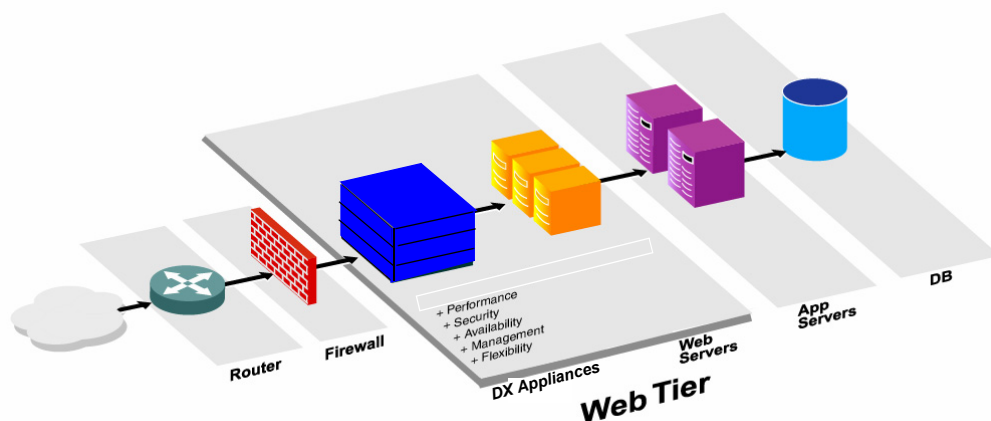
**Figure 3: Increasing Capacity in Your Data Center *without* the DX Platform**



The DX platform includes the unique ActiveN™ scaling and high-availability feature, which allows new DX devices to be incrementally added as needed to meet growing demands. More than just a high-availability system, the ActiveN feature enables the entire DX cluster to act as a single device, linearly scaling performance. Up to 64 units can be clustered in a mesh topology, and the ActiveN feature ensures that if a single DX platform becomes unavailable, the workload is automatically redistributed among the remaining units, providing N + 1 redundancy.

To double the capacity in your data center with the DX (Figure 4), you add a DX appliance for every N Web servers. For example, if you currently have five Web servers with a single DX, to double your capacity, you add five new Web servers and one new DX. Depending on the average traffic load, you may be able to add up to eight Web servers for each DX appliance.

**Figure 4: Increasing Capacity in Your Data Center with the DX Platform**



### **Increasing Web Transaction Success Rate through Caching and Application Rules**

The DX platform provides two methods for increasing the success rate of your Web transactions: symmetric caching and an AppRules Control Environment.

#### **Symmetric Caching**

The DX platform's on-board DRAM cache, coupled with the ability to rewrite HTTP content on-the-fly to enable browser-side caching, ensures that no bandwidth is wasted on downloading unnecessary objects. A 3G caching capability enables the DX platform to locally store commonly-requested objects in fast DRAM and quickly respond to requests for those objects at speeds that only silicon-based storage can achieve. The server never sees the request, preserving valuable cycles for serving dynamic content.

The DX platform's ability to dynamically rewrite content also allows previously uncacheable HTTP objects to be stored on the client browser so that the next time the object is requested, it is locally served without wasting WAN bandwidth or consuming data-center processing cycles.

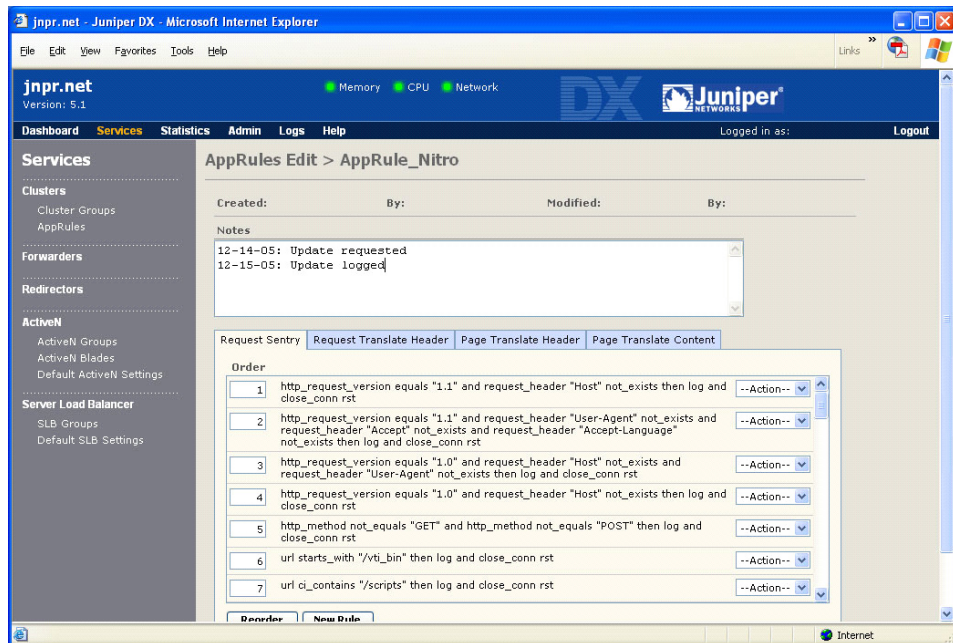
### AppRules Control Environment

The DX platform features a flexible bi-directional HTTP header and body inspection/content rewriting capability called AppRules™ which enables IT to modify application behavior on active traffic flows to compensate for inefficiencies or other problems without altering the application code itself. Users can choose from an existing template of general optimization rules or, using a simple GUI-based wizard, they can write their own custom "if-then" rules based on any combination of factors for both incoming user requests and outgoing server responses.

Some AppRules examples include:

- AutoSSL™, which automatically converts HTTP to HTTPS for complex applications such as OWA
- HTTP transaction assurance, which intercepts server errors and automatically resubmits the request to other servers
- URL rewrite, which allows IT to hide back-end directory structures for security purposes

**Figure 5: Use the AppRules wizard to modify application behavior in real time.**



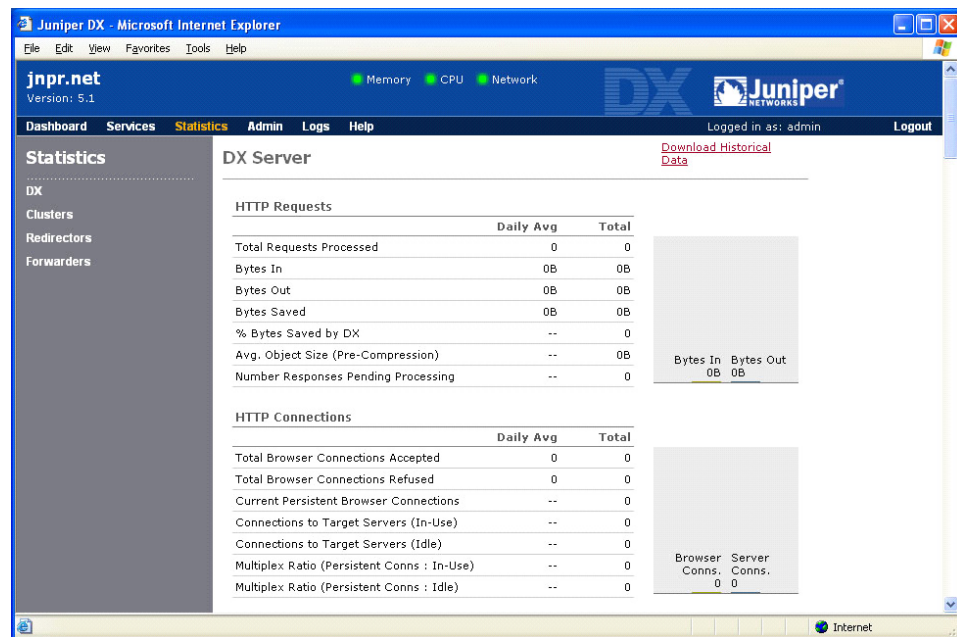
## Balancing Server Load Globally

For widely distributed environments supporting multiple data centers, the DX platform offers a global server load balancing (GSLB) feature that allows clients to be connected to the data center best equipped to fulfill their request, regardless of location. GSLB routing decision algorithms include active/standby status, closest data center, least-loaded data center, or a combination of these and other decision metrics. Acting either as a DNS transparent proxy or a full DNS BIND agent, the GSLB feature can be incorporated into any DNS infrastructure. A combination of GSLB and DX platform stickiness means clients will always connect to the same server in the same data center, ensuring server-resident client information is always available.

## Monitoring and Reporting

The DX platform provides IT with real-time and historical reports that offer a complete overview of Web-based application performance. Through the DX dashboard, users can assess the health of their data center at a glance and quickly identify and drill down to find and correct problems. All data can be exported and viewed via a Web interface, providing universal access to performance statistics.

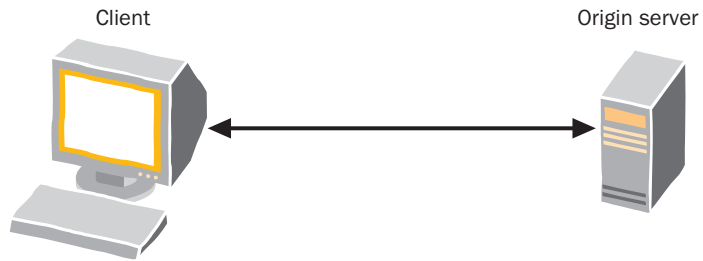
**Figure 6: Reporting capabilities provide a wealth of historical performance data**



## DX Platform Services

Without the DX, a client accesses data from a server or device in an enterprise data center through a combination of authentication servers, SSL terminators, server load balancers, proxies and so forth. The server or device with the content must respond to each client. The server can be of any type, such as Web, application, database, file, and so on. In a simplified view, the client talks to the server and the server delivers the requested content (Figure 7).

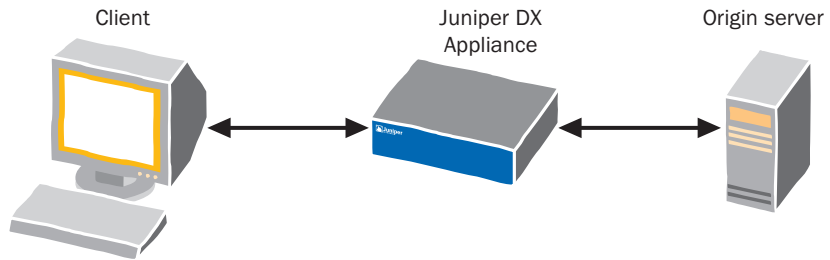
**Figure 7: Simplified View of Client to Enterprise Data Center Connection**



TTW\_029

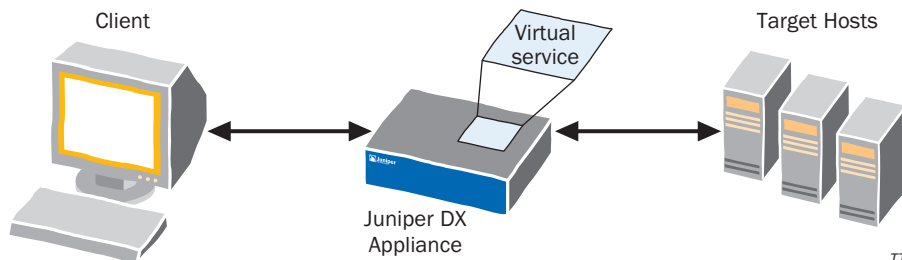
With the DX, a client accesses data from a server or device in an enterprise data center through the DX. The server or group of servers with the requested content responds to the DX. The DX platform performs the functions of SSL termination, server load balancing, proxy, and so forth. The content servers are called target hosts. Figure 8 shows the modified network view with the DX appliance inserted between the client and target host. Figure 9 shows the DX appliance communicating with a group of target hosts.

**Figure 8: Client to Enterprise Data Center Connection with a DX Appliance and Single Server**



TTW\_030

**Figure 9: Client to Enterprise Data Center Connection with a DX Appliance and Group of Servers**



TTW\_031

The DX platform offers the following services between the client and target hosts (servers in the data center) based on the type of traffic and behavior required:

- **Server Load Balancer (SLB)**—When the SLB service is configured, the DX performs load balancing for servers using non-HTTP protocols (UDP and TCP). The DX balances the traffic load by listening for incoming traffic on a specific Virtual IP address and port, distributing the traffic across the target hosts in an SLB group.
- **Forwarder**—When the Forwarder service is configured, the DX appliance performs load balancing and SSL termination for secure connections over TCP and non-HTTP protocols. It listens for incoming traffic on a specific virtual IP address and port, and then simply passes the traffic to the appropriate target host without processing it. The DX does not attempt to accelerate the outgoing traffic.
- **Cluster**—When the Cluster service is configured, the DX appliance performs server load balancing, SSL termination, and HTTP acceleration for a group of Web-based servers and applications. It listens for incoming Web traffic and distributes the traffic across a group of Web servers according to a specified load-balancing policy. The server responses are accelerated back to the requesting client.
- **Redirector**—When the Redirector service is configured, the DX appliance forces clients to send requests to an alternate server URL. It listens for incoming Web requests on a specific virtual IP address and port and redirects the client to an alternate Web server, forcing the client to resend its HTTP request directly to that URL.

## Network Topologies

---

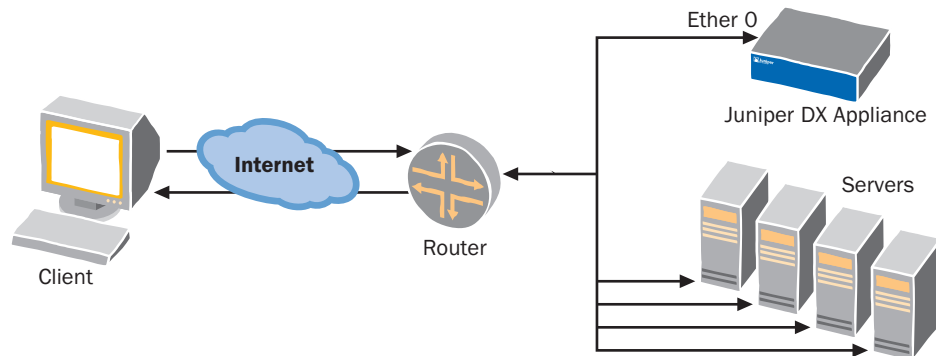
The DX Application Acceleration Platform is designed to work with any network architecture. The DX is deployed in the data center network in either a one-arm or two-arm configuration.

- “Deploying the DX Appliance in a One-Arm Topology” on page 14
- “Deploying the DX Appliance in a Two-Arm Topology” on page 14

### Deploying the DX Appliance in a One-Arm Topology

In the one-arm topology, the DX appliance is deployed on the same network as the server cluster or server farm (Figure 10). This is the simplest way to deploy the DX appliance.

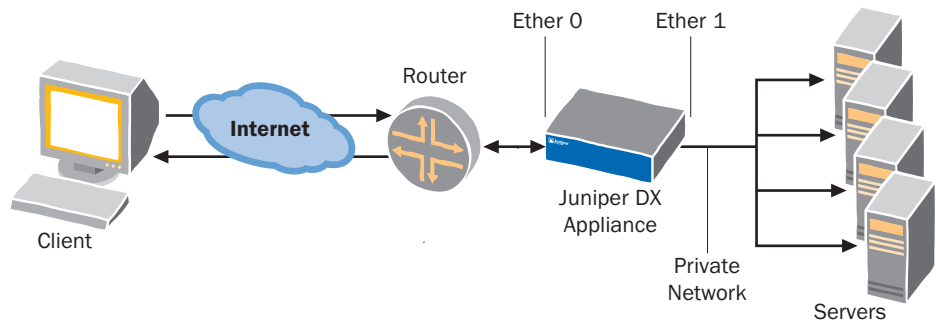
**Figure 10: Accelerating Traffic with the DX Appliance in a One-Arm Topology**



### Deploying the DX Appliance in a Two-Arm Topology

In the two-arm or in-line topology, the DX appliance is placed between the client and the server cluster or server farm (Figure 11). The advantage of this topology is that you can hide the servers behind another subnet; the DX appliance holds the public address and the servers have private addresses. In this configuration, you must change your default gateway to the DX appliance and configure the forwarding service on the DX appliance.

**Figure 11: Accelerating a Web Server Cluster with the DX Appliance in a Two-Arm (In-Line) Topology**





## The DX Product Family

The DX product family is composed of six members: the DX 3200/3280; the DX 3600/3680; the DX 3650-FIPS; and the DX 3670. All models have a serial (console) port, AC power connections and LED indicator. They are built to be mounted in a standard 19" rack.

The DX 3x00 models have software SSL and the DX 3x80 models have hardware SSL. The DX 3650-FIPS is FIPS 140-2 Level 3 compliant, offering an even higher level of security to organizations seeking compliance with the Federal Information Processing Standard (FIPS).

Table 1 describes key hardware features for each of the models..

**Table 1: Hardware Models of the DX Product Family**

Feature/Model	DX3200	DX3280	DX3600	DX3680	DX3650-FIPS
Interfaces	2 x FE	2 x FE	■ 4 x GE or ■ 2 x FE and 2 x GE	■ 4 x GE or ■ 2 x FE and 2 x GE	■ 4 x GE or ■ 2 x FE and 2 x GE
Chassis Size	1U	1U	2U	2U	2U
SSL	Software	Hardware	Software	Hardware	Hardware
DRAM	2 GB	2 GB	4 GB	4 GB	4 GB
Cache	100 MB	100 MB	800 MB	800 MB	800 MB
Flash	512 MB	512 MB	512 MB	512 MB	512 MB
Processing	1 x 2.8 GHz P4	1 x 2.8 GHz P4	2 x 3.2 GHz Xeon	2 x 3.2 GHz Xeon	2 x 3.2 GHz Xeon
Redundant Power	No	No	Yes	Yes	Yes

Table 2 describes key application features for each of the models.

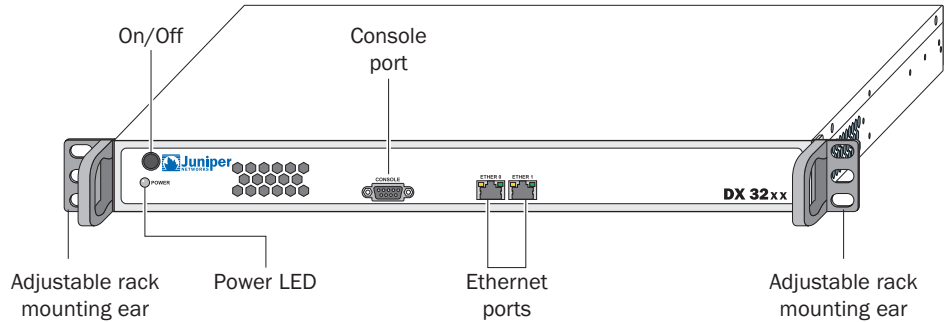
**Table 2: Application Features of the DX Product Family Models**

Feature/Model	DX3200	DX3280	DX3600	DX3680	DX3650-FIPS
Number of SLB VIPs	512	512	1024	1024	1024
Servers per VIP	32	32	64	64	64
Number of Cluster VIPs	128	128	256	256	256
Servers per VIP	32	32	64	64	64

### Chassis Connections and Labels

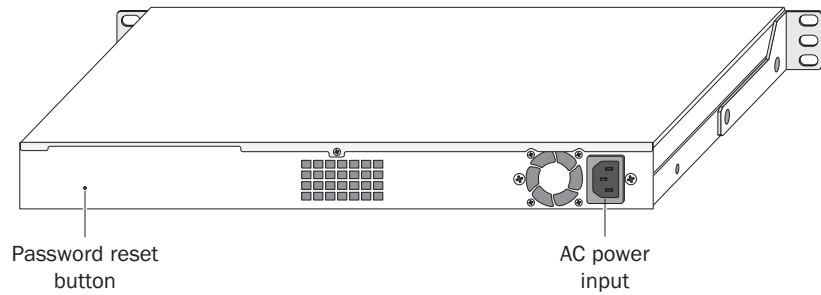
Figure 12 shows a front view and Figure 13 shows a rear view of the DXs that are available in the 1U chassis size.

**Figure 12: Front View of the DX3200/3280 Appliance**



TTW\_042

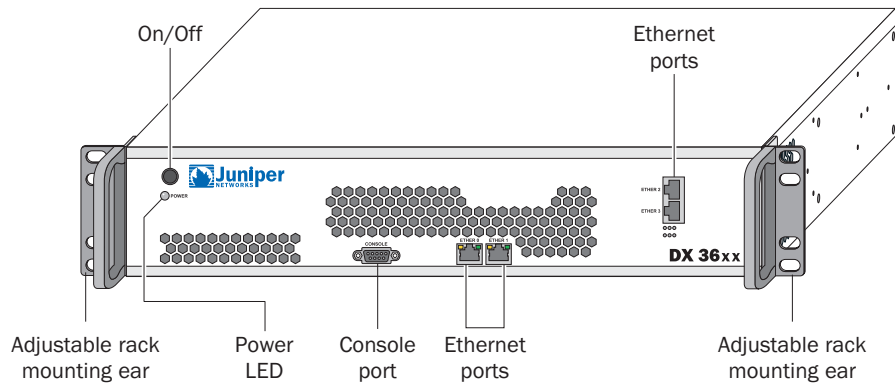
**Figure 13: Rear View of the DX3200/3280 Appliance**



TTW\_043

Figure 14 shows a front view of the DXs that are available in the 2U chassis size. In particular, it shows a model with two 10/100/1000BaseT ports and two fiber gigabit Ethernet ports.

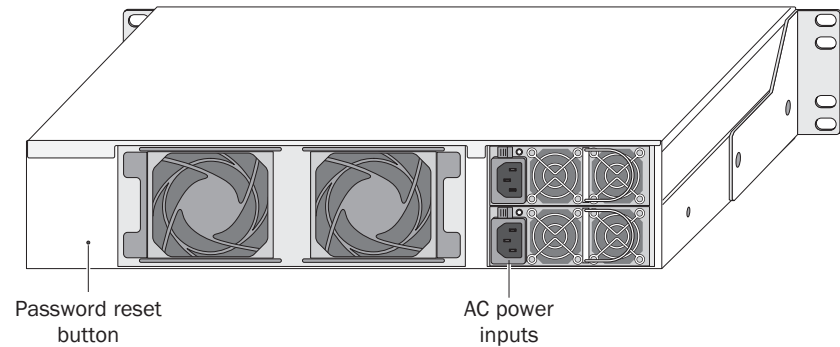
**Figure 14: Front View of the DX3600/3650/3680 Appliance**



TTW\_044

Figure 15 shows a rear view of a DX 2U Chassis.

**Figure 15: Rear View of the DX3600/3650/3680 Appliance**



TTW\_045

## DX Product Licensing Options

This section describes available licensing options for the DX Application Acceleration Platform. The version 5.1 release contains a base license and three additional license options with added functions. They are presented in the following sections:

- Overview on page 17
- DX Base License Features on page 19
- HTTP Acceleration License Features on page 21
- Advanced HTTP Acceleration License Features on page 22
- GSLB License Features on page 23

### Overview

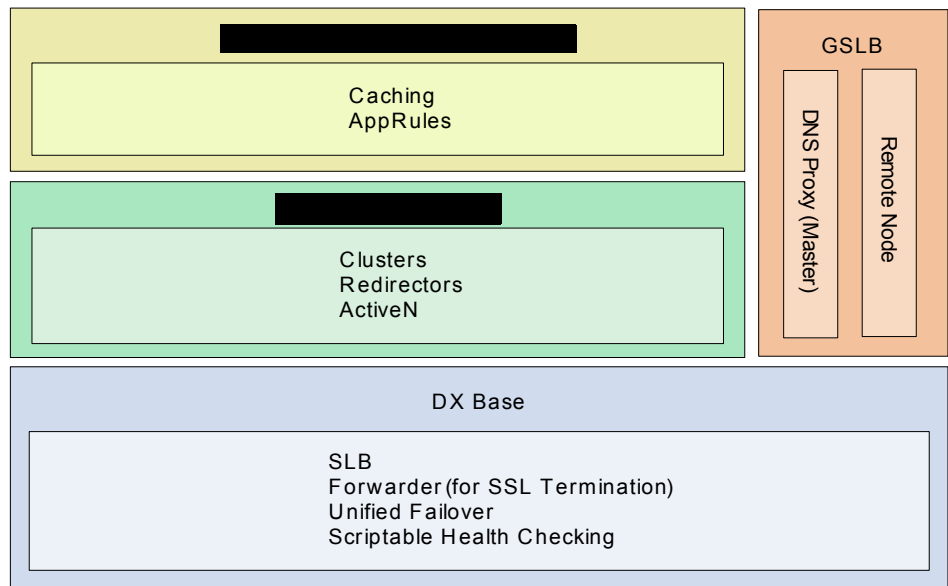
Four license options are available for the DX Application Acceleration Platform version 5.1, each offering additional levels of functionality. These licensing options are enabled through software upgrades.

- **DX Base**—This is the base license for all DX Application Acceleration Platforms. It provides Server Load Balancing (SLB) functionality, Secure Socket Layer (SSL) Termination, and the associated administrative features. This license is used when the DX appliance is deployed as a standalone server load balancer.
- **HTTP Acceleration**—This upgrade license adds HTTP acceleration and compression, including transport connection multiplexing, to the DX Base license. These additional features are used when the DX appliance is deployed for the performance benefits gained from full HTTP acceleration.

- Advanced HTTP Acceleration**—This upgrade license adds adaptive content processing capabilities to the combined DX Base and HTTP Acceleration licenses. These additional features are used when customized HTTP traffic acceleration and on-board caching is desired for increased performance.
- Global Server Load Balancing (GSLB)**—This upgrade license provides GSLB functionality to the DX Base license alone or in combination with the HTTP Acceleration license, or HTTP Acceleration and Advanced HTTP Acceleration licenses. This additional feature is typically used when load balancing and failure recovery across multiple sites is required.

Figure 16 shows the licensing hierarchy and the associated services and features that the license provides.

**Figure 16: DX License Dependencies**



**NOTE:** Refer to Chapter 7, “Administering Your DX Platform” on page 113 for information about how to obtain HTTP Acceleration, Advanced HTTP Acceleration, and GSLB licenses.

## **DX Base License Features**

The DX Application Acceleration Platform can be configured to perform server load balancing across a group of target hosts and SSL termination. The DX Base license provides customers with the following features to support this application:

- **Layer 4 TCP/UDP Load Balancing**

Identifies which application protocols are included in each packet and uses this information to hand-off the packet to the appropriate blade or cluster to balance traffic loads across a cluster of servers based on individual session information and status.

The DX platform load balances HTTP, HTTPS (SSL), FTP, and most TCP and UDP protocols using common policies, such as weighted round robin, least connections, and fail-over chaining. It also provides scriptable health checking for programmatic verification of external devices and services (ICMP, HTTP, SMTP, FTP, and so forth).

- **SSL Transport and DoS Security**

Provides secure communications between the client and origin servers (End-to-end SSL termination) and between the client and the DX appliance (one-way SSL termination).

The DX platform terminates and tunnels any TCP protocol over SSL for session-based, point-to-point, or client-to-gateway SSL security (including mail, Telnet, and so on). It supports ARC4, RSA, DES, 3DES, AES, MD5 and SHA-1 encryption algorithms. The DX platform also defends against SYN flood and denial of service (DoS) attacks. The Forwarder service can be configured on the DX appliance to direct traffic to a specified set of servers without accelerating it.

- **Half and full Network Address Translation (NAT)**

The DX appliance modifies the destination IP address (half-NAT) to enable clients to access multiple target hosts using a single public IP address. The DX appliance modifies both the destination and source IP addresses (full-NAT) to enable multiple target hosts to access the Internet using a single public IP address.

- **Failover**

Server failover occurs when a primary DX appliance fails and a backup DX appliance takes over operations. SLB failover occurs when the master DX appliance that is part of an SLB group fails and the slave DX appliance with the lowest node ID must take over as master and rebalances the traffic load across the remaining DX appliances in the SLB group. ActiveN failover occurs when the master DX appliance in an ActiveN group fails and the slave DX appliance with the lowest IP address takes over as master and owns the associated VIP address.

The new failover method enables an administrator to configure a DX appliance to perform server failover, SLB failover, and ActiveN failover with a single mechanism. Prior releases required independent configuration of each type of failure recovery method. Independent configuration will be removed from future releases of the DX Application Acceleration Platform.

- RADIUS and LDAP authentication

The DX appliance caches successful login attempts through the Remote Authorization Dial-In User Service (RADIUS) or Lightweight Directory Access Protocol (LDAP) server to reduce load on the authentication server, dramatically improving the performance of authentication and authorization

- Scriptable Health Checking

The DX appliance can examine the content from a target host to determine if it is correctly handling requests and then direct traffic according to custom-defined rules for this Layer 7 health checking.

- Administration Tools

The DX platform provides a Web browser interface (HTTP, HTTPS) to simplify configuration and management, but also supports a command line interface via SSH, Telnet, SCP, or console (RS-232). The browser interface contains an administration dashboard and provides e-mail alerts, SNMP support through Juniper MIB, and configuration synchronization across multiple DX platforms, simplifying configuration and management for large scale deployments.

The DX platform provides role-based multi-level administrative access and offers complete system, administration and audit logs. Reports can be generated showing more than 200 historical and real-time statistics, available by second, minute, hour, day, month and year

- Configuration Flexibility

The DX platform supports one- and two-arm configuration modes and client IP transparency to simplify configuration and installation into an existing network. The DX appliance can be a drop-in replacement for existing server load balancing equipment, or can complement existing server load balancing equipment. The DX can be configured in a redundant configuration SLB applications.

The DX platform offers a wide range of functionality, including load balancing, compression, SSL, TCP offload, HTML rewrite, AAA (accounting, authentication, and authorization), and application firewalls. The DX supports 802.1Q VLAN tagging to differentiate packets belonging to different VLANs in a multi-VLAN environment. All of the DX features and functions can be used simultaneously while maintaining performance linearity.

## **HTTP Acceleration License Features**

The HTTP Acceleration license adds Layer 7 SLB, HTTP traffic acceleration and compression to the DX Base license. The HTTP Acceleration license requires that the DX Base license is installed and valid and provides customers with the following additional features:

- Acceleration

The transport connection multiplexing engine of the DX platform reduces server connections by a ratio of up to 1,000 to 1. The platform terminates and persistently maintains separate internal and external TCP and HTTP/S connections (full TCP and HTTP Proxy) and multiplexes HTTP/S requests. The DX performs real-time HTTP 1.0 to HTTP 1.1 conversions and uses client connection keep-alive for quicker performance on return visits.

- Compression

Compression is supported by Internet Explorer 7.0 and Netscape 4.0, or later. The DX platform compresses all HTML, SHTML, DHTML, JHTML, PHTML, Javascript, J2EE, JSP, CSS stylesheets, XML, and SOAP protocol content. The internal compression policy engine contains more than 4,000 compression policies to ensure 100 percent page fidelity. Compression can be controlled programatically for any MIME type (such as .doc, .xls, .ppt, Flash, and so forth) on a per-object or per-object-class basis.

- Protocol Scrubbing (HTTP and TCP)

Protocol scrubbing ensures only valid, well-formed HTTP/S requests reach servers. The DX platform never passes packet fragments. It delivers full-function Authentication and Authorization using RADIUS or LDAP and client certificate authentication. The DX inspects the entire content stream, blocking, logging, or rewriting bad URLs and malicious requests. It also provides buffer overflow inspection and protection.

- Native HTTP Protocol Communication

The DX platform dynamically inspects, verifies and rewrites client requests or server responses. It can act on HTTP headers, POSTs, SOAP and HTML, JavaScript, and so forth.

- SLA Monitoring and Analysis

The DX platform tracks, monitors, and logs server response time and client download time for each HTTP/S request and response.

- ActiveN High Availability

The DX appliance uses its embedded Layer 4 switch to distribute user requests to additional DX appliances, balancing the load over multiple active blades. It supports a self-healing mesh of up to 64 DX platforms, actively processing traffic to one or more VIPs with cascading failover and linear scaling. The DX appliance can be configured for active-active or active-standby high-availability.

- Layer 7 Load Balancing

The DX platform supports full Layer 7 load balancing based on any request method, protocol version, URL, cookie, other header, POST data, header or body content, SOAP, or XML content. It uses the Juniper-patented Fewest Outstanding Requests load balancing algorithm for HTTP/S to deliver optimal load distribution and performance.

- DX Services

The DX appliance Has four services available with the HTTP Acceleration license. It uses its Cluster service to distribute HTTP traffic to a specified set of Web servers and accelerate the outgoing Web traffic to the requesting clients. It uses the Redirector service to send a client to a particular server. No processing of the traffic is performed by the DX appliance in this case. The SLB service is described in the DX base license and the ActiveN service is described above.

### **Advanced HTTP Acceleration License Features**

The Advanced HTTP Acceleration license adds customized HTTP traffic acceleration and on-board caching to further improve performance. The Advanced HTTP Acceleration license requires the DX Base and HTTP Acceleration licenses be installed and valid and provides customers with the following additional features:

- 3G Caching

The DX appliance can be configured to use internal memory-resident cache for frequently-requested content to provide improved response times and reduced network bandwidth usage for subsequent requests for the same content. Caching also improves scalability and provides complete operational transparency to both client and server.

- Application Control

While the SLB license contains selected application rules you can apply as specified, the Advanced HTTP Acceleration license lets you create new and edit existing application rules (AppRules). This AppRules control environment enables bi-directional modification of HTTP applications with a wide range of actions, including alert, block, groom, transform, repair, and rewrite on the content stream. A GUI-based AppRules wizard supports simple creation of "if/then" rules for modifying application behavior without rewriting underlying code and a rules template allows selection and enforcement of predefined AppRules. Content manipulation, such as compression and 3G Caching, can be controlled with near-infinite granularity for maximum flexibility.

- Network and Transport Security

The DX appliance can be configured with AutoSSL. This feature rewrites HTML "on the fly" to secure content without modifying the application.



- Transaction Assurance

The DX platform detects transaction errors by incorrect content within the page itself or by error code. It repairs, retries, and redirects requests, shielding users from errors and increasing transaction success rate.

### **GSLB License Features**

The GSLB license adds global server load balancing capability to the DX platform. The DX Base license must be installed and valid. The GSLB license may be combined with the HTTP Acceleration license and/or the Advanced HTTP Acceleration license.

Global server load balancing allows customers with installations in multiple sites to continue operations when one, or more, of their sites goes down. GSLB takes the failed site out of the load balance rotation until it is available again. Dynamic load balancing can be applied across several sites, distributing requests to the data center best equipped to fulfill them.

The DX Application Acceleration Platform can be configured as the authoritative device for a Virtual IP address or as a proxy filter for a DNS server. DNS resolution decisions are made based on active/standby status, closest data center to origin of request (round-trip time), least-loaded data center (bandwidth, packets, connections), DX loading (memory or CPU utilization), or a combination of these metrics. The DX can act as a full DNS proxy (using BIND) or transparent DNS cache. Communication conducted between DX appliances occurs over a secure communication channel.

### **New Features in this Release**

---

The following features have been introduced in DXOS release 5.2:

- Enhanced support for the Juniper Secure Access SSL VPN solution to enable the DX appliance to be compatible with the optimized Network Connect Protocol (NCP) for cluster configurations.
- Enhanced TCP Dump, SSL Dump, and historical and SLB statistical information to improve monitoring and troubleshooting of DX appliances and SLB services.
- Inspection and notification of configuration errors related to ActiveN group, cluster and forwarder VIPs.
- For DX cluster services, a user is no longer required to log in to each application in the enterprise within the same domain.



## Chapter 2

# Application and DX Product Concepts

This chapter describes key application acceleration concepts used when the DX Application Acceleration Platform is deployed and configured in an enterprise data center.

This chapter includes the following topics:

### ***The Basics***

- “Multi-Level Administrative Rights” on page 26
- “Server Load Balancing” on page 28
- “SSL Termination” on page 33
- “HTTP(S) Authentication” on page 39
- “Health Checking” on page 46
- “Failover” on page 48

### ***Advanced Topics***

- “ActiveN” on page 49
- “3G Caching” on page 60
- “Using Overdrive Application Rules” on page 63
- “Global Server Load Balancing” on page 66
- “Forward Proxy Acceleration” on page 74

## Multi-Level Administrative Rights

---

To enable better management and user accountability, different levels or classes of user access have been implemented on the DX Application Acceleration Platform. The classes of users are called “Roles”. The level of access increases as needed to perform various management tasks. This allows you to differentiate:

- Users versus Administrators versus Operators
- Network administration versus Security administration

Conceptually, roles can be grouped as follows:

- The user’s interaction with the DX appliance is completely passive, i.e., nothing can be changed on the DX appliance. Users can display information but can not make any configuration or operational state changes. This is useful for users in the Network Operations Center (NOC) that need to view information on all devices but not make any changes.
- Operators have access to the DX management features used for daily operations. Operators should not be allowed to make configuration changes. Operators can only view information and enable/disable services and target servers. Operators should not be able to severely impact the operation of the DX.
- Administrators are the only ones that may make permanent changes to the DX configuration. Administrators can access all the functions to configure and troubleshoot problems on the DX.

### User Access Levels

A user can be assigned to one or more roles as defined in Table 3. Access to the DX must be controlled by a unique username and password combination. Once you are connected via local console, Telnet, or SSH, you are prompted to enter a username and a password.

#### Default Account on the DX Application Acceleration Platform

The default account for the DX is:

- Username: `admin`
- Password: `admin`
- Role: administrator (see below for description)

The first time you log into a DX through the serial console port, you must log in with the default username and password. As part of the first time configuration procedure, you will have an option to change the password for the default account. It is recommended that you change the default password or disable the account after initial configuration. The default account cannot be deleted and the role cannot be changed.

If you upgrade to a newer version of the firmware from a 2.3.X or 3.0.X DX appliance, you will need to login using the default username and the same

password that you had previously defined for the default username on the DX before the installation of the new firmware.

## User Roles

Table 3 describes the various user roles available on the DX platform.

**Table 3: Roles**

Role	Description and Tasks Performed
administrator	The administrator has complete access to all DXSHELL commands on the DX. Administrators may add new users and change user attributes.
network_administrator	The network_administrator can execute all DXSHELL commands, except those related to SSL.
network_operator	The network_operator can execute all DXSHELL commands that don't change the configurations and settings to the DX, except those related to SSL. In addition, the network_operator can enable and disable the following: <ul style="list-style-type: none"> <li>■ Target Servers</li> <li>■ State of services</li> <li>■ Server</li> <li>■ Telnet</li> <li>■ Web Administration Server</li> <li>■ SSH</li> <li>■ SNMP</li> </ul>
security_administrator	The security_administrator can execute all DXSHELL commands for SSL features only.
security_operator	The security_operator can view all SSL configuration and statistics, but cannot change the configuration related to those features.
user	The user can view all status information and statistics, except SSL related information, and cannot make any configuration changes or service state changes to the DX. This is extremely useful for users in a NOC that can only view information on devices.
target_operator	The target host operator has the same capabilities as a user, but can also enable, disable, pause, or unpause a target host within a cluster.

## Valid User Names and Passwords

- Usernames and passwords are case-sensitive.
- Usernames must be between 4-16 characters long.
- Passwords must be at least six characters long.

## Server Load Balancing

---

Server load balancing (SLB) is a mechanism by which the network traffic load from clients accessing content on one or more Web or non-Web servers is distributed across all of the available servers. Balancing the traffic across servers improves the overall performance for the clients making the requests by having the nearest or least busy server handle their requests. Load balancing applies to all types of servers (application servers, file servers, database servers, Web servers, and so forth).

For example, if you have only one Web server responding to all the incoming HTTP requests for your Web site, the capacity of the Web server may not be sufficient to manage the high volume of incoming traffic as the number of client requests increases. Some users would see poor performance from the Web site in the form of pages that load slowly or actions on the page appear to happen in a delayed fashion. Enterprises solve this problem by increasing the capacity of the Web server and adding additional Web servers (creating a server cluster) to manage the increased load.

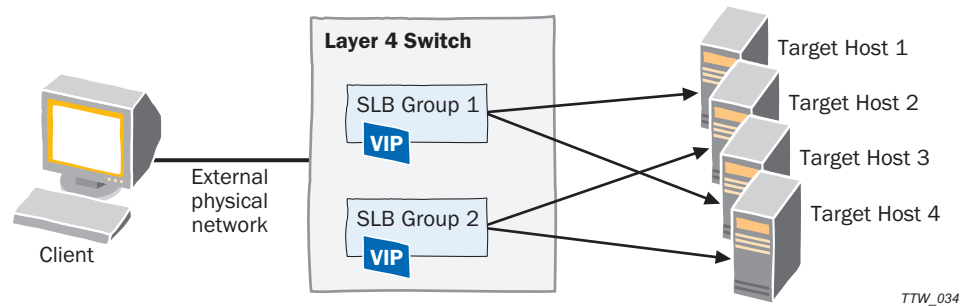
When the DX Appliance acts as an SLB, it receives requests from all of the clients wanting to access the servers in the server cluster. The DX appliance is the termination point of the client requests and issues its own requests to the various servers in the cluster. The DX can also provide scalability and failover protection in this application.

### ***SLB with the DX***

The DX appliance uses Layer 4 Switching (L4S) to load balance incoming client connections over FTP, SMTP, and other non-HTTP protocols to the target host that has the smallest load. Traffic flows undergo full- or half-NAT translations if configured.

#### **SLB Grouping**

The L4S switch has a concept of a “group” that is similar to the cluster that exists within the DX. A group represents a collection of target hosts, any one of which is capable of servicing a request. Load balancing rules are then applied to a particular group. Corresponding to each group is a Virtual IP address (VIP) which is aliased on the L4S. Multiple groups can be created. Figure 17 shows what might occur for a physical/logical combination.

**Figure 17: Server Load Balancing Groups**

TTW\_034

There is an external physical network and it refers to the target network into which the L4S is placed. In the example shown in Figure 18, the first L4S Group has two target hosts and the second L4S Group also has two.

### SLB Group Health

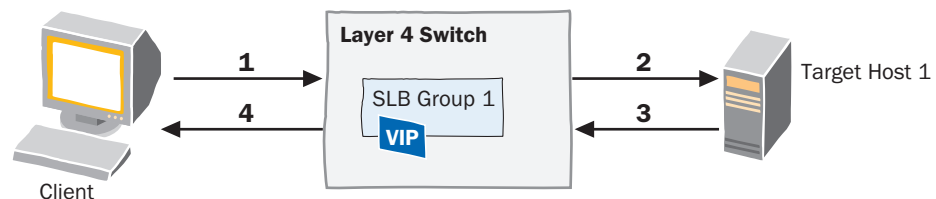
To properly balance traffic between the various target hosts, the L4S must be aware of the health of each target host and remove non-responsive hosts from rotation. The DX tries to establish a TCP connection with each target host. If the connection is successfully made, then the target host is considered operational. If the TCP connection fails, then the target host is considered down.

### Port Symmetry

With Port symmetry, the Destination IP is changed (called half-NAT), or both a Destination IP and Source IP (full-NAT) rewrite is performed.

### Connection Handling

TCP connections, as made between the client, the L4S, and the target host, are symmetric in nature. The inbound client packets are sent to the L4S and the outbound target host packets are sent via the L4S. The destination IP rewrite, source IP rewrite, or both are determined by the L4S configuration. The mechanism can be illustrated as shown in Figure 18 in full- and half-NAT configurations.

**Figure 18: NAT Operation**

TTW\_035

**Table 4: Full- and Half-NAT Operation**

Notes	Full-NAT	Half-NAT The DX Appliance acts as a Gateway for the Target Host
1	SIP = C DIP = RL_VIP	SIP = C DIP = RL_VIP
2	SIP = RL_NAT DIP = TH_1	SIP = C DIP = TH_1
3	SIP = TH_1 DIP = RL_NAT	SIP = TH_1 DIP = C
4	SIP = RL_VIP DIP = C	SIP = RL_VIP DIP = C

The L4S switch forwards packets from a client to the appropriate target hosts. The packet forwarding mechanism operates on both new connections and existing ones. For new connections, as identified by a TCP SYN packet, the L4S must determine an appropriate destination. If there is no appropriate destination (as determined by all target hosts for a Group in a non-responsive state), then the packet is dropped. The determination of an appropriate target host where the packet is to be forwarded is determined by the active load balancing scheme as applied to the Group. These balancing schemes are discussed in “Load Balancing Policies” on page 30.

The L4S also monitors the packet flow of each TCP connection to determine when to purge the L4S client connection table entries. In order to resolve a proper TCP teardown, the L4S must know whether it was the client or the server initiated the close. In both full- and half-NAT modes, the FIN and RST packets route through the L4S, and the L4S notes to which TCP session the FIN or RST corresponds. The L4S then forward the TCP session information on to the outbound gateway/router.

### Load Balancing Policies

Load balancing policy dictates how the traffic is distributed among all the active servers.



If a client’s requests are being directed to the same target host within a cluster, forwarder, or SLB group (client sticky is enabled), this behavior takes precedence over the load balancing policy.

The DX appliance supports the following six server load balancing policies:

- “Round-Robin Algorithm” on page 31
- “Weighted Round-Robin Algorithm” on page 31
- “Least Connections Algorithm” on page 32
- “Weighted Least Connections Algorithm” on page 32



- “Maximum Connections Algorithm” on page 32
- “Backup Chaining Algorithm” on page 33

### Round-Robin Algorithm

All active servers in a cluster are contained in a list. The DX reads the list sequentially for each new TCP session, sending requests to the servers in order, and returning to the top of the list when it reaches the end of the list. When more than one DX is actively accessing the server cluster in the list (it is not a redundant device), the first request is allocated to a randomly selected server in the list to prevent first requests from each device from using the same starting server. Each device then follows the sequential order to redirect subsequent requests. Once a server is assigned a request, the server is moved to the end of the list. This keeps the servers equally assigned.

For example, with a single device and three active servers S1, S2, and S3 in the list, the first request is sent to S1, the second request is sent to S2, the third request is sent to S3, the fourth request is sent to S1, the fifth request is sent to S2, and so forth. If a second DX was present, it might send its first request to S3, then S1, S2, S3, S1, and so forth.

The round-robin algorithm is better than a random allocation because the requests are equally divided among the available servers. The algorithm does not necessarily balance the traffic equally if the servers are of different capacity or if the size of the requests varies greatly.

### Weighted Round-Robin Algorithm

As with the standard Round-Robin algorithm, all active servers are contained in a list and the DX reads the list for each new TCP session that is requested. In the weighted round-robin algorithm the server chosen to receive the request is based on its assigned weight. The larger the weight assignment, the greater the number of requests that server receives. For example, if server S1 has a weight of 2 and server S2 has a weight of 1, S1 would receive twice as many packets on average as S2.

The weight can be calculated using the following equation:

$$C_i = C \left( \frac{w_i}{\sum w} \right)$$

where:

$C_i$  = Connections to Server i

$C$  = Total number of connections

$w_i$  = Weight of server i

$\sum w$  = Sum of weights of all servers

A server can be taken out of rotation by assigning a weight of zero. The weighted round-robin algorithm resolves the differences in servers that is found in the standard round-robin algorithm. It does not balance the processing time associated with the requests. Standard round-robin can be considered as a special type of weighted round-robin, where all the servers have equal weight.

### Least Connections Algorithm

The server chosen to receive a request is based on the number of outstanding active connections the DX has to each of the active servers. When a new request comes in from the client, the server with the least number of active connections is chosen.

### Weighted Least Connections Algorithm

The server chosen to receive a request is based on the weights assigned to each of the target servers and the number of outstanding active connections the DX has to each server. For example, a server with 200 outstanding connections and weight of 2 is same as a server with 100 outstanding connections and weight of 1.

The load distribution is asymmetrical owing to the difference in the weights of the server. When a new request comes in from the client, the server with the least load is chosen. The least load is determined based on the current number of active connections and its weight, and can be calculated using the following formula:

$$S_i = \min\left(\frac{C1}{W1}, \frac{C2}{W2}, \dots, \frac{Cn}{Wn}\right)$$

where:

$S_i$  = Next server to be given the new connection

$Cn$  = Current number of established sessions to nth server

$Wn$  = Weight of the nth server

$\frac{C_i}{W_i}$  = Effective number of connections to nth server

A weight of zero takes a server out of rotation, but health checking is still active. The standard least connections policy can be thought of as special case of weighted least connections, where all the target servers have weight of one.

### Maximum Connections Algorithm

As with the Round-Robin algorithms, all active servers are contained in a list and the DX reads the list for each new TCP session that is requested. In the maximum connections algorithm, the servers receive requests based on the maximum number of connections (maxconn) it can service. The DX sends the first N number of concurrent connections to the first server in the list. The second server is given the next N concurrent connections, and so on. This behavior is chained across all the active target hosts.

For example, if Target Host 1 is configured to have a maxconn value of 300, the first 300 session requests are sent to Target Host 1. Requests beyond this first 300 are sent to Target Host 2. If Target Host 2 is sequentially configured to have 500 connections, the next 300 requests plus the next 200 requests are sent to Target Host 2 and the following 100 requests are sent to Target Host 3. The range for the maxconn value is from one to 2000, with a default value of 200.

### **Backup Chaining Algorithm**

All servers in a cluster are contained in a list and the DX reads the list for each new TCP session that is requested. In the backup chaining algorithm, the first active target host (the primary server) in the list is chosen to receive the request. The primary server is always the first target server was configured. All other servers are considered backup servers.

For example, the DX sends all requests to the primary server. If the primary server fails, then the first backup server (which becomes the next active server) in the list is chosen for subsequent connections. If this server becomes unavailable, the next backup server in the list is used. The list wraps to the top when it reaches the last server in the list.

It is important to add target hosts to your cluster in the order you wish them to be used, or in order of decreasing importance. You must also specify whether you want to revert back to the original target server as soon as that server becomes available after a failure, or revert to the original server only when the backup server goes down.

### **Selecting a Policy**

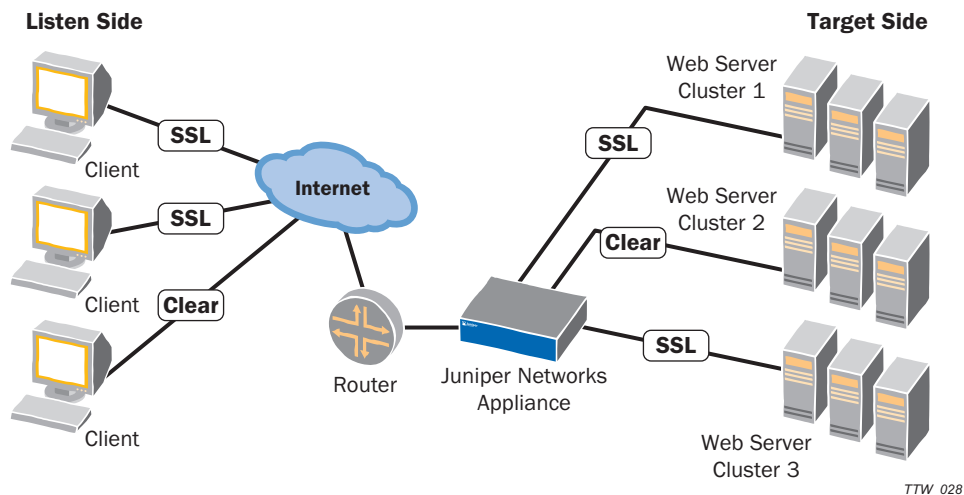
In summary, the configuration of a load balancing software or hardware should be decided based on the particular requirement. For example, if you want to load balance servers providing static HTML pages or light database driven dynamic Webpages for a given Web site, the round robin policy is likely to be sufficient. If some of the requests take longer than others to process, then one of the more advanced load balancing algorithms should be used to provide intelligent monitoring and to distribute the load based on cluster server capacity or other criteria.

## **SSL Termination**

---

Configuring the DX to serve data using SSL is easy. The DX sits in front of your server(s), holds your site certificates and keys, and processes the incoming and outgoing SSL transactions. This off-loads resource-intensive SSL processing from your servers and allows them to focus on serving content. Refer to Figure 19.

**Figure 19: Listen and Target-Side Illustration**



The communication between clients and the DX appliance (listen side) and between the server clusters and the DX appliance (target side) can be either clear or over SSL. Secure or non-secure communication is specified on a cluster-by-cluster basis.

The DX can also act as an SSL Forwarder. In Forwarder mode, the DX performs the SSL encryption or decryption, and then forwards the HTTP or non-HTTP traffic directly to the server or client. In the Forwarder mode, the client connection gets terminated at the DX, and the DX opens a new connection to the server. The DX then forwards HTTP and non-HTTP traffic transparently from the client to the server, which means it never initiates termination of a connection. That is done by either the client or the server.

**Basic Conventions and Terms**

Data travels between the server cluster and the DX, and also between the DX and the client browser. Data that flows between the DX and the client browser is termed “Listen” traffic. Data that flows between the DX and the target server cluster is termed “Target” traffic.

- LISTEN Traffic: is traffic between the DX and the client browser
- TARGET Traffic: is traffic between the DX and the server the DX is accelerating
- SSL settings for the target and listen sides are set independently

Whether the DX uses SSL is specified on a cluster-by-cluster basis. For example, for cluster 1 the DX can have SSL enabled on the listen side and disabled on the target side, while for cluster 2 the DX can have SSL enabled on both sides, etc.

With these two major divisions in mind, let's look at an already-configured server cluster named cluster 1 (dx% represents the command prompt).

```
dx% show cluster 1
Cluster [1]
Description:
Listen Address: 10.10.10.25
```

```

Listen Interface: ether0
Listen Netmask: 255.255.255.255
Listen Port: 495
Listen SSL Status: disabled
Listen SSL Protocol: sslv23
Listen SSL Certfile: sslcert
Listen SSL Keyfile: sslkey
Listen SSL Keypass: none
Listen SSL Ephemeral Keyfile:
Listen SSL Ephemeral Keypass: none
Listen SSL Ciphersuite: all
Listen SSL Cipherfile:
Listen SSL AutoChain Status: enabled
Listen SSL AutoChain RootCert: enabled
Client Authentication: disabled
CA Certfile:
CA CRL File:
CA Trust File:
Client Certificate Authentication Type: local
Client Certificate Forwarding: disabled
Client Certificate Forwarding Format: DERBase64
Listen TargetsDown Mode: blackhole
QoS : none
DSR Status: enabled
Health Check Retry: 2
Health Check Connection Interval: 1
Health Check Connection Timeout: 2
Health Check Request Status: disabled
Health Check Request Interval: 15
Health Check Request Timeout: 15
Health Check Request Resume: 1
Health Check Request Url Path: /
Health Check Request Return Code: 200
Health Check Request Size: 0
Health Check Request String:
Health Check Request User Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows
NT
Convert 302 Protocol Status: disabled
Weblog Status: disabled
Weblog Destination: syslog
Weblog Format: common
Weblog Syslog Host:
Weblog Syslog Port: 514
Weblog Batch memory allocated for this cluster (in MB): 10
Weblog Batch Copy Time 1:
Weblog Batch Copy Time 2:
Weblog Batch Copy Time 3:
Weblog Batch Copy Interval: 0
Weblog Batch Retry Interval: 60
Weblog Batch Scp Directory:
Weblog Batch Scp Username:
Weblog Batch Scp Keyfile:
Weblog Batch Host:
Weblog Batch Compression: enabled
Weblog Delimiter: space
Connection Binding Status: disabled
Rule Set File:
AppRule Processing: disabled
AppRule Limit Retry Post: 32768
HTTP Authentication Status: disabled
HTTP Authentication Method: WWW
Authentication Realm:
Authentication Response Text:

```

HTTP Authentication Protocol: RADIUS  
Authentication Redirect Status: disabled  
Authentication Redirect Page URL: /auth.shtml  
Authentication Redirect Host:  
Authentication Redirect Protocol: http  
Authentication Password MaxAge: 1  
Authentication Password MaxLength: 8  
Authentication Empty Password Allowed: disabled  
HTTP Authentication Cache Status: enabled  
Authentication Cache MaxAge: 60  
Authentication SSO Status: enabled  
Authentication SSO Domain: juniper.net  
Authentication SSO Cookie Name: DXAUTH  
Authentication SSO Cookie Expire: 6000  
RADIUS Server Key:  
RADIUS Server Timeout: 10  
RADIUS Server Retries: 3  
RADIUS Server 1 IP:  
RADIUS Server 1 Port: 1812  
RADIUS Server 2 IP:  
RADIUS Server 2 Port: 1812  
HTTP Authentication Auditing: enabled  
Audit Level: failures  
OWA Status: disabled  
Client IP Transparency: disabled  
HTTP method connect: disabled  
HTTP method extended: disabled  
HTTP method webdav: disabled  
Compression 2k\_padding: global  
Compression browser ie4: global  
Compression browser ie50: global  
Compression browser ie51: global  
Compression browser ie55: global  
Compression browser ie6: global  
Compression browser ie7: global  
Compression browser ieother: global  
Compression browser konqueror: global  
Compression browser ns4: global  
Compression browser ns6: global  
Compression browser opera: global  
Compression browser other: global  
Compression browser safari: global  
Compression cmt status: global  
Compression cmt 1: global  
Compression cmt 2: global  
Compression cmt 3: global  
Compression flushthreshold: global  
Compression force: global  
Compression http10: global  
Compression javascript: global  
Compression msoffice: global  
Compression octetstream: global  
Compression optimization: global  
Compression policy: global  
Compression shockwave: global  
Compression target mode: none  
Compression target encoding: standard  
Compression text\_css: global  
Compression text\_html: global  
Compression text\_plain: global  
Compression text\_xcomponent: global  
Compression text\_xml: global  
Customiplogheader: global

```

Forwardclientcert headername: global
sacompat status: disabled
sacompat advanced url 1: /dana/j
sacompat advanced url 2:
sacompat advanced url 3:
Sticky Method: none
Sticky Cookie Mask: ippport
Sticky Cookie Expire: 0
Sticky Cookie Passheader: enabled
Sticky Client IP Timeout (minutes): 120
Sticky Client IP Leader: none
Sticky Client IP Followers: none
Balance Policy: fewestoutstandingrequests
Balance Policy UrlHash len: 0
Targetname:
Target SSL Status: disabled
Target SSL Protocol: sslv23
Target SSL Certfile:
Target SSL Keyfile:
Target SSL Keypass: none
Target SSL Ciphersuite: common
Target SSL Cipherfile:
Target SSL Timeout: 1440
Target SSL AutoChain Status: enabled
Target SSL AutoChain RootCert: enabled
Target Local IP:
Target: 192.168.70.2:24
    Status: unpaused
    Weight: 1
    Max. Connections: 0
Target: 192.168.51.100:32
    Status: unpaused
    Weight: 1
    Max. Connections: 0
QoS : none
Cache: None.

```

The parameters in the above output are described in Table 5.

**Table 5: SSL Output Parameters**

Parameter	Description
Listen Port	Port used by the DX to communicate with client browsers. Port 443 is the standard port for SSL traffic.
Listen SSL Status	Specifies whether or not communication between the DX and client browsers use SSL. <ul style="list-style-type: none"> <li>■ Enabled—SSL is used for communication.</li> <li>■ Disabled—SSL is not used for communication.</li> </ul> <p><b>NOTE:</b> If the value is set to disabled, the next five values are ignored (Listen SSL Protocol, Listen SSL Certfile, Listen SSL Keyfile, Listen SSL Keypass, and Listen SSL Ciphersuite).</p>
Listen SSL Protocol	Protocol used for SSL communications with client browsers: <ul style="list-style-type: none"> <li>■ sslv2—version 2</li> <li>■ sslv3—version 3</li> <li>■ sslv23—version 2 or version 3. This value is typically chosen for listen traffic to allow communication with the greatest number of SSL-supported client browsers.</li> <li>■ tlsv1—version 1</li> </ul>

**Table 5: SSL Output Parameters (continued)**

Parameter	Description
Listen SSL Certfile	Name of your certificate file. democert, shown in the example output, is a sample certificate shipped with the DX. You must use demokey with democert.
Listen SSL Keyfile	Name of your keyfile. demokey, shown in the example output, is a sample key shipped with the DX. You must use democert with demokey.
Listen SSL Keypass	If your private key is encrypted with a password, enter that password here. If a password has been entered, then <b>*****</b> is displayed. Otherwise, none is displayed.
Listen SSL Ciphersuite	Specifies which Ciphersuite to use with SSL: <ul style="list-style-type: none"> <li>■ strong</li> <li>■ export</li> <li>■ common</li> <li>■ all—all Ciphersuits are accepted. This value is typically used for listen traffic to allow communication with the greatest number of client browsers.</li> </ul> For an explanation of each cipher suite and a list of included ciphers, refer to the section “SSL Cipher Suite Details” at the end of this chapter.
Targetname	Name of cluster 1. For example sss1.yourdomain.com.
Target SSL Status	Specifies whether or not communication between the DX and cluster 1 uses SSL. <ul style="list-style-type: none"> <li>■ Enabled—SSL is used for communication.</li> <li>■ Disabled—SSL is not used for communication.</li> </ul> <b>NOTE:</b> If the value is set to disabled, the next six values will be ignored (Target SSL Protocol, Target SSL Certfile, Target SSL Keyfile, Target SSL Keypass, Target SSL Ciphersuite, and Target SSL Timeout).
Target SSL Protocol	Protocol used for SSL communications with cluster 1: <ul style="list-style-type: none"> <li>■ sslv2—version 2</li> <li>■ sslv3—version 3</li> <li>■ sslv23—version 2 or version 3. This value is typically chosen for listen traffic to allow communication with the greatest number of SSL-supported client browsers.</li> <li>■ tlsv1—version 1</li> </ul> <b>NOTE:</b> It is possible to do target-side SSL without the target SSL certfile, keyfile, and keypass. In that configuration, you would have communication using in SSL, but without client authentication.
Target SSL Certfile	Name of the certificate file for cluster 1. This may be left blank if target SSL status is disabled.
Target SSL Keyfile	Name of the key file for cluster 1. This may be left blank if target SSL status is disabled.
Target SSL Keypass	Password for encrypted private key for cluster 1.



**Table 5: SSL Output Parameters (continued)**

Parameter	Description
Target SSL Ciphersuite	<p>Specifies which Ciphersuite to use with SSL:</p> <ul style="list-style-type: none"> <li>■ strong</li> <li>■ export</li> <li>■ common</li> <li>■ all—all Ciphersuits are accepted. This value is typically used for listen traffic to allow communication with the greatest number of client browsers.</li> </ul> <p>For an explanation of each cipher suite and a list of included ciphers, refer to the section “SSL Cipher Suite Details” at the end of this chapter.</p>
Target SSL Timeout	The amount of time, in minutes, to wait before the DX appliance closes the SSL session with a cluster due to inactivity. Default is 1440 minutes (24 hours).
Target Hosts	List of clusters that the DX appliance is accelerating, including whether each is enabled or disabled. If no clusters are being accelerated, “none” is displayed.



**NOTE:** When Target side SSL is enabled, the DX appliance is actually an SSL client to the target servers. Target SSL Certfile, Target SSL Keyfile, and Target SSL keypass are only used when the DX appliance must be authenticated by the target servers as a valid client.

## HTTP(S) Authentication

This chapter provides a description of HTTP(S) Authentication for the DX Application Acceleration Platform, discussing the following topics:

- Overview on page 40
- Authentication, Authorization, and Auditing (AAA) on page 40
- Authentication Methods on page 41
- Password Change Request on page 45

## Overview

Enterprise customers increasingly look to Juniper Networks for user authentication functionality to provide secure access to Enterprise applications and HTTP(S) content. One of the challenges of, and barriers to, migrating from client/server applications to Web-based Enterprise applications is security. Authenticating users prior to allowing them access to proprietary HTTP or HTTPS applications is essential.

Because the DX handles all the connections to users, and delivers all of the HTTP and HTTPS traffic to users, it is logical that the DX support user authentication. The authentication methods that Juniper supports are RADIUS, LDAP, and LDAPS. The choice of RADIUS is based on the fact that it is a well-entrenched technology that is known and deployed by many of Juniper's customers.

RADIUS can also act as a proxy for several other authentication methods. Some commercial and non-proprietary RADIUS server software packages have the ability to query an external authentication source like an LDAP server or an RSA SecurID server. This gives the DX the ability to move into environments that use other methods of authentication while not having the native support for them. It is expected that all of the other authentication methods will be supported natively on the DX as needed.

## Authentication, Authorization, and Auditing (AAA)

HTTP(S) Authentication fulfills the Authentication function in “AAA” (Authentication, Authorization, and Auditing). Authentication simply identifies a user as who they say they are. The Authorization and Auditing parts of “AAA” are not addressed by this feature.

The DX uses the collected authentication data (i.e., username and password) to satisfy the Authorization part of “AAA” by relaying this information onto the configured authentication server, along with the resource (URL). This provides you with fine-grained control of per-user access to the content fronted by the DX.

The ability of the DX to provide the Authorization part of “AAA” is dependent upon the abilities of the authentication server to provide this service. If the authentication server does have this ability, the DX will pass the collected authentication data along with the user requested resource (i.e., URL) to the authorization server for permissions analysis.

### Collecting the Authentication Data

The HTTP specification provides multiple ways to acquire authentication data from the user. The most popular method is to use the WWW-Authenticate and the corresponding Authorization HTTP headers. This method is designed to be used by the origin server. The browser provides the Authorization HTTP header for every request once the user is authenticated. (The header name is misleading when used in the context of “AAA”).

These HTTP headers are not “stackable.” Only one occurrence of each in the request headers is usable. For example, if a Web server requires authentication and it finds multiple Authorization HTTP headers in the request, should it (will it) walk through each of them attempting to find the one that works? This would lead to bad security, as well as bad performance.

Additionally, how would a browser “know” to send more than one of these headers with each request? Every time authentication is needed on a request, the browser assumes the previous credentials were not sufficient and prompts for new ones, overwriting the previous one.

Since there is not one solution that is right for all cases, the DX supports all the above methods. The configuration of these options is per-cluster.

The Authorization header is passed on to the server by default. To remove this header you must write an Application Rule. Passing the Authorization header on can be a nice feature in situations where the origin servers are required to perform authentication in addition to the DX and the authentication source is shared. This only pertains in situations where WWW-Authenticate is being used.

### **Authentication Cache**

For every HTTP request coming to an authentication enabled cluster there is a authentication request sent to authentication server. This can overload the authentication servers. Typically authentication servers can handle 100-200 requests/second but the typical HTTP traffic rate is much higher. Authentication cache brings balance by caching successful login attempts and reducing the number of authentication requests forwarded to authentication server.

The data stored in authentication cache is:

- User name and password (input from the user)
- Cluster IP address and port (input from the configuration)
- Password last modified timestamp (input from the authentication server)
- Cache entry expire timestamp (input from the configuration)

The authentication cache size is not configurable. The default size is 1 MByte per multiplexer or 2 MBytes total.

Authentication caching is global and not on a per-cluster basis. The default is that caching is enabled, and authentication caching is persistent across server process restarts.

## **Authentication Methods**

The following authentication methods are supported.

### **RADIUS**

RADIUS is the most popular authentication mechanism deployed. There are multiple commercial, as well as freely available (e.g., open source) offerings available. Each of these servers has slightly different feature sets, but they all share the core RADIUS communication functionality.

The RADIUS authentication protocol is not a perfect one and indeed is not a very secure one, but it is “secure enough.” Through the use of shared keys and simple encryption techniques, the data contained in the RADIUS data stream is not visible by sniffing the wire.

RADIUS authentication requests contain the username and password of the person attempting to gain access to the resource, a request identifier, and little else. The RADIUS authentication response contains only the request identifier and the pass/fail/error status code.

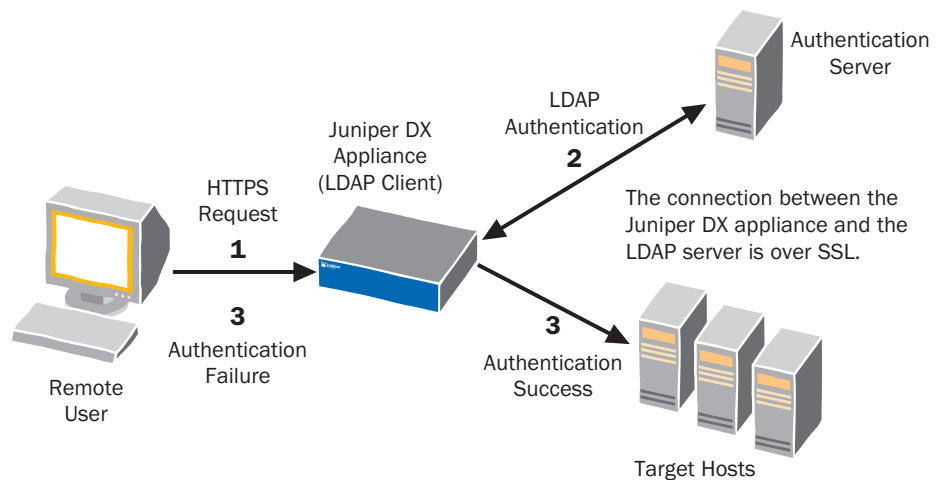
RADIUS servers that are supported include; FreeRADIUS, Cisco, Funk Software's Steel-Belted RADIUS, and possibly others.

### LDAP

LDAP has the ability to perform Authentication as well as per-user/per-URL Authorization. When the DX (LDAP client) connects to an LDAP server over SSL, the LDAP v.3 server authenticates itself by sending its server certificate to the DX (refer to item (1) in Figure 20). The DX then needs to determine whether or not the Certificate Authority (CA) who issued the certificate is trusted.

The LDAP server may also request that the client send a certificate to authenticate itself (2). This process is called “certificate-based client authentication” or “mutual authentication”. After receiving the client's certificate, the LDAP server determines whether or not the CA who issued the certificate is trusted. If the CA is trusted, the server uses the subject name in the certificate to determine if the client has access rights to perform the requested operation. In order to use SSL, you need a certificate database to hold the CA certificate and (if certificate-based client authentication is used) the client's certificate.

**Figure 20: LDAP Authentication**



**NOTE:** The DX acts as an LDAP client, and there can only be one Certificate Authority Authentication Server. This means that in network topologies with multiple clusters, each cluster must address the same Certificate Authority Authentication Server for authentication to work.

### Forward Client Certificate

At its simplest level, the DX authenticates a user over a server based upon his username and password. Using “Forward Client Certificate”, the authentication has been extended to include authentication scenarios based upon client authentication (refer to Figure 21):

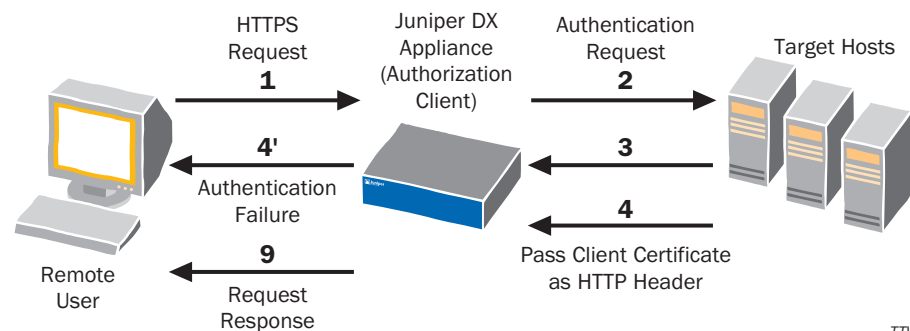
- Authenticate users against a remote server based on the presented client certificate. In this scenario, the DX appliance authenticates the user.
- Accept presented client certificate but does not authenticate locally on the DX. In this scenario, the DX is enabled for client authentication, but does not authenticate the user. The user is authenticated on the target host.
- Forward the client certificate as an HTTP header. This may apply to both scenarios already listed.

When SSL to the target host is enabled, the client certificate on the DX can be used to establish the target SSL connections. This is another way to get the client certificate to the target host. This has a downside of negating the multiplexing capability of the DX. The following situations can result:

- Clients are authenticated against a database using client certificate.
- Clients are enabled for authentication, but authentication is not done on the DX appliance; the DX appliance forwards the client certificate to the target host for the authentication.

It is possible to reuse the client certificate on the listen side to establish an SSL connection to the target side. This has a downside of disabling the multiplexing capability of the DX. Currently, when the DX has successfully authenticated the user, the HTTP request is forwarded to the target host. The DX can send the HTTP authorization header, however, the client certificate itself is not forwarded. With Forward Client Certificate the client certificate is forwarded to the target host as part of the HTTP request for further security checks including application level authorization.

**Figure 21: Authentication with Forward Client Certificate**



TTW\_018

**Use Case: Health Care Applications**

In health care applications, physicians electronically authenticate patient charts with the backend health care system. Each physician is assigned an encrypted digital signature: a secured signature password that cannot be altered or forged by another user. These certificates are used to limit access to patient information and the application logs the access, as required by the Health Insurance Portability and Accountability Act Of 1996. In order to allow Chart One applications to run unaltered when fronted by the DX, the client certificate needs to reach the backend application.

**Forward Client Certificate Features**

In order to allow downstream applications and devices to validate and authorize the user information, the following requirements are supported:

**Client Certificate as an HTTP Header**

- Allows operators to enable/disable forwarding the client certificate to the target host as an HTTP header
- Supports this capability per-cluster
- Allows operators to define the name of the inserted HTTP header
- Allows operators to choose the format in which the certificate is to be sent. The allowed options are:
  - DER format (X509 base-64 encoded)
  - PEM format.
- Supports client authentication enabled and authenticate against a remote LDAP data store using client certificate
  - Allows operators to enable/disable using client certificate for user authentication
  - Supports this capability per-cluster
  - The username/password is extracted from the client certificate
- Supports client authentication enabled and authenticate locally
- Supports client authentication enabled but don't authenticate locally
  - Allows operators to enable/disable authentication on the DX. This capability should be done in a way that the DX is able to ask for the client certificate.
  - Once the DX is able to receive the client certificate, the certificate is passed to the target host and DX acts as a passthrough.

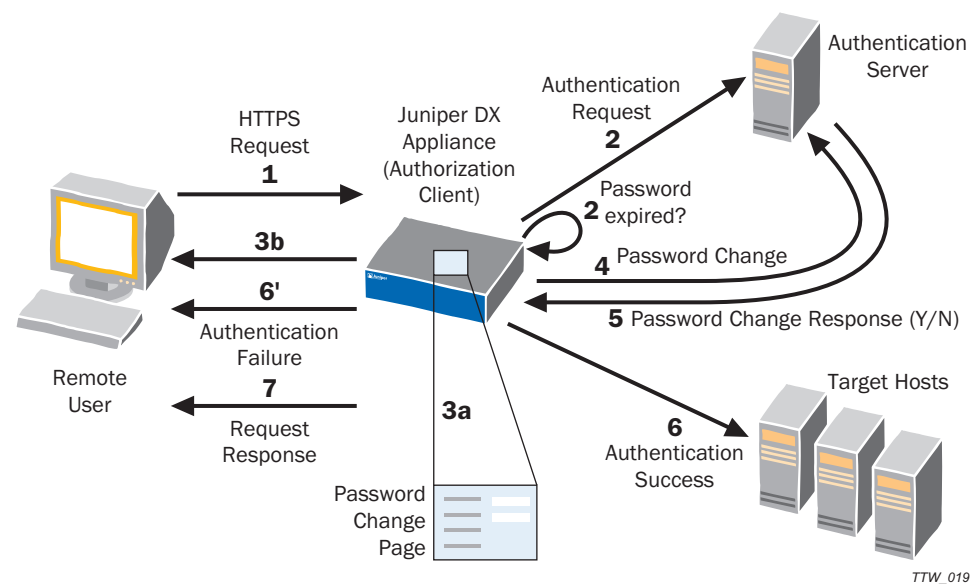
## Password Change Request

Cross-platform authentication is a single, centralized password database that can be used to authenticate users on Unix- and Windows-base systems, and other systems such as Macintosh or NetWare.

Microsoft's Active Directory is a directory based authentication system and because LDAP-based authentication is supported on the most recent Microsoft systems (including Windows 2000 and XP), and is also supported on Linux and other Unix systems, LDAP is a good choice for a cross-platform authentication system.

A typical interaction for user authentication when password challenge is employed is shown in Figure 22.

**Figure 22: Authentication with Password Change Request**



1. The user logs in by entering his username and password.
2. The DX LDAP client passes the information to the LDAP server (may be the Active Directory) and checks for the last time the password was modified.
3. If the password has expired:
  - a. The DX invokes the custom password change page with the appropriate fields. The custom page may be a local URL on the DX or a remote URL.
  - b. The user inputs the requested new password (an old password, if needed). This information is parsed by the DX.
4. The DX forwards the user's response to the Active Directory (LDAP Server).
5. The Active Directory (LDAP Server) may accept or deny the change.

6. If accepted, the request is passed to the target host and processed. If it fails, the user is notified.
7. The DX forwards the Active Directory (LDAP Server) response to client.

### **Use Case: On-Line Banking (Password Change on Password Change) Example**

An on-line banking company would like to implement a DX appliance as an intelligent Web front-end. They would also like to replace the actual infrastructure with a DX appliance. In their existing infrastructure, the user's password validity is three months (12 weeks). After 12 weeks, when the user accesses a Web page that requires authentication, the DX appliance should prompt the user with:

```
200: Password Management Page
Change Password Page Username: [User Name]
New Password:
```

At this point, the user should input his new password. The DX appliance must then send this new password to the AD server using LDAP when the user clicks on the [Submit Query] button.

### **Creating a Custom Password Change Page**

The elements of the challenge dialog are specified using DXSHELL. At a minimum, three fields are allowed for the challenge-response/password change page (refer to step 4): The Administration user should be able to specify the name for these fields. For example, when a dialog box for password change needs to be presented, the three fields may be:

- Username
- Old Password
- New Password

The challenge-response support in authentication is per-cluster is disabled by default. The write permissions are only allowed for users with the role of Administrator and Security Administrator.

## **Health Checking**

---

The DX Application Acceleration Platform provides in-band Layer 4 health checking automatically. Layer 4 and Layer 7 Out-of-Band health checking as well as scriptable health checking may also be configured.

### ***In-Band Layer 4 Health Checking***

In-band Layer 4 health checking is accomplished with the use of a standing connections count. This count indicates the minimum number of connections that must exist between the DX appliance and a target host to consider the target host up. When the number of connections to a target host falls below the standing connections count, new connections are initiated automatically. If connection failures are encountered in this process, and the target host does not respond within four retries, the target host is considered down.



### Out-of-Band Layer 4 Health Checking

Layer 4 health checking of clusters (specifically the connections between the DX appliance and target hosts) and forwarders is performed automatically, and cannot be disabled. You can configure the interval at which checking occurs, the maximum amount of time for a check, and the number of times a check can fail before the target host is considered unavailable.

### Out-of-Band Layer 7 Health Checking

The DX appliance can perform Layer 7 (L7), content-based health checking for your target hosts. With L7 health checking, the DX appliance examines content from a target host to determine if it is correctly handling requests. The DX appliance stops sending client requests to a target host that is having problems, resuming only once the target host has passed a specified number of successful health checks. L7 health checking is disabled (by default) and configured on a per-cluster basis.

When a target host is assigned to more than one cluster, the health check settings for the first cluster are used, and health check settings for the second and subsequent clusters are ignored. You should use the same L7 health check configuration for all clusters that contain the same target hosts.

Health checking can also be extended to an SMTP server. In this method, the DX establishes a TCP connection with the SMTP server and sends an initial handshake message (HELO). If the server responds with a valid response (a response code of 250), then the server is marked up. If not, the server is marked down. The same timeouts used for health checking of other ports also apply to SMTP health checking. The SMTP health checking method does *not* work with Secure SMTP. It works only with plain-text SMTP servers.

### Scriptable Health Checking

Scriptable Health Checking allows you to write Expect/TCL scripts that can dynamically pause and unpaue target hosts. For example, a script can be written to do an “HTTP GET” on a particular target host. If the HTTP result code is unexpected, the target host can be taken out of rotation. You import the script into the DX appliance, configure it for execution, and execute it.

Scriptable Health Checking requires a license from Juniper Networks before it can be used. Contact your Juniper Networks Sales Representative for information.

### Cluster Health Checking Policies

Table 6 shows how the DX appliance handles the success and failure of L4 and L7 health checking for clusters:

**Table 6: Cluster Health Checking Policies**

Health Checking Configured	Layer 4 Success	Layer 4 Failure	Layer 7 Success	Layer 7 Failure
Layer 4 only	Target host marked UP	Target host marked DOWN	NA	NA
Layer 4 and Layer 7, health check interval is same for both	No effect.	Target host marked DOWN	Target host marked UP	Target host marked DOWN

**Table 6: Cluster Health Checking Policies**

Health Checking Configured	Layer 4 Success	Layer 4 Failure	Layer 7 Success	Layer 7 Failure
Layer 4 and Layer 7, health check interval is different for each*	No effect.	Target host marked DOWN	Target host marked UP	Target host marked DOWN

\*This case is used when you want to limit the number of Layer 7 health checks, but maintain better visibility into when a target server goes down.

## Failover

Starting with release 5.1, you can specify a single failover configuration that applies to all of the following services:

- ActiveN
- Cluster, Forwarder, and Redirector (server)
- Server Load Balancing (SLB)
- Global Server Load Balancing (GSLB)

Prior to release 5.1, failover had to be configured separately for each service (and GSLB failover was not supported). To use the new failover method, you must disable the individual failover configurations for each service (the new failover method is mutually exclusive with the previous methods).

When the new failover method is activated, failover is enabled for each active service that supports it (currently the SLB, Forwarders, Clusters, ActiveN, and GSLB). The Appliance Discovery and Failover Protocol (ADFP) is used to dynamically discover all DX peers in the same network that are enabled for failover. Peers on remote networks, such as remote GSLB nodes, can be defined manually as static peers.

A master node can be designated manually or negotiated among the peers. The master node aliases the VIPs, floating VIPs, and VMACs for the other peers, which remain in standby mode. Whenever the master fails over, two SNMP traps are generated (**failoverStateMaster** by the new master, and **failoverStateStandby** by the previous master).

In general, the services run only on the master; however, if you activate both the DX server and ActiveN to perform load balancing across multiple nodes, the server runs on each node, and failover monitors only ActiveN.

### Failover Processing

When failover is activated, and a master is established with one or more peers in standby mode, any of the following events cause the master to failover to the peer with the lowest node ID:

- The master detects a link is down. The master fails over to avoid having two masters. When the peers detect the master is unavailable, one of the standby peers becomes the master.
- Any one of the supported services fails on the master.
- The `forcemaster` setting is enabled on a standby peer. Standby peer becomes the new master, unless the current master has the `forcemaster` setting and a lower node ID.
- Failover is disabled on the master, or the master fails or is shut down.

If failover detects that the server is not running, the DX waits an additional three seconds to avoid failing over during a restart.

Table 3 describes the processing that occurs during a master/standby transition.

**Table 3: Master/Standby Transition Processing**

Standby Switching to Master	Master Switching to Standby
<ul style="list-style-type: none"> <li>■ Enabled services that support failover are started --alsogslb agent and resolver</li> <li>■ VIPs are aliased by each service</li> <li>■ Floating VIPs are aliased</li> <li>■ The virtual MAC address (if enabled) is registered with the interface.</li> <li>■ SNMP <code>failoverStateMaster</code> trap is sent, and the event is logged (log identifier is FO)</li> <li>■ ADFP Active packet is sent to all the peers</li> </ul>	<ul style="list-style-type: none"> <li>■ Active services that support failover are stopped</li> <li>■ VIPs are dealiased by each service</li> <li>■ Floating VIPs are dealiased</li> <li>■ The virtual MAC address (if enabled) is de-registered with the interface.</li> <li>■ SNMP <code>failoverStateStandby</code> trap is sent, and the event is logged (log identifier is FO)</li> <li>■ ADFP Standby packet is sent to all the peers</li> </ul>

When a standby DX is trying to become the master, it waits to receive a Standby packet from the previous master. If a Standby packet is not received, the standby DX assumes the previous master is down, and the standby peer becomes the master.

## ActiveN

The DX Application Acceleration Platform performs health checks on target hosts. When target hosts fail, the DX can route the traffic to other available target hosts. However, if the DX is deployed in a standalone mode, and the DX stops responding because of network errors (or for any other reason), the target hosts (and therefore, the Web site or applications) may be unavailable to the client until the issues are resolved.

The DX can be deployed in three different topologies to increase system availability that will be discussed in detail in other sections:

- Active-Standby
- Active-Active
- ActiveN

To provide you with a better understanding of the various topologies, the “Glossary” on page 427 will provide a series of terms that will be used in the explanations.

“Configuring ActiveN” on page 239 provides an overview of the three different topologies used to increase system availability. If you would like detailed information on how high availability configurations work, refer to “Layer 4 Switching and ActiveN” on page 52.

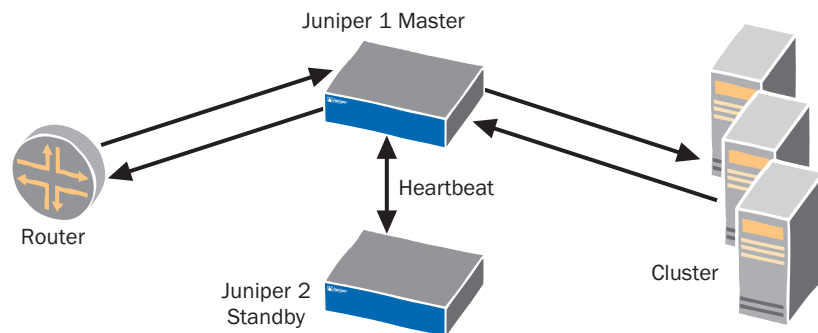
## Topologies

### Active-Standby Topology (Active One)

Active-Standby Configuration is a two-appliance configuration where one DX processes client traffic and load-balances the client requests (the active unit) while the other (standby unit) listens to the active unit’s heartbeat and waits to take over as the active unit in case the active unit fails. The heartbeat is sent between the two the DXs using Ether 0 (default) or another bind address if configured.

In the event of failure of the active unit, the standby unit detects the failure within five seconds, and then takes over as the active unit. This heartbeat interval is configurable. During the takeover, the standby DX broadcasts gratuitous ARP messages to advertise that it now owns the Virtual IP and the Virtual MAC address previously associated with the active unit. This causes any upstream routers to recognize the new interface ports and route subsequent client requests to the standby (now the active unit).

**Figure 23: Active-Standby Topology**

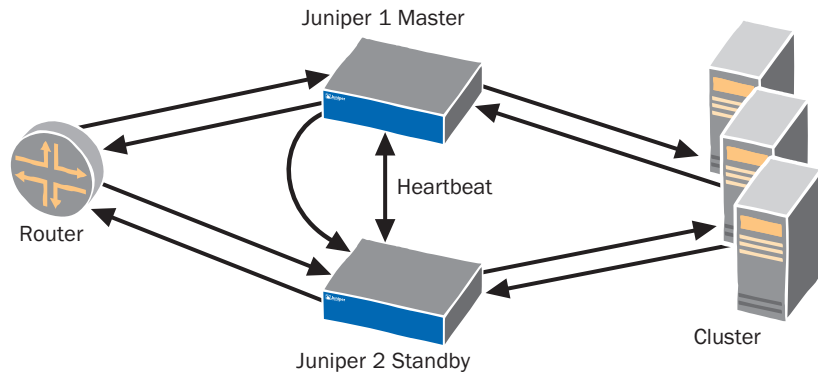


TTW\_007

While the active-standby topology is an effective way of assuring high availability of the site, it is not an efficient use of the DXs because only one of them is processing requests at any one time. An active-active or ActiveN topology is the recommended approach. For additional information on how to configure an active-standby system, refer to “Configuring ActiveN” on page 239.

### Active-Active Topology

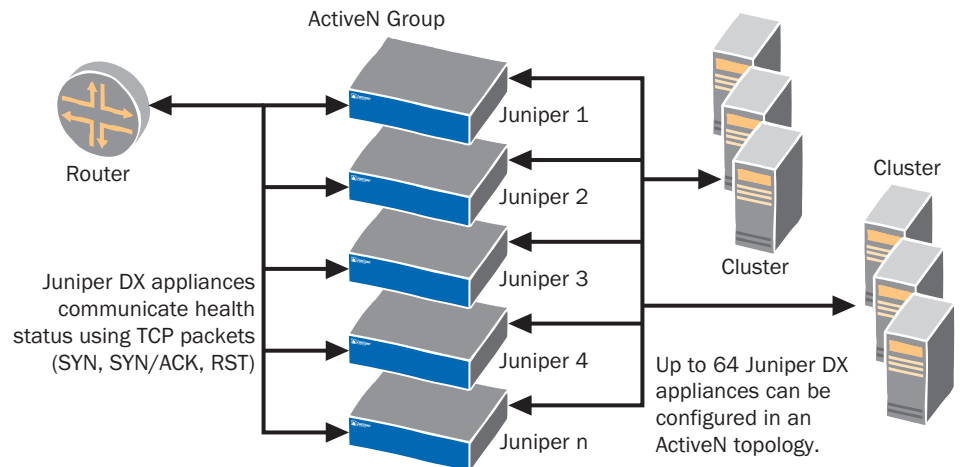
The Active-Active Configuration is a two-appliance configuration where both DXs are actively processing client traffic and load balancing the client requests. One of the DXs is the token “Master” and if the Master fails, the remaining DX takes up the Master role, taking and redistributing requests from clients. The ActiveN topology is recommended for high availability and high scalability over active-standby configurations.

**Figure 24: Active-Active Topology**

TTW\_008

**ActiveN Topology**

ActiveN is an extension of the active-active topology that allows scaling of the network. ActiveN allows up to 64 (N) DX Application Acceleration Platforms to actively process traffic destined for a VIP without the need for an external “Server Load Balancer” (SLB); refer to Figure 25. This allows horizontal scaling of DXs to process multiple gigabits of outbound response data, and enable configurations that are highly resistant to failure.

**Figure 25: ActiveN Topology**

TTW\_009

ActiveN ensures that all operational DXs continue to process traffic regardless of how many (or which) peer units are lost or disabled. ActiveN is used in network configurations where multiple DX Application Acceleration Platforms are deployed. Any one of the DXs can be the Master (or active unit) that takes the requests from clients and redistributes the traffic to the rest of the DX.

If the Master DX fails, one of the remaining DXs takes up the Master role, taking and redistributing requests from clients. ActiveN is based upon the Layer 4 switch functionality built into the DX.

## **Layer 4 Switching and ActiveN**

This chapter describes Layer 4 Switching and ActiveN for the DX Application Acceleration Platform, discussing the following topics:

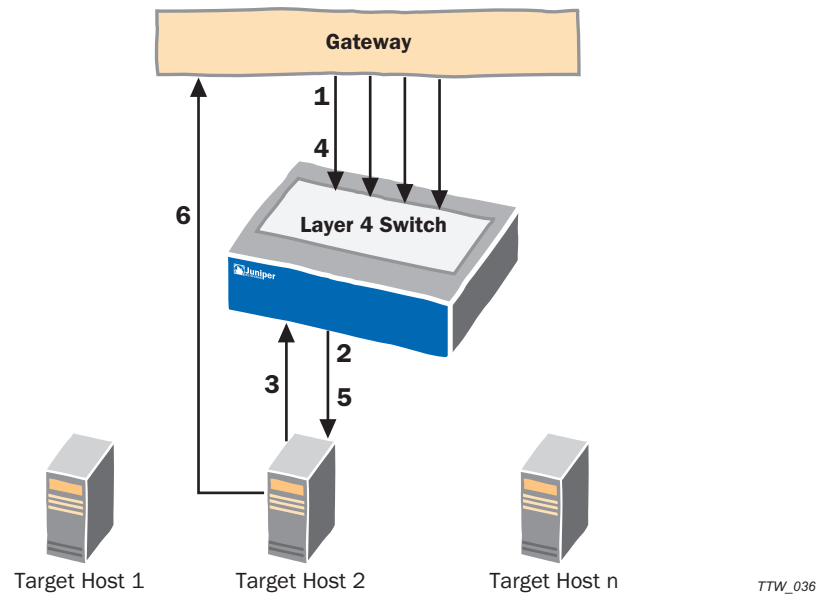
- Overview on page 52
- The Layer 4 Switch Concept on page 52
- Layer 4 Switching with Network Acceleration on page 53
- ActiveN Operation on page 55
- Client IP Sticky on page 59

### **Overview**

The ActiveN technology is based upon a Layer 4 switch that is build into each DX. A Layer 4 Switch (L4S) is a packet-based switch based on the OSI “transport” layer. Layer 4 switches identify which application protocols (i.e., HTTP, SMTP, FTP, etc.) are included with each packet and uses this information to hand-off the packet to the appropriate blade or cluster.

### **The Layer 4 Switch Concept**

Layer 4 switches are used to alleviate server loads by balancing traffic across a cluster of servers based upon individual session information and status. When an L4S is placed in front of cluster of servers running a particular application and a client makes a request for that application, the switch determines which server should handle the request, often based upon current server loads. Once the forwarding decision is made, the switch binds that session to a particular server. Figure 26 shows a typical model of Layer 4 switching with the target host configured for Direct Server Return (DSR).

**Figure 26: Layer 4 Switching Example**

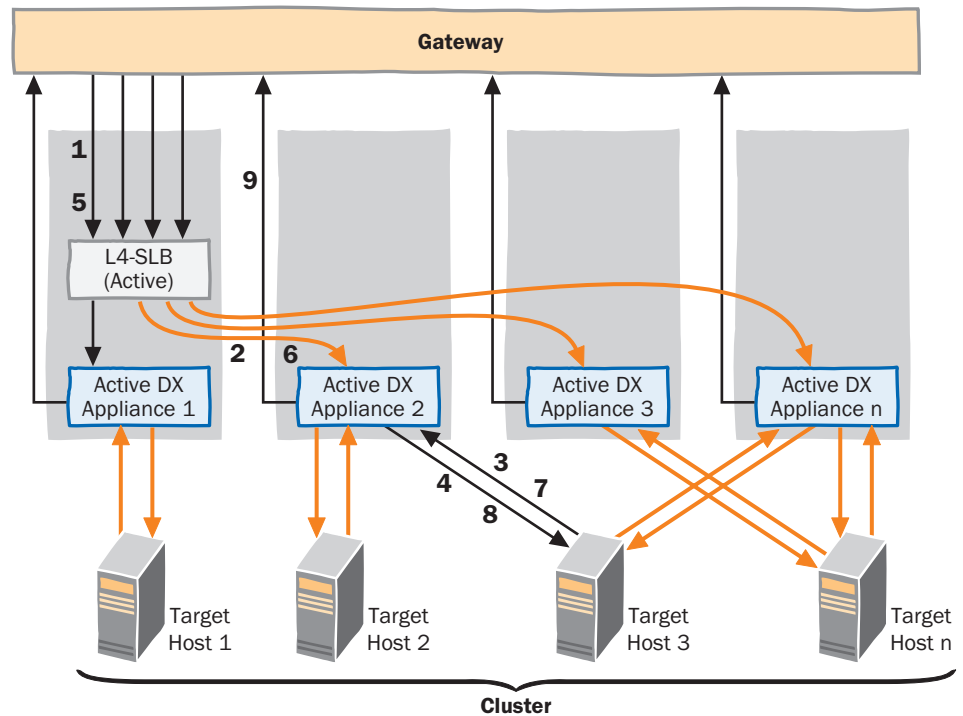
1. A request (SYN) arrives from the client.
2. The Layer 4 Switch forwards the request to the most available DX (Target Host 3 in this example).
3. Target Host 3 terminates the connection and sends an acknowledgement (SYN-ACK) to the client.
4. The client sends a request.
5. The L4S forwards the request to Target Host 3.
6. Target Host 3 sends the response back directly to the client.

While this topology improves the performance of the site by implementing load balancing, it presents a single point of failure. If the L4S malfunctions for any reason, the site goes down.

### Layer 4 Switching with Network Acceleration

Each Juniper DX has a L4S built into it. This switch can be used in front of a group of DXs to act as a Server Load Balancer (SLB). The DXs are free to perform their normal acceleration operations. Figure 27 shows a topology where the L4S within the DX is used for load balancing, and the target host configured for Direct Server Return (DSR).

**Figure 27: Layer 4 Switching with Network Alteration Example**



TTW\_037

1. A request (SYN) arrives from the client.
2. The Layer 4 Switch (SLB) forwards the request to the most available DX (the Juniper 2) for acceleration and distribution.
3. Juniper 2 forwards the request onto one of the target hosts within the cluster (Target Host 3 in this example).
4. Target Host 3 terminates the connection and sends an acknowledgement (SYN-ACK) to the client.
5. The client sends a request.
6. The L4S forwards the request to the same DX (Juniper 2) for acceleration and distribution
7. Juniper 2 forwards the request onto one of the target hosts within the cluster (Target Host 3 in this example).
8. Target Host 3 sends the response back to the DX (Juniper 2).
9. The DX sends the response directly back to the client using Direct Server Return (DSR).

This topology improves the performance of the site by implementing load balancing and acceleration, but it still presents a single point of failure. If the L4S malfunctions for any reason, the entire site goes down. This is the problem that ActiveN technology was designed to prevent.



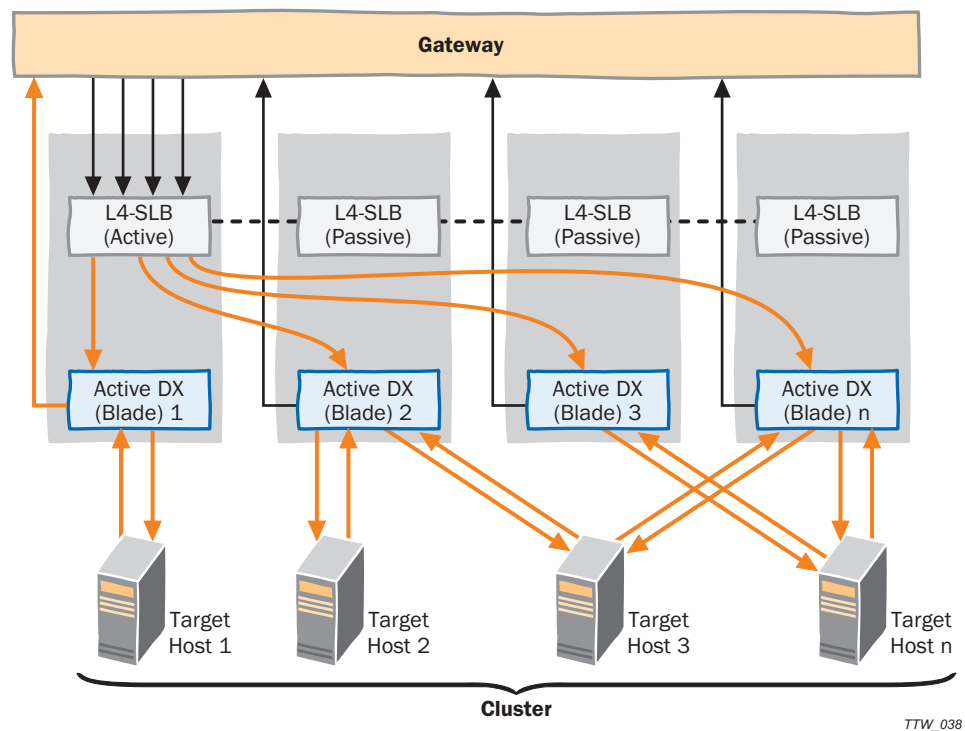
## ActiveN Operation

ActiveN is designed to improve two aspects of networks operations:

- Reliability
- Scalability

The ActiveN topology uses the Layer 4 Switch (L4S) within the DX to distribute user requests to configured DX appliances (also known as blades). An example of an ActiveN topology is shown in Figure 28.

**Figure 28: Typical ActiveN Topology**



ActiveN uses two different methods to improve network reliability and scalability; Failover at the L4S level, and Health Check at the Blade and Target Host levels.

- Failover is for L4S redundancy
- Health Check is to improve network reliability and scalability

### Failover

In a healthy ActiveN configuration only one of the L4S performs the Server Load Balancing (SLB) operations (the active L4S). The L4S in each of the backup DX (passive L4Ss) monitor the active L4S, and are ready to take over the L4S responsibilities immediately if a problem is detected. The L4S uses the same failover mechanism employed on the DX platform. The active L4S sends Redundancy Multicast Messaging Protocol (RMMP) health messages that the other L4S receives. If a certain number of health messages are not received within a time

window, the second L4S takes over the processing of new requests. (Note that the RMMP messages are actually passed at the Layer 2 level.)

The L4S uses a virtual MAC address. When the active L4S dies, the virtual MAC is removed from the interface and the backup L4S replaces it's real MAC address with the virtual one.

You can determine the failover state of a DX by typing the command:

```
dx% show activeN failover
```

For example:

```
dx% show activeN failover
Failover: enabled
Mcast addr: 239.0.0.1
Bind addr: not configured
Node Id: auto
Peer Port: 9199
Force master: disabled
Vmac: disabled
My node: 26890
Failover state: active
```

### Layer 4 Switch Health Check

In order to properly balance traffic between the various DX blades, the L4S must be aware of the health of each blade and remove them from rotation if they are not operating correctly. To monitor this, the DX watches when a TCP connection is established to each DX blade. If the connection is successful, the blade is operating. If the TCP connection fails, then the blade is considered down.

The L4S has a mechanism for finding the blades that belong to a group, determining their MAC address, and then determining their health. The user designates blades using the interface IP address/port for the particular blade (for example, "172.16.0.10:80" or "172.16.0.10:443"). This is the critical information that the L4S needs to determine the MAC address (for example, by using an ARP request to get the MAC). Once a MAC address is obtained for a blade and a successful TCP connection is established to the blade (as a health check), then the blade is officially rotated into the L4S group and it is ready to accept client requests.

You can determine the health state of a blade by typing the command:

```
dx% show activeN blade
```

```
blade 1
-----
Real IP: 10.0.201.18
Blade MAC: 0:e0:81:2e:c4:90
State: UP
-----
blade 2
-----
Real IP: 10.0.201.19
Blade MAC: 0:e0:81:2e:e2:3e
State: UP
-----
```

The line that says State: UP indicates that the blade passed Layer 2 ARP Learning.

You can determine the health state of a group by typing the command:

```
dx% show activeN group
Group an_group
Vip: 10.0.201.20
Port: 443
Sticky: disabled
Total Blades: 2
Active Blades: 2
Blades:
Index Status Local Real IP Mac
1 UP YES 10.0.201.18 00:e0:81:2e:c4:90
2 UP NO 10.0.201.19 00:e0:81:2e:e2:3e
```

**NOTE:** Layer 4 Switch Failover and Layer 4 Switch Health Check are two separate and distinct processes. You can have a situation where the L4S are all reporting that they are enabled and working (active or standby), but the health check is down because the blades are non-responsive (either not working or not enabled).

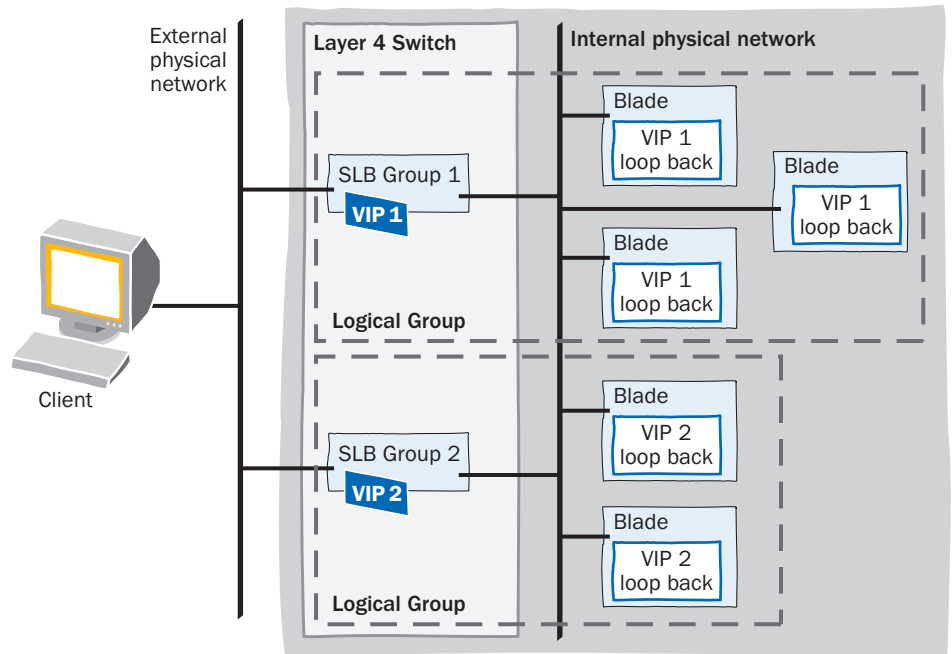
### Port Symmetry

In order to minimize the amount of packet rewriting that the L4S must perform, the ActiveN VIP must match the Cluster VIP. This allows only the MAC to be rewritten (sometimes referred to as the MAC Address Translation, MAT), instead of requiring that the entire TCP layer be rewritten. This saves the checksum overhead incurred due to port rewriting. For example, if the L4S is advertising `192.168.10.100:80`, then the DX blades in the corresponding group should be set to IP address `192.168.10.100` on loopback and listen on port 80.

### Layer 4 Switch Grouping

Within the L4S, there is the concept of a “group” that is similar to the concept of a cluster that exists within the DX. A group represents a collection of homogenous DX blades, any one of which is capable of servicing a request. Load balancing rules are then applied to a particular group. Corresponding to each group is a virtual IP address (VIP) that is aliased on the L4S. Multiple groups can be created. Figure 29 shows this from a physical/logical combination.

**Figure 29: Layer 4 Switch Groups**

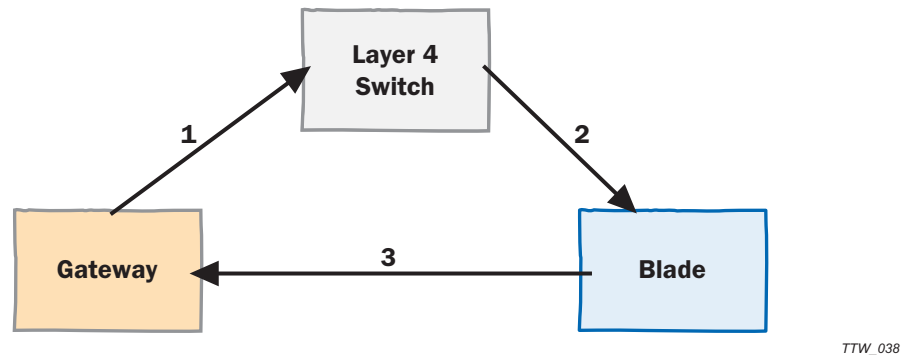


There is an internal physical network and an external physical network, where the internal network refers to the “backplane” of the DX, and the external network is the customer network into which the DX is being placed. In the example shown, the first group has three DX blades while the second group has two. The VIPs for the DX blades are placed on loopback (lo) while the VIPs for the groups in the L4S appear on the network interface.

**Connection Handling**

The TCP connections made between the client, the L4S, and the target blades are asymmetric in nature. Client packets are sent to the L4S, but outbound target blade packets are sent directly back to the client. This implements a Direct Server Return (DSR) arrangement as shown in Figure 30.

**Figure 30: DSR Operation**



where:

- 1 and 2 are client originating TCP control or data packets
- 3 is Blade originating TCP control or data packets.

The L4S forwards TCP control or data packets from a client to the appropriate DX blade (2). The packet forwarding mechanism operates on both new connections and existing ones. When a new connection comes in, as identified by a TCP SYN packet, the L4S must determine an appropriate destination. If there is no appropriate destination (as determined by all blades for a group being in a non-responsive state), then the packet is dropped.

The packet is forwarded to the appropriate blade in the group. This can be programmed to be either the blade with the least number of outstanding connections, or each blade, in turn, in a round-robin fashion. The command for setting the switching policy is:

```
set activen advanced policy <leastconn | roundrobin>
```

Each connection is uniquely determined by its layer 3 and layer 4 components. The DX uses a combination of the source IP/port and destination IP/port (although not together) to determine the appropriate destination. The first time a TCP connection comes in, the DX uses the destination IP/port to look up first a group, and then a valid target blade MAC address. Subsequent packets (e.g., not TCP SYN packets) are mapped directly to the MAC address based on the source IP/port.

The L4S also monitors the packet flow for each TCP connection to determine when to purge the L4S client connection table entries. The difficulty in doing this lies in the fact that the L4S only sees half of the TCP session (the client's packets). In order to resolve a proper TCP teardown, the L4S must know whether the client initiated the close, or the server initiated the close. The DX blades route their FIN and RST packets through the L4S. The L4S notes which TCP session the FIN or RST corresponds to, and forwards it on to the outbound gateway/router.

An aging system is also used to time-out entries in the L4S client connection table. This is because stale connections can expire due to lost hosts, etc. These stale connections accumulate over time and consume unnecessary resources.

## Client IP Sticky

Client IP Sticky refers to a property of the load balancer where the same server is chosen for multiple TCP connections when the subsequent requests come from the same client. When a TCP connection arrives on a listen VIP:Port, the DX looks in a "sticky entry table" to see if there is an entry for the client's IP address. If there is an entry present in the table, then the server is retrieved and the session is created. If there is no entry is present in the sticky entry table, then the load balancing policy is applied and the selected server is listed in the sticky table along with the client's IP address. A sticky entry is kept in the table until it exceeds the sticky timeout value set using the command:

```
set activeN group <name|all> blade sticky timeout <minutes>
```

The command "set activeN sticky timeout" is not per group, but rather it is a global command that affects all the groups.

There may be cases when a sticky entry could be deleted prematurely. One case is when the server goes down before the sticky timeout expires, and a new request from the same client arrives. In that case, the entry is flushed and a new server is fetched and re-inserted.

## 3G Caching

---

This chapter describes the caching functionality implemented in the DX Application Acceleration Platform and the associated DXSHELL enhancements. Caching stores frequently requested content in memory on the DX (in-memory cache) to provide improved response times and reduced network bandwidth usage for subsequent requests for the same content.

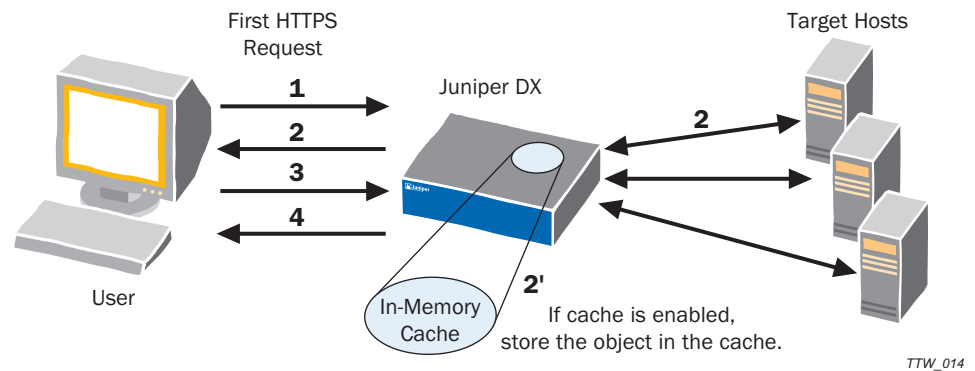
During an initial request for content (or an object), the content is requested from the origin server, read off of the server's hard-disk, and served to the client. At the same time, it is stored in the cache. Subsequent requests retrieve the content directly from the cache. Caching these requests in the DX's memory can greatly accelerate these transactions while at the same time reducing server load and network bandwidth.

The 3G Cache feature is a licensable item, and requires having an activation key in your license file. If you are upgrading from Release 3.3 and need to use 3G Cache, you must request a new license key from the Juniper Technical Support site (see "Obtaining a License Key" on page 124). Note that 3G Cache is independent of Overdrive AppRules licensing, and that AppRules are needed to populate and retrieve objects from the cache.

### **The Juniper Solution**

The DX can provide in-memory cache in reverse-proxy mode, servicing requests from clients for a large number of target hosts. The caches are typically deployed to achieve either content acceleration or to help off-load the server. When used for server off-load, the DX (and cache) is in transparent reverse-proxy mode.

In this mode, a request for content to the origin servers (1) is made to a Virtual IP Address (VIP) on the DX (refer to Figure 31). Once the DX receives the server response content (2), it simultaneously delivers the content to the client (3), and also stores the content in its cache/storage (2'). For each subsequent request (4), the response (5) comes from the cache in the DX.

**Figure 31: Cache Request Flow**

## Cache Usage Scenarios

Caching may be used in different scenarios. Some typical ones are:

- Objects are retrieved from in-memory cache - a cache-hit
- Objects are present in the memory but stale - a cache miss
- Objects are not present in the in-memory cache - a cache miss

These conditions are described in Table 1.

**Table 1: Cache Usage Conditions**

Cache Status	Object Present	Object Absent
Cache Enabled	Cache-Hit	Cache-Miss
Cache Disabled	N/A	Normal DX Operation

## Caching Features

The DX supports the following high-level caching features.

### Caching and Cache Management

Caching and cache management work in reverse-proxy mode. They allow:

- Objects to be cached in the memory
- Caches to be configured independently of clusters. They are assigned to clusters and enabled or disabled in a manner similar to that of target hosts. Several clusters may use the same cache.
- AppRules to define which objects are to be cached within a particular cluster. A caching AppRule has no effect if a cluster's cache is disabled.

### Cache Persistence

AppRules are used to specify the lifetime of the cached objects (i.e., how long to save objects in-memory).

**Cache Storage**

The cache storage is in-memory, and this implementation provides end users with high performance and great reliability.

**Transparency**

Clients to the DX are not aware that the DX is caching objects. The flexible controls of the AppRule framework allow administrators to use caching with applications that are incompatible with typical general-purpose caches.

**Cache Load Balancing**

Cache load balancing is unnecessary since the cache is in-memory only.

**Cache Statistics**

The following classes of statistics are provided:

- Cache Operational Statistics: Memory usage and other relevant data necessary to monitor the “health” of a cache
- Cache Content Statistics: Object sizes and hitcounts
- Cluster cache-usage: HTTP and I/O statistics similar to target host stats
- Cluster AppRule Stats: Cluster statistics with caching AppRule usage

**Cache Placement and Expiration Policy**

The AppRules are used to specify which objects to cache, and for how long. Refer to “Show Cluster Cache Commands” on page 293 for additional information.

**Multi-Encoding**

The cache is capable of not only storing objects in their native format (i.e., HTML, a text document, etc.), but also “derived” formats as well. In particular, it has the ability to store objects in the cache that have been compressed and processed by Page Translator Content rules. These derived formats are called “encodings.” This allows for higher throughput because the effort to repeatedly produce a compressed version of a cacheable object is no longer required.

Internally, this means that a single cache entry may actually be stored in multiple encodings. Because of this, a single entry may take up more room in the cache than its literal byte count may imply. For example, if a 30K page is stored in the cache in its native, uncompressed format as well as in its compressed encoding format, there is more than 30K of the cache's memory consumed. However, if clients only ask for a derived format, then only that format is stored. This means that if all browsers to make requests through a DX appliance support compressed documents, then only compressed documents will be stored in the cache. This has the effect of using less space in the cache than would otherwise be required if the document was stored in its native format.



## Using Overdrive Application Rules

---

This section provides an overview of the OverDrive Application Rules Translator (AppRules) feature. It describes the operation, grammar, and limitations of the feature. Configuration and management of application rules is described in Chapter 18, “Configuring OverDrive Application Rules” on page 297.



**NOTE:** AppRules is an optional feature and it requires a license key. See “Obtaining a License Key” on page 124 for details.

---

This section contains the following topics:

- “Basic Application Rule Concepts” on page 63
- “Types of Application Rules” on page 63
- “Application Rule Grammar” on page 65

### Basic Application Rule Concepts

Application Rules are simple rules written in plain language that are used to programmatically describe real-time changes that made to requests and replies passing through the DX. This allows flexible applications that allow sites to respond to changing business needs. With AppRules, you can make automatic changes to user requests without making expensive changes to back-end applications. For example, you can route all requests for pictures (gif or jpg) to a particular server.

Application Rules also ensure request completion by re-initiating a request sequence based on parameters such as incoming client request headers, server response information, or response content from the application. For example, you can automatically initiate a retry if there was an internal server error (HTTP error 500) or if the application returns a particular keyword such as “Unavailable” in its response.

### Types of Application Rules

Application Rules are segmented into various types based on how and when they are processed within the DX appliance. Application Rules can be applied to either incoming requests from the client, or to outgoing data or responses from the servers. When examined from a very high level, rules are either oriented around security-based connection management or around request and/or reply translations.

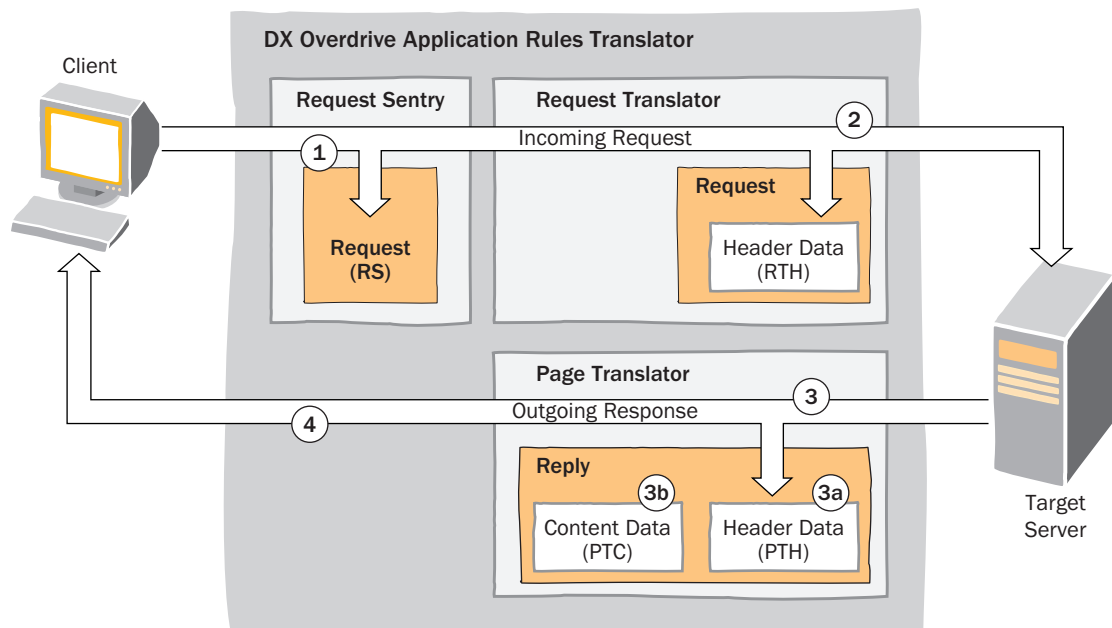
There are four types of application rules:

- Request Sentry (RS)—operate at the connection level by allowing, denying, or possibly redirecting a request based on certain criteria.
- Request Translator Header (RTH)—operate on the HTTP header segment of the incoming request. This includes the URL and query string, along with the headers that may be part of the request.

- Page Translator Header (PTH)—operate on the outgoing replies at the header level.
- Page Translator Content (PTC)—operate on the outgoing replies at the content level.

These are shown in Figure 32.

**Figure 32: DX Appliance Processing of the Various AppRules Types**



**AppRules Process Flow**

When the various types of AppRules are configured, the DX appliance processes incoming requests as follows:

1. Incoming requests enter the Request Sentry (RS). The RS acts as the security gateway for all incoming requests, ensuring that they conform to criteria defined in the RS AppRules.
2. If the request meets the criteria of the RS AppRules, then the request is passed to the Request Translator. Actions are performed on the request header data, as defined by the Request Translator Header (RTH) AppRules and the request is delivered to the target server.

3. After the request has been processed by the target server, the outgoing response is passed to the Page Translator. The Page Translator has two components:
  - a. Header Translation—Page Translator Header (PTH)
  - b. Content Translation—Page Translator Content (PTC).

Actions are performed on the response header data and content data, as defined by the PTH and PTC AppRules.

4. The response is then returned to the client that made the original request.

### **Application Rule Grammar**

Application rules are roughly comprised of two parts—test conditions and actions. A single rule can contain multiple test conditions and multiple actions (although some rules only allow a single action). Actions are executed only when all of the test conditions have been met.

A “rule” is created when test conditions and actions are placed together. The rule defines what pieces of data can be analyzed (test variables), how they can be tested (test conditions), and what to do when the tests are true (actions). For example, a rule might be: “If the URL equals /index.html, then redirect the request to server <http://www.myserver.com> using the same URL as the one supplied in the request as the redirect URL”.

The basic syntax for application rules is as follows:

```
<rule_type>: <test_condition> [and <test_condition>...] then <action> [and <action> ]
```

where:

- **<rule\_type>** is a mnemonic indicating the application rule type, RS, RTH, PTH, or PTC. It is followed directly by a colon.
- **<test\_condition>** specifies a particular test condition statement. Multiple test conditions may be applied. The keyword **and** separates multiple conditions. See “Test Conditions” on page 298.
- **<action>** designates the action that is performed when all test conditions for a certain rule have been met. Some application rules only allow one action, and some allow multiple actions. The keyword **and** separates multiple actions. See “Actions” on page 304.

It is customary to separate each logical component by some amount of arbitrary whitespace, although this is not required. Single line comments can be placed in the ruleset by placing a pound sign (#) at the beginning of the line.

For example:

```
# This is a comment.
```

For detailed information about writing and using the OverDrive Application Rules Translator, see Chapter 18, “Configuring OverDrive Application Rules” on page 297.

## Global Server Load Balancing

---

Corporate network installations with multiple sites can use the DX platform Global Server Load Balancing (GSLB) feature to continue serving client requests when one (or more) of their sites goes down. GSLB automatically takes a failed site out of rotation until it is available again. The DX Appliance Acceleration Platform has the added benefit that it can dynamically load balance across several sites.



GSLB is an optional feature that requires a license. Contact your Juniper Sales Representative to obtain the license for this feature.

---

GSLB is important in installations with multiple, replicated sites that are distributed across different physical data center locations. For instance, you might have data centers in New York, Chicago, and Seattle, and want to seamlessly shift clients among the centers. If a center goes down, requests are automatically shifted to the remaining centers without any intervention on your part. When the center returns to service, it is re-entered into the mix.

GSLB is implemented by manipulation of DNS records. Remote clients contact the site's DNS server, ask for a hostname, and receive a response containing a list of IP addresses of servers that host the site. The clients then attempt to contact the first host on the list. If that host does not reply, the client then tries the next, continuing in this fashion until either a successful contact has been made or the client reaches the end of the list. Global load balancing can be achieved by manipulating the order of the addresses in the list.

In the simplest case, the DNS server knows nothing about the state of the hosts it is serving and simply follows a pre-determined algorithm. The most common algorithms used are the round-robin algorithm that rotates among a pre-defined set of hosts or the random algorithm that randomly selects the hosts. By adding a mechanism by which the DNS server can check the health of the hosts, the algorithm can be extended to perform more advanced load balancing, and even be used to cleanly remove hosts from service. This mechanism offers advantages during both planned events, such as scheduled downtime, and unplanned events, such as a network outage.

### **GSLB with the DX**

The DX Application Acceleration Platform acts a DNS proxy that collects health and performance information with a variety of load-balancing algorithms to determine where to direct client requests. Where the DX appliance resides in the network, and its GSLB features are described in the following sections:

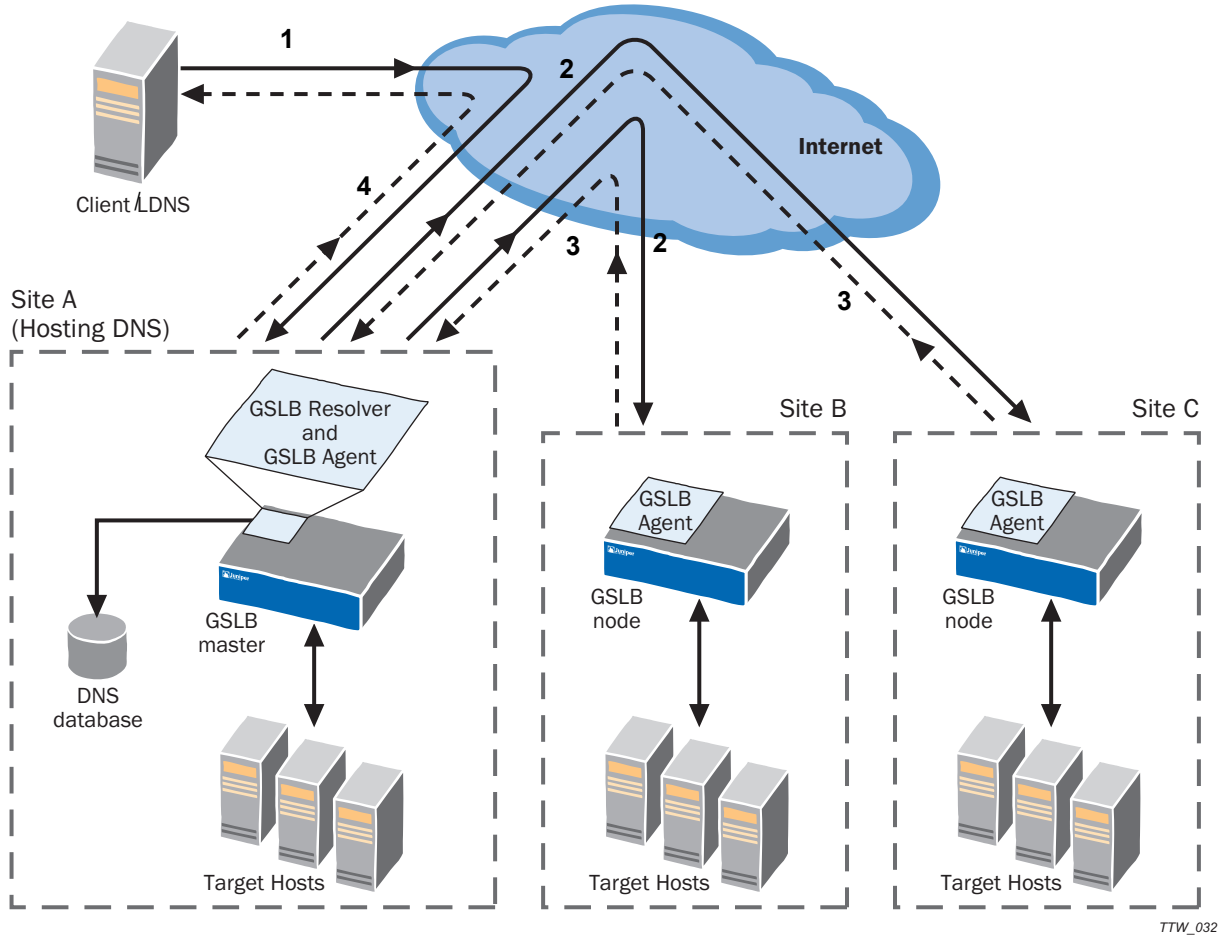
- “Network Topology” on page 67
- “DNS Proxying with the GSLB Resolver” on page 68
- “Health Checking” on page 69
- “Global Load-Balancing Policies” on page 70
- “Tracking Performance with Statistics” on page 72
- “Deployment Considerations” on page 74

### Network Topology

For a GSLB application, a DX appliance is deployed in each data center and each manages a number of target hosts. All of the hosts provide the same function, serving a particular Web page for example. One of the data centers contains the DNS master for a domain and all DNS requests are sent to it. The GSLB license is activated on the DX appliance at this site. This DX appliance acts as a DNS name server and is the *GSLB master*. The GSLB master is configured to communicate with other DX appliances, *GSLB remote nodes*, at the other data center sites. The GSLB master contains a *GSLB resolver* that answers DNS requests. The GSLB resolver can also be configured to host DNS records itself or pass the requests to a target DNS server, such as a local DNS server (LDNS). A *GSLB agent* can run on every GSLB node and sends health and performance statistics to the resolver. The DX appliance that is the GSLB master can also be a GSLB node. See Figure 33.

In Figure 33, a client requests access to a host for a Web site. The client's LDNS asks the GSLB master in site A for the IP address of the host. The GSLB master sends a DNS response with the IP address of the GSLB node with the best match. This address is based on the health information and load-balancing method provided by each GSLB agent.

**Figure 33: The DX Appliance in a Global Server Load Balancing Application**



TTW\_032

**DNS Proxying with the GSLB Resolver**

The GSLB resolver on the GSLB master listens on a Virtual IP address (VIP) for DNS requests and forwards them intact to the real DNS server, similar to the manner in which the server forwards requests from a listen VIP to a target host. If you elect to use the internal DNS server (see “Configuring the Local DNS Server” on page 346), the target server enters loopback mode, and BIND is used, listening on loopback.

The GSLB resolver listens for both TCP and UDP requests and can forward them to the real DNS server. For instance, if the client makes a UDP request, the GSLB resolver forwards the request to the DNS server as a UDP request.

Requests going to the DNS server are inspected and handled based on the kind of request it is:

- Request is an address record lookup request—A further check is performed to see if the name is one of the configured *GSLB groups*. A GSLB group is a single DNS hostname that can be resolved to multiple IP addresses (of the GSLB master and nodes specified in the group).
  - Request is one of the GSLB groups—The GSLB resolver composes the DNS response directly based on the current health status of members in the GSLB group and the global load-balancing algorithm selected.
  - Request is not one of the GSLB groups—The request is forwarded intact. Responses from the DNS server are passed directly to the client; no caching of DNS responses is performed.
- Request is not an address record lookup request—The request is forwarded intact.

Requests can be made using the UDP and TCP protocols. If the request is made using UDP, the GSLB resolver responds with a UDP response. UDP requests are limited in size, as defined by RFC 1035. If a UDP response is larger than 512 bytes it is truncated and a truncation bit is set in the DNS packet header. When this bit is set, the client should retry the request as a TCP request. If the client chooses to make a second request, this is considered to be a completely separate transaction, and the new response may be slightly different, depending on the group members' health and the progression of the load-balancing algorithm.

If the request is made using TCP, the GSLB resolver responds with a TCP response. TCP DNS requests have a maximum length of 65,535 bytes.

### Health Checking

Health checking in the form of a reachability check (ping) is used to verify the the availability of the IP addresses of GSLB group members. The GSLB resolver sends each ping asynchronously. Each IP address receives a ping every second, and must reply within three seconds to be considered available. If the three second response time expires, the IP address is sent another ping, and given another three seconds to reply. If no reply is received, a third ping is sent. If the node fails to respond to this final ping within three seconds, the IP address is considered to be down.

Health checking requests are handled in a variety of ways based on the current situation and availability of the members of the GSLB group:

- IP addresses that are members of more than one GSLB group are checked for availability only once. For example, if a member IP appears in five groups, it is only sent a ping once every three seconds; not five times every three seconds.
- Health checking continues on IP addresses that are considered to be down. As soon as it responds to a ping, it is considered to be available.
- If a request is made on a GSLB group that has no members configured, the GSLB resolver forwards the request to the upstream authoritative DNS server and responds back to the client with the response from the upstream DNS server.

- If a request is made on a GSLB group that has one or more members configured and all of the member IP addresses fail health checks, the following occurs:
  - If an IP address is set for the GSLB group indicating where to send failed requests, the GSLB resolver responds with that address.
  - If an IP address is not set for failed requests, the resolver responds with an answer that does not contain any A records.

Once all members of a group have had their availability checked, the group's load-balancing algorithm is used to determine the final ordering of members in the DNS response.

### **Global Server Load Balancing**

The DX platform can be configured to provide load balancing based on the following two factors:

- Performance metrics—The client is directed to a DX appliance based on selected performance metrics, including load, network bandwidth, and availability.
- Proximity—The client is directed to a DX appliance based on selected performance metrics and the lowest Round-Trip Time (RTT) measured between the DX appliance and the client's LDNS.

### **Global Load-Balancing Policies**

The DX platform supports five GSLB policies: Four take place strictly at the GSLB Master.

- Round robin—IP addresses within the GSLB group are returned to LDNS clients in a sequential fashion, with each request getting the next IP in the group. This corresponds to the "cyclic" setting in a conventional DNS server such as BIND. Ping checks are made to each IP within the group, one per second. If the IP fails to respond to three ping checks in a row, it is considered to be down and is removed from rotation until it responds successfully to three ping checks. Weights assigned for weighted round-robin load balancing are ignored if this is the round robin policy is selected.
- Weighted round robin—IP addresses within the GSLB group are returned to LDNS clients in sequential fashion similar to the standard round robin policy with the exception that weights are assigned to each each GSLB node (member). An IP address is served the number of times specified by its weight before moving on to the next IP address. Ping checks are made to each IP within the group, one per second. If the IP fails to respond to three ping checks in a row, it is considered to be down and is removed from rotation until it responds successfully to three ping checks.
- Random—IP addresses within the GSLB group are returned to LDNS clients in a random order. Ping checks are made to each IP within the group, one per second. If the IP fails to respond to three ping checks in a row, it is considered to be down and is removed from rotation until it responds successfully to three ping checks.



- Fixed—IP addresses within the GSLB group are returned to LDNS clients in the order they were configured.
- Forward—No load balancing or health checking is performed. Every request is forwarded to the specified Target DNS.
- Performance Metric—IP addresses within the GSLB group are returned to LDNS clients based on selected performance metrics. This active load balancing is accomplished by collect statistics from the GSLB agents on each GSLB node. The results of the metrics received determines the IP address to return. See “Metric-based Global Server Load Balancing” on page 71 for details.



At startup time, all GSLB nodes are considered to be down until responses to pings are received. For this reason, it may take a few seconds for the GSLB nodes to appear in DNS queries.

### **Metric-based Global Server Load Balancing**

When the metric-based policy is specified for a GSLB group, the GSLB master collects selected performance metrics from the participating GSLB remote nodes (DX appliances). See Table 2.

**Table 2: Performance Metrics Used in Global Server Load Balancing**

<b>Metric</b>	<b>Description</b>	<b>Range</b>	<b>Default</b>
Connections	Maximum number of connections on the GSLB remote node. Includes all client and target host connections for all cluster, forwarders, and redirector services on the node. It does not include health-checking or SLB connections.	0–4294967295	0
SLB sessions	Maximum number of unused Layer 4 sessions within the SLB service on the GSLB remote node.	0–4294967295	0
Network interface usage	Maximum amount of network traffic load (in KB or MB) handled by the GSLB remote node. Based on the busiest interface input or output data rate. Ignores rate-limiting applied with a license key.	0–4294967295	125000000
Memory usage	Percentage of memory usage on the GSLB remote node.	0–100	80
CPU usage	Percentage of CPU throughput in use on the GSLB remote node.	0–100	80
Target host availability	Minimum number of target hosts available on the GSLB remote node. Includes unpaused and available target hosts for all clusters and forwarders on the GSLB remote node.	0–100	0
Round Trip Time (RTT) to the local DNS (LDNS) server	Response time, in milliseconds, to ping from GSLB remote node to LDNS and back.	0–300000	3000

### **Assigning Weights**

Weights are assigned to each metric on a per-site basis. They are specified as a number between zero and 100. The default is zero for all metrics.

When a DNS request is made to the GSLB master, the master gathers statistics from the other GSLB remote nodes. Each of the metrics' weights are taken into account, and a total score is calculated for the GSLB remote node. If a particular metric has a weight of 0, it is ignored. If a GSLB remote node does not respond within the time set by the GSLB master's timeout, it is given a total score of 0 for that request.



If all GSLB groups using a particular GSLB remote node have all their metric weights set to zero, no requests are sent to that GSLB node.

Once a score has been determined for a particular GSLB remote node, it is combined with the weight set for the GSLB Group member to determine a member score. This final member score determines the IP address returned to the LDNS.

### Tracking Performance with Statistics

There are five categories of GSLB statistics available: GSLB agent, remote node, resolver, group, and member. These statistics are described in Table 3.

**Table 3: GSLB Statistics**

Statistic	Description
<b>GSLB Agent</b>	Collected on every GSLB node and viewable on every DX appliance, whether configured as the GSLB master or a GSLB node. Full and abbreviated displays available.
Metrics requests received	Number of requests received from other GSLB agents.
Metrics replies sent	Number of responses sent to other GSLB agents.
RTT requests received	Number of round-trip time (RTT) requests received from other GSLB agents.
RTT replies sent	Number of RTT responses sent to other GSLB agents.
<b>GSLB RemoteNode</b>	Collected and viewable on the GSLB master. Provides information about the communication between the GSLB master and the GSLB node's agent. Full and abbreviated displays available.
Status	Administrative status of the remotenode—Up or Down.
RTT requests sent	Number of RTT requests sent to this remote node.
RTT replies received	Number of RTT replies received from this remote node.
RTT errors	Number of RTT errors received from this remote node.
Metric requests sent	Number of metrics requests sent to this remote node.
Metric Replies	Number of metrics replies received from this remote node.
Metric Errors	Number of metrics errors received from this remote node.
<b>GSLB Resolver</b>	Collected and viewable on the GSLB master. Provides information about the communication between the client LDNS and target DNS servers. Full and abbreviated displays available.
UDP requests	Number of UDP DNS requests made to the GSLB resolver by an LDNS server.
UDP replies	Number of UDP DNS replies made by the GSLB resolver to an LDNS server, including replies from the target DNS server.
UDP forwards	Number of UDP requests forwarded to the target DNS server.

**Table 3: GSLB Statistics (continued)**

<b>Statistic</b>	<b>Description</b>
UDP replies from DNS server	Number of UDP replies to the GSLB resolver from the target DNS server.
UDP errors	Number of UDP DNS error messages generated by the GSLB resolver.
TCP requests	Number of TCP DNS requests made to the GSLB resolver by an LDNS server.
TCP replies	Number of TCP DNS replies made by the GSLB resolver to an LDNS server, including replies from the target DNS server.
TCP forwards	Number of TCP requests forwarded to the target DNS server.
TCP replies from DNS server	Number of TCP replies to the GSLB resolver from the target DNS server.
TCP errors	Number of TCP DNS error messages generated by the GSLB resolver.
Total requests	Total number of DNS requests made to the GSLB resolver by an LDNS server.
Total replies	Total number of DNS replies made by the GSLB resolver to an LDNS server, including replies from the target DNS server.
Total forwards	Total number of requests forwarded to the target DNS server.
Total replies from DNS server	Total number of replies to the GSLB resolver from the target DNS server.
Total errors	Total number of DNS error messages generated by the GSLB resolver.
Request type A	Number of A record requests made to the GSLB resolver.
Request type NS	Number of NS record requests made to the GSLB resolver.
Request type CNAME	Number of CNAME record requests made to the GSLB resolver.
Request type SOA	Number of SOA record requests made to the GSLB resolver.
Request type PTR	Number of PTR record requests made to the GSLB resolver.
Request type MX	Number of MX record requests made to the GSLB resolver.
Request type Other	Number of other valid DNS requests made to the GSLB resolver.
<b>GSLB Group</b>	Collected and viewable on the GSLB master. Full and abbreviated displays available.
Total requests	Number of A or CNAME requests handled by the GLSB group.
Pending requests	Number of pending requests. These are typically requests that are waiting for an RTT response.
Total replies	Total number of DNS replies generated by this GSLB group.
Normal replies	Number of DNS responses containing group members.
FailIP replies	Number of DNS responses containing only the failed IP address.
Empty replies	Number of empty DNS responses.
Errors	Number of internal errors.

**Table 3: GSLB Statistics (continued)**

Statistic	Description
<b>GSLB Member</b>	Collected and viewable on the GSLB master. Full and abbreviated displays available.
Times served	Total number of times the GLSB member has been used in a DNS response.
Times served first	Number of of times the GSLB member has been provided as the first IP address in the DNS response.

### Deployment Considerations

Deployment of GSLB is quite simple in most cases. Your site does not need to change its DNS server configuration at all. However, since DNS server IP addresses are maintained by external domain registrars, you must make the following minor changes:

1. Use one of the following methods to make the DX appliance acting as the GSLB master the primary DNS address:
  - Set the GSLB resolver listen address on the DX appliance acting as the GSLB master to a new IP address and put in a request to your registrar to update your DNS record to point to the DX appliance.
  - Set the GSLB resolver listen address on the DX appliance acting as the GSLB master to the IP address of the current DNS server, and hide the current DNS server behind a Network Address Translation (NAT) device.
2. Define the remote GSLB nodes. Although the IP addresses of these nodes must be publicly available, the health-checking IP addresses for GSLB DNS responses do not. Because the configuration allows for the health-checking addresses to be defined independently from the actual IP addresses published by DNS, the health-checking IP addresses can be located on a private back-channel.

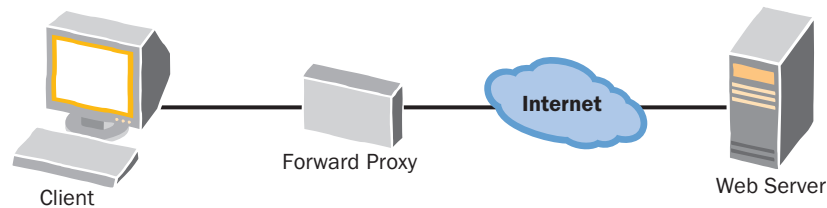
## Forward Proxy Acceleration

A forward proxy server supports Internet access for a number of clients through a single server for security, caching, or filtering as opposed to a reverse proxy server that redirects requests from a client for a Web site to a number of servers. The DX normally operates as a reverse proxy server.

The forward proxy server can be a gateway to the Internet in an enterprise. When this proxy service is contained on a single machine, it can act as an authenticated gateway through firewalls, and prevent direct Internet access to clients. The proxy server may also provide a cache to store frequently used and accessed Web sites, graphics, and other elements. The proxy server can be used to filtering the information that clients can access because all requests for Web pages go through the proxy server. For example, the forward proxy server can block advertisements and pop-ups or entire Web sites.

A forward proxy server sits on the network between the client and the Internet (Figure 34). It may be the same machine that provides the Internet connection and firewall service.

**Figure 34: Forward Proxy Network Setup**



TTW 012

### Sample Scenarios

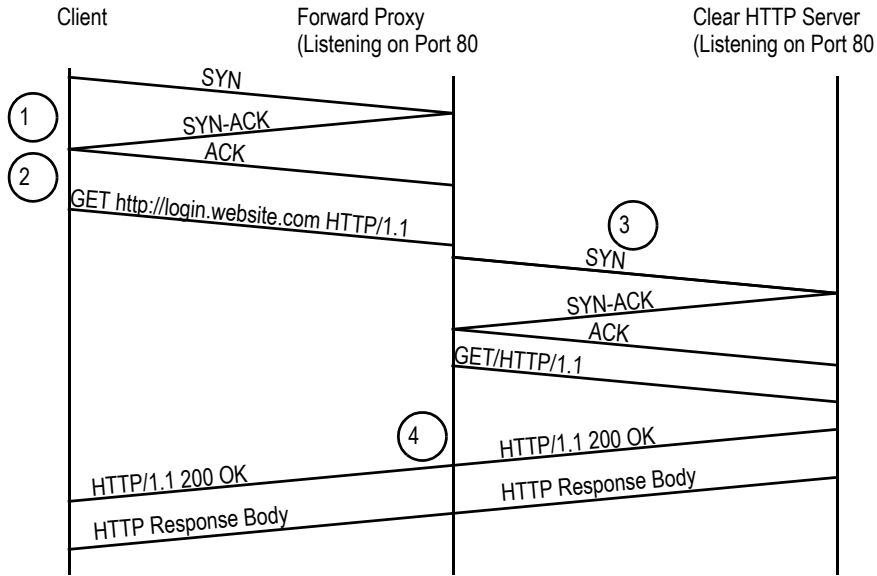
Three common scenarios are presented to provide additional background about the interactions between HTTP browsers and forward proxy servers. The scenarios assume use of the network setup in Figure 34 and that the forward proxy only listens on port 80. (In the Internet Explorer this is set up under Tools > Internet Options > Connections > LAN Settings > Proxy Server).

#### Scenario 1: Clear Request for a Clear Page

This scenario is used by browsers to retrieve clear pages through a forward proxy. Figure 35 shows the requests and responses for each step.

1. The client establishes a TCP connection to port 80 of the forward proxy.
2. The client sends a GET `http://login.website.com` HTTP/1.1 request for a clear (non-SSL) page to the forward proxy. Note that the URL includes the “http://.”
3. The forward proxy uses DNS to resolve `login.website.com` to an IP address, establishes a TCP connection to port 80 of that IP address, and sends GET / HTTP/1.1.
4. The response from the Web server is forwarded back to the client. The forward proxy can manipulate the HTTP headers as needed.

**Figure 35: Clear Pages through a Forward Proxy**



**Scenario 2: CONNECT Request for a Secure Page**

This scenario is used by browsers to retrieve SSL pages through a forward proxy. Figure 36 show the requests and responses for each step.

1. The client establishes a TCP connection to port 80 of the forward proxy.
2. The client sends CONNECT login.website.com:443 HTTP/1.1 to the forward proxy.

For example:

```
CONNECT login.website.com:443 HTTP/1.1
User-Agent: Mozilla/5.0 (Windows; U; Windows NT 5.0; en-US; rv:1.7.5)
Gecko/20041107 Firefox/1.0
Proxy-Connection: keep-alive
Host: login.website.com
```

3. The forward proxy uses DNS to resolve login.website.com to an IP address, and establishes a TCP connection to port 443 of that IP address.
4. The forward proxy sends back a Connection Established response to the client.

For example:

```
HTTP/1.0 200 Connection established
Proxy-agent: Apache/1.3.26 (Unix) mod_ssl/2.8.10 OpenSSL/0.9.6e
```

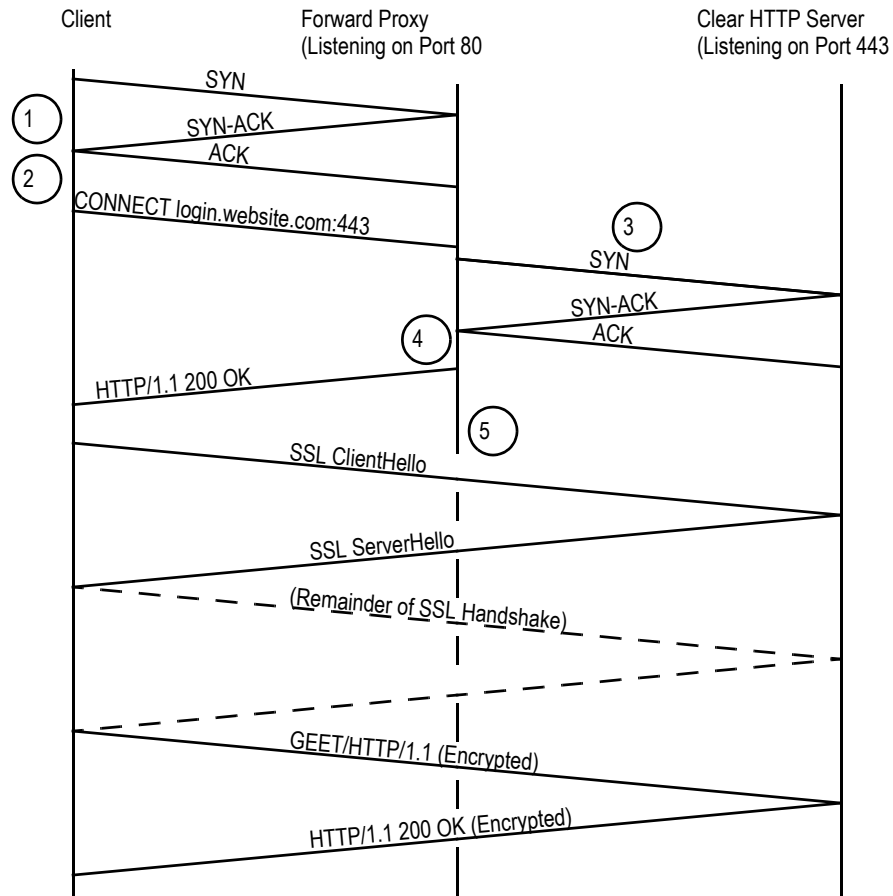
5. At this point, the client establishes an SSL connection on the existing TCP connection (i.e., by exchanging ClientHello and ServerHello messages, and so forth), but the other endpoint of the SSL connection is the Web server at login.website.com, not the forward proxy. The forward proxy simply forwards bytes back and forth between the client and login.website.com.

Essentially, the CONNECT method allows tunneling of other TCP-based protocols (like SSL) over HTTP. The CONNECT method is available in all HTTP versions.



**NOTE:** In this scenario, the forward proxy cannot inspect or modify the application data.

**Figure 36: SSL Pages through a Forward Proxy**



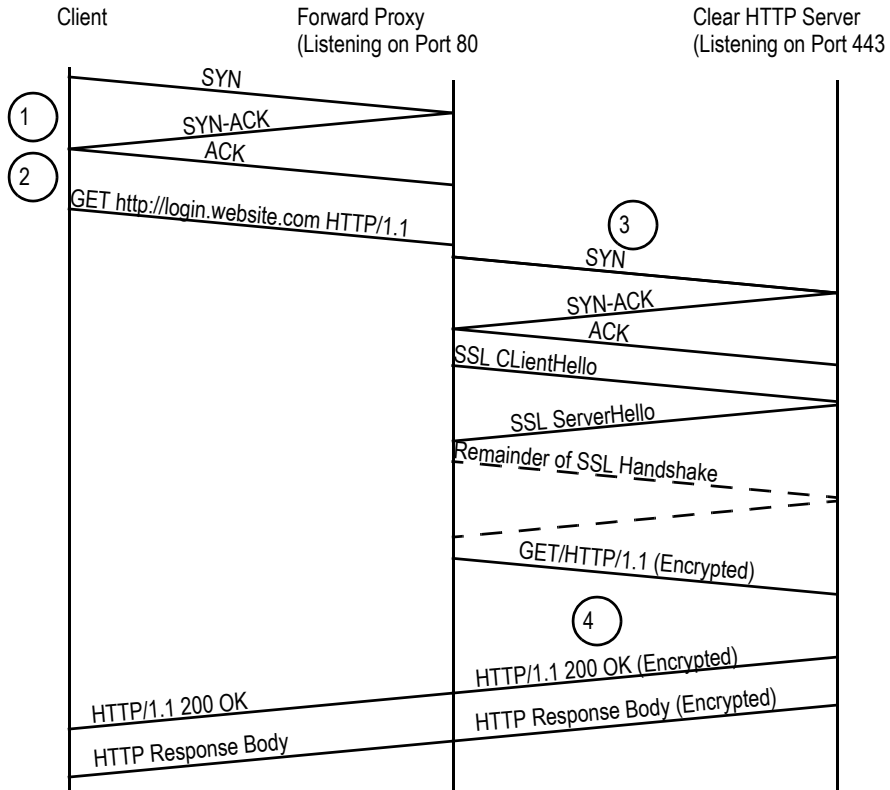
### Scenario 3: Clear Request for Secure Page (without CONNECT)

This scenario illustrates the only way that a forward proxy could inspect and modify the application data for SSL connections. This method is not typically used by browsers security reasons, but it may be used by custom clients. Figure 37 shows the requests and responses for each step.

1. The client establishes a TCP connection to port 80 of the forward proxy.
2. The client sends a GET `https://login.website.com` HTTP/1.1 request for an SSL page to the forward proxy.

3. The forward proxy uses DNS to resolve *login.website.com* to an IP address, establishes a TCP and an SSL connection to port 443 of *login.website.com*, and sends GET / HTTP/1.1.
4. The SSL response from the Web server is decrypted and forwarded back to the client in the clear. The forward proxy may manipulate the HTTP headers.

**Figure 37: Clear Request for a Secure Page (without CONNECT)**



A variation on this scenario is to have an SSL connection between the client and the forward proxy, but a different SSL connection than the one between the forward proxy and the Web server.

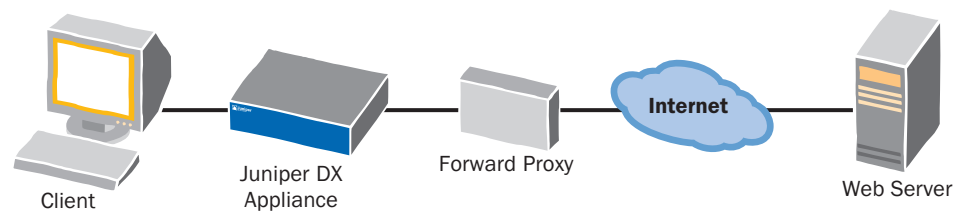
**Forward Proxy Acceleration with the DX**

The DX Application Acceleration Platform uses the Forward Proxy Accelerator feature to accelerate HTTP traffic served by a forward proxy. The DX itself is NOT the forward proxy. The Forward Proxy Accelerator is an optional feature that requires a license. Contact your Juniper Networks Sales Representative to obtain a license.



In this application, the DX sits between the client and the forward proxy. It transforms normal HTTP requests (such as GET, POST, PUT) as usual using compression, OverDrive (AppRules), and so on. The DX also detects HTTP CONNECT requests from clients, and forwards data on those connections between the client and the forward proxy without any transformation. Previously, if the DX was located in front of a forward proxy, it could only support scenarios (1) and (3). With the Forward Proxy Accelerator feature, the DX supports the CONNECT method in scenario (2) as well. Figure 38 shows a diagram of the network setup.

**Figure 38: Forward Proxy Network Setup**



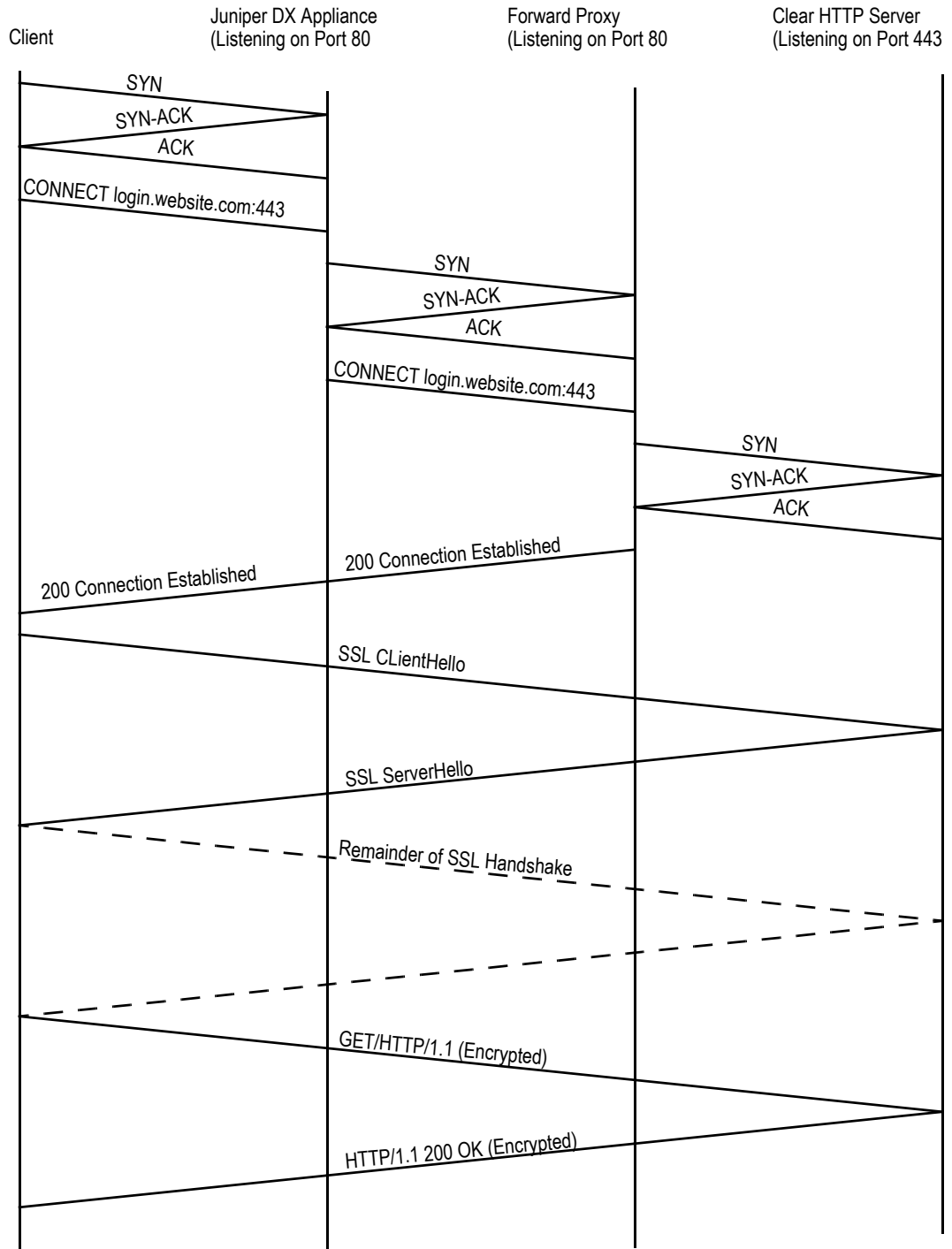
Connection binding is required to ensure that target sessions for CONNECT method requests are not reused. <sup>TTW 013</sup>

Figure 39 shows a diagram of the connection setup for the CONNECT method.



**NOTE:** Because browsers primarily use scenarios (1) and (2), the DX appliance will only be able to accelerate clear traffic and not SSL traffic.

**Figure 39: Forward Proxy with DX Application Acceleration Platform CONNECT Method**



## Chapter 3

# Using the DX Administrative Interfaces

This chapter describes the remote administrative interfaces available for configuring and monitoring the DX Application Acceleration Platform. It includes the following topics:

- “Overview” on page 81
- “Using the Command Line Interface” on page 81
- “The Web User Interface (WebUI)” on page 86
- “SNMP Agent” on page 90

## Overview

---

The DX Application Acceleration Platform provides a variety of administrative interfaces to suit your environment and security needs:

- The DX command line interface (DXSHELL)—The DXSHELL contains a comprehensive set of commands that allow you to view and change every aspect of the DX configuration. For a list of all commands and their options, refer to the *Command Line Reference* manual.
- The browser-based Web user interface (WebUI)—The WebUI provides a graphical interface that allows you to view and change the most frequently used configuration options.
- An SNMP agent—The DX includes a custom Management Information Base (MIB) that allows you to view the DX's configuration and status through SNMP. The SNMP agent also sends generic and enterprise-specific traps.

## Using the Command Line Interface

---

The DXSHELL is used to view and configure the DX appliance. It is described in the following sections:

- “Accessing the Command Line Interface” on page 82
- “Working in DXSHELL” on page 83

## Accessing the Command Line Interface

The DX command line interface, DXSHELL, is accessible through the following methods:

- Remotely via Secure Shell (SSH) login
- Remotely via Telnet login
- Directly via the serial connection on the DX (console port)

All three methods provide identical access to the DXSHELL.

### Using SSH to Access the DX Appliance Command Line

The DX can be accessed through an SSH client. Using SSH ensures that while you are connected to the DX, all information passing between you and the DX is encrypted for security. You must have an SSH client or application installed and functioning on the computer from which you are accessing the DX.

Use these steps to connect using SSH:

1. If you are using a command line SSH client, type the following command:

```
ssh admin@<IP address of DX>
```

If you are using a PC with a terminal emulator application that supports SSH, configure it to connect to the IP address of the DX. When you are prompted for the username, either enter “admin” for the default account, or the name of a user account that you have created.

2. Enter the password for the DX when prompted.

The % prompt indicates that you have reached the Juniper Networks DXSHELL.

### Using Telnet to Access the DX Appliance Command Line

The DX can be accessed through a standard Telnet client. You must have a Telnet client or application installed and functioning on the computer from which you are accessing the Juniper Networks DX.



**NOTE:** The DX appliance's Telnet administration service must be turned-on to connect to the DX appliance using Telnet.

---

Use these steps to connect using Telnet:

1. If you are using a command line Telnet client, type the following command:

```
telnet <IP address of DX>
```

If you are using a PC with a terminal emulator application, configure the emulator to connect to the IP address of the DX.

2. Enter the username and password that you set for the DX when prompted.

The % prompt indicates that you have reached the Juniper Networks DXSHELL.

### Using a Console Port to Access the DX Appliance Command Line

The DX can be accessed through a direct serial connection to the console port on the back of the unit. The console connection must be used for the first-time configuration. After that, it provides out-of-band management capability.

Use these steps to connect using the console port:

1. Connect one end of the supplied null modem cable to the serial (console) port on the rear of the unit.
2. Connect the other end of the cable to the COM1 port of a PC running terminal emulation software or any standard RS-232 terminal. Use 9600 baud, 8 bits, and no parity (refer to “Installing Your DX Appliance” on page 97 for details).
3. Open a terminal session and press ENTER to bring up communication with the DX.
4. Enter the username and password for the DX when prompted. If this is the first time that you have logged in, use the default account with the username “admin” and the password “admin.” The % prompt indicates that you have reached the Juniper Networks DXSHELL.

### Working in DXSHELL

Several features of the DXSHELL simplify its usage. These are discussed in the following sections:

- “Making Changes from the Command Line” on page 83
- “Using Command Abbreviation” on page 85
- “Getting Help” on page 85
- “Logging Out of DXSHELL” on page 86

### Making Changes from the Command Line

Three groups of commands—`show`, `set`, and `clear`—are used to view and change all of the configurable parameters for the DX. A complete list of parameters that can be modified with the `set` and `clear` commands, along with examples, is provided in the *Command Line Reference*.

After using the `set` and `clear` commands to make changes, an asterisk (\*) appears in front of the command line prompt, indicating that configuration settings have been changed, but the changes have not yet been saved. With the exception of a few commands, changes do not take effect and are not saved until you enter the `write` command. If you have not yet entered the `write` command, you can revert to the configuration settings that existed before changes were made by entering the `reload` command.

The following `set` commands control the state of the DX and are the only commands that take effect immediately, without execution of the `write` command:

- `set server [ up | down ]`
- `set admin ssh [ up | down ]`

- set admin telnet [ up | down ]
- set admin webui [ up | down ]
- set admin snmp [ up | down ]
- set admin soap down/up
- set activeN disabled/enabled
- set slb disabled/enabled
- set boot
- set cluster target host [hardpaused|softpaused|unpaused]
- set forwarder target host [hardpaused|softpaused|unpaused]
- set slb group target host [hardpaused|softpaused|unpaused]
- add user
- clear user role
- delete user
- set user class
- set user disabled
- set user enabled
- set user mustchange
- set user password
- set user role
- clear server stats
- clear cluster <name> stats
- clear cluster <name> sticky clientip entry
- clear slb stats
- clear slb group <name> stats
- clear slb group <name> sticky entry
- clear forwarder <name> stats
- clear forwarder <name> sticky clientip entry
- clear activeN stats
- clear cache <name> stats
- clear health script <name |all> stats
- clear log system
- clear log apprule
- clear log audit
- clear log health script
- clear redirector <name> stats
- capture file
- capture license
- capture loginbanner

- some of the `import` commands



If you wish to preserve the configuration changes from these commands (so they remain active on the next boot-up), you must follow these `set` commands with a `write` command.

### Using Command Abbreviation

The Command Abbreviation feature allows you to type abbreviated DXSHELL commands that are then resolved and executed by the DX appliance. The output delivered by the execution of unambiguous commands is the same as its non-abbreviated command equivalent. However, if a command is ambiguous, an error is issued, such as “Ambiguous Keyword”. The DX also suggests possible matches:

```
dx% c1 cluster
Ambiguous keyword: "c1"
Possible matches:
clear
cls
dx%
```

For example, the DXSHELL command used to check health interval for Cluster 1 is:

```
dx% show cluster 1 health interval
```

The abbreviated command equivalent is:

```
dx% sh clu 1 he in
```

### Guidelines

Command abbreviation is subject to the following restrictions:

- Both commands and parameters can be abbreviated. For example, you can abbreviate the `show` command to `sh` because `show` is the only command that begins with `sh`.
- The abbreviation must contain enough letters to differentiate it from the other commands and parameters at that level. For example, `sh c1` is not unique in its parameter so you must type `sh clu` to specify `show cluster`. Similarly, the command `sh clu 1 st` is not a unique command as there are two possible interpretations: `show cluster 1 stats` and `show cluster 1 sticky`. A unique, abbreviated command for `show cluster 1 stats` would be `sh clu 1 sta`.
- The determination of a unique command or parameter is made dynamically. User-defined names (a cluster name, for example) are not considered part of the command syntax check, and a command that can be resolved by the system without considering user-defined names or strings will be executed.

### Getting Help

Type `help` or press the `tab` key to see a list of commands or parameters that can be used at that time.

### Logging Out of DXSHELL

Disconnect from the DXSHELL at any time by entering either the `exit` or `quit` command, as follows:

```
dx% exit
```

or

```
dx% quit
```

## The Web User Interface (WebUI)

---

The WebUI provides access to the most commonly used DX configuration parameters in a familiar and easily-accessible Web interface. Users with the Administrator role have read-write access to all pages on the WebUI. Users with all other access roles have read-only access to the WebUI pages. This includes users with access roles `network_administrator`, `network_operator`, `security_administrator`, or `user` (refer to “Multi-Level Administrative Rights” on page 26).

**NOTE:** You must have Netscape version 6.x or later, Internet Explorer version 5.x or later, or Opera version 6.x or later installed for proper operation of the WebUI.

The WebUI is described in the following sections:

- “Enabling the WebUI Server” on page 86
- “Setting the WebUI Interface to Communicate over SSL” on page 87
- “Accessing the WebUI” on page 87
- “Working with the WebUI” on page 88
- “On-Line Help in the WebUI” on page 89
- “Logging out of the WebUI” on page 90

### Enabling the WebUI Server

If you did not enable the WebUI Server during initial configuration, or if it is not otherwise available, you must access the DXSHELL to enable it.

To enable the WebUI server:

1. Access DXSHELL either through a direct terminal connection or remotely using SSH or Telnet. The default port for the WebUI is 8090.
2. From the DXSHELL, enter the following commands:

```
dx% set admin webui port <number>
dx% set admin webui up
dx% write
```



## Setting the WebUI Interface to Communicate over SSL

If you plan on accessing the WebUI over an unsecured connection, you should enable “Secure Socket Layers” for the WebUI. This is an optional process that should only be used when extra security is needed.

1. Access DXSHELL either through a direct terminal connection or remotely using SSH or Telnet. This may already be in place if you are continuing from the previous section.
2. From the DXSHELL, enter the following commands:

```
dx% set admin webui ssl keyfile demokey
dx% set admin webui ssl keypass
dx% set admin webui ssl certfile democert
dx% set admin webui ssl enabled
dx% write
Writing configuration.
Done.
```

This example uses the dummy key and certificate files named `demokey` and `democert`, respectively. If you are installing the DX in a production environment, make sure you have valid key and certificate files in base-64 encoding. Instructions for importing these files from a variety of environments, as well as converting them to base-64, appear in “Importing Existing Keys and Certificates” on page 197.

When importing key files from different environments, occasionally they will need to be converted using the OpenSSL software. For information on this program, refer to the open SSL Web pages at:

<http://www.openssl.org/>

3. To see the current WebUI SSL setup, type the command:

```
dx% show admin webui
The DX responds with the current setup:

Port: 8090
SSL Status: enabled
SSL Keyfile: demokey
SSL Keypass: none
SSL Certfile: democert
Session Expire Time: 900
Web UI: up
```

## Accessing the WebUI

To log in to the WebUI:

1. Open a Web browser (you may need to be inside your company's firewall to access the Web interface).
2. Type the DX host name or IP address along with the port on which the WebUI is listening (the default port is 8090) in your browser's address bar. The URL may look something like this:

http://192.168.100.100:8090 or http://dx.yourdomain.com:8090



**NOTE:** It is possible to configure a WebUI administrator to listen on an IP address (10.0.20.0, for example) and use port 8090. At the same time, a cluster of target hosts may be configured to use the same IP and port (10.0.20.0:8090). When a configuration change is made that requires a restart of the multiplexing engine, a WebUI administrator page could be displayed. To prevent this from occurring, you should not use the administrator port as a cluster port.

- When prompted, enter your username and password.

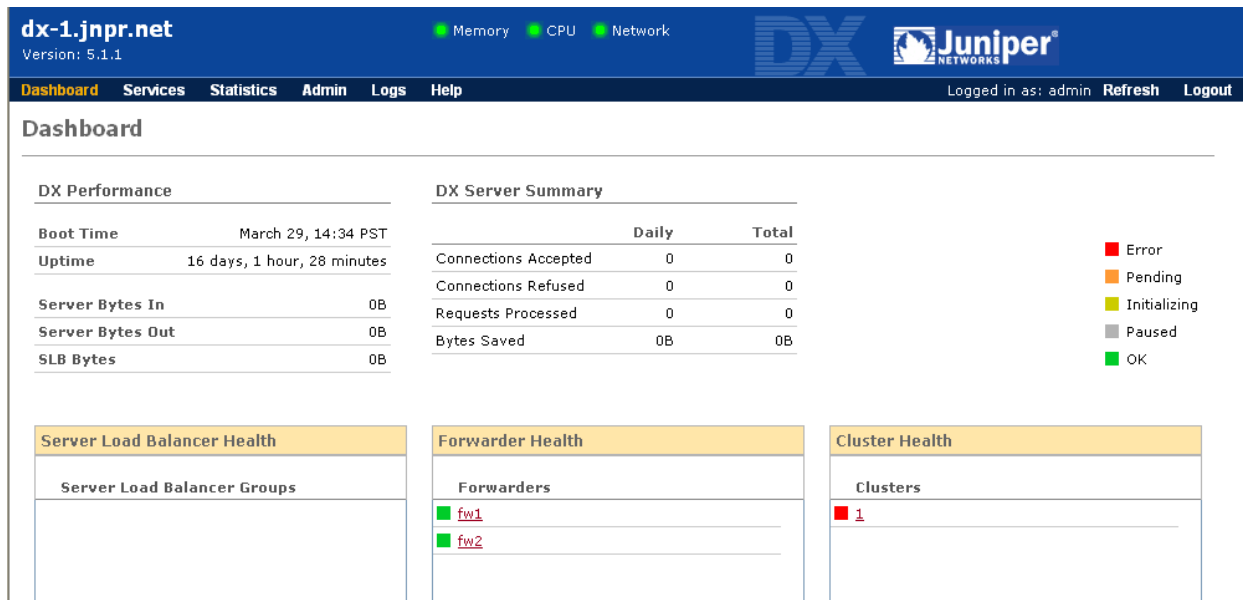
Use the default username `admin` and password `admin` or one of the previously-defined user accounts. The password is the one that you set during the first-time configuration.

You are now logged in. Use the WebUI dashboard that is displayed to configure your DX appliance, as described in “Working with the WebUI” on page 88.

### Working with the WebUI

After logging in, the WebUI dashboard is displayed (Figure 40) showing the current performance of the DX appliance and the health status of any configured SLBs, Forwarders, and Clusters.

**Figure 40: The WebUI Dashboard**



From the navigation bar at the top of the dashboard, you can access Services, Statistics, Admin, Log, and Help pages allowing you to configure and monitor your DX appliance. The Services, Statistics, Admin and Log pages contain a menu on the left with links to pages where you can configure or monitor your DX appliance. The Help page takes you to online information about the DX Appliance. Table 4 lists the pages, menus, and a brief description of tasks that can be performed on those pages.

**Table 4: Description of WebUI Pages and Menus**

Navigation Bar Link	Menu Links	Description
Dashboard	None	Current status information only.
Services	Clusters, Forwarders, Redirectors, ActiveN, Server Load Balancer	Contains pages to configure clusters, forwarders, redirectors, ActiveN, and SLB services.
Statistics	DX, Clusters, Forwarders, Redirectors	Contains graphical representation of statistics for the DX server, clusters, forwarders, and redirectors.
Admin	DX server, Admin services, Audit trail, Date & time, Email, Failover, Keys & Certificates, Logging, Network, Remote Host Health, SNMP TCPDump, Transport, TSDump, Upgrade, and Users	Contains pages to configure and monitor all aspects of the DX appliance.
Log	Audit, AppRules Log, Health, System	Displays the log file associated with the selected menu.
Help	None	Takes you to online help topics.

### **Making Changes with the WebUI**

The WebUI lets you view and change settings with a familiar forms-based Web interface.

To make changes:

1. Select the desired option or enter the desired value.
2. Click the SAVE button at the bottom of the page. Your changes will be saved and applied immediately.

If you make a mistake and do not want to save your changes, you can click your browser's refresh button to get a fresh copy of the page. You can also select one of the other settings pages from the left-hand navigation menu and your changes will not be saved.

### **On-Line Help in the WebUI**

On-line help is available by clicking on the terms that appear next to each field. When you click on a term, a pop-up window opens and the term and its definition appear at the very top of the window.

## Logging out of the WebUI

When you have finished your WebUI session, you should log out of your administration session using the Logout button. Then close the browser window and quit your Web browser to prevent anyone from re-opening your WebUI session. This prevents someone from using that browser to access the WebUI.

If you forget to log out, the session automatically times out after a fixed period. You then must log in again before you are able to access the WebUI.

## SNMP Agent

---

The SNMP agent supports SNMP Version 2c for SNMP get and getnext, and version 1 and 2c for SNMP traps. The SNMP agent does not support the SNMP set operation. Security is provided through SNMP community strings. The default community strings are “public” for the SNMP getnext operation. The community strings can be modified through either DXSHELL or the WebUI. SNMP traps do not have a default setting; you must configure a trap.

Juniper Networks is registered as Enterprise 6213. Detailed SNMP Management Information Base (MIB) and trap definitions for the SNMP agent can be found in the following Juniper Networks Enterprise MIB documents:

- DX-MIB: Juniper enterprise top level MIB definitions
- DX-CONFIG-MIB: Juniper enterprise configuration MIB definitions
- DX-STATS-MIB: Juniper enterprise statistics MIB definitions
- DX-TRAP-MIB: Juniper enterprise trap definitions (SNMP v1.0)
- DX-TRAPv2-MIB: Juniper enterprise trap definitions (SNMP v2.0)

Users may specify up to two trap hosts for receiving SNMP traps. The agent will send the SNMP trap to the specified hosts when appropriate. The SNMP agent can send version 1 and version 2 trap formats. Traps are not sent when no host is specified.

The SNMP agent supports the standard MIB, RFC 1213-MIB II, and the following generic traps:

- ColdStart
- WarmStart
- LinkDown
- LinkUp
- Authentication Failure

The SNMP agent also supports the Enterprise SMNP traps shown in Table 1.

**Table 1: Enterprise SNMP Traps Supported**

Trap Name	Description
failoverStateActive	Indicates that the Juniper Accelerator is assuming the active role.
connectionThresholdTrap	Indicates that the Juniper Accelerator has reached the threshold for the maximum number of connections on the client side.
TargetServerStateUp	Indicates that the target server is up.
TargetServerStateDown	Indicates that the target server is down.
vipStateDown	Indicates that the VIP is down.
vipStateUp	Indicates that the VIP is up.



## Part 2

# Installation Information and Procedures

This part of the *Installation and Administration Guide for DXOS* provides procedures for installing and configuring your DX appliance on start-up.

These topics can be found in the following chapters:

- Chapter 4, “Installing Your DX Appliance” on page 95
- Chapter 5, “Performing Initial Configuration of the DX Appliance” on page 99





## Chapter 4

# Installing Your DX Appliance

This chapter describes the steps needed to install your DX Application Acceleration Platforms. An overview of the tasks required, prerequisites, and procedures are provided in the following topics:

- “Installation Overview” on page 95
- “Network Configuration Information Needed” on page 96
- “Connect a terminal to the console port on the DX appliance.” on page 97
- “Connect the DX appliance’s primary Ethernet interface to your network using a standard Ethernet cable.” on page 97
- “Power-up the DX Appliance based on the model you have:” on page 97

### Installation Overview

---

Installation requires adding no hardware or software to your Web servers. It also requires no modification or preparation of the content to be accelerated. Of course, the DX is completely transparent to end users, requiring no special plug-in or software download.

This is a high-level overview of the steps required to install the DX:

- Connect the power and network cables.
- Connect the DX console port to a terminal or a computer with a terminal emulation program, then provide the DX with basic network and target host information.
- Integrate the DX into your Web traffic flow.

## Network Configuration Information Needed

Table 2 shows the information required for first-time configuration of the DX appliance.

**Table 2: Network Configuration Information Required for First-Time Configuration**

Required Information	Example
<p><b>IP address</b></p> <p>The IP address for the Remote Administration Interface port for this DX appliance. This can be any arbitrary valid IP address on your subnet.</p>	192.168.4.76
<p><b>Netmask</b></p> <p>The Netmask (subnet mask) of this DX appliance.</p>	255.255.0.0
<p><b>Fully-qualified host name</b></p> <p>The public name of this DX appliance that will be set in DNS records.</p>	dx.juniper.net
<p><b>Default route</b></p> <p>The Default route (sometimes called the gateway) for this DX appliance.</p>	192.168.0.1
<p><b>DNS domain</b></p> <p>The DNS Domain (sometimes known as the DNS suffix) where this DX appliance is installed.</p>	juniper.net
<p><b>Primary nameserver</b></p> <p>The Primary nameserver for this DX appliance.</p>	192.168.0.5
<p><b>Username</b></p> <p>The username for this DX appliance. The default username is admin.</p>	admin
<p><b>Password</b></p> <p>The password for this DX appliance. The default password is admin.</p>	admin

## Installing Your DX Appliance

---

Once you have collected the information needed and the prerequisites are in place, follow these steps to install your DX appliance:

1. Connect a terminal to the console port on the DX appliance.



**NOTE:** Because it is sometimes difficult to reach the DX appliance console port once it is mounted, consider completing the first-time configuration before mounting your DX appliance into an equipment rack or server cabinet.

---

- a. Connect the supplied null-modem cable to the serial console port on the rear of the unit, running on a PC.
  - b. Connect the other end of the null-modem cable to the COM 1 port of a PC running any standard (RS-232) terminal or terminal emulator software (such as Windows HyperTerminal or SecureCRT).
2. Connect the DX appliance's primary Ethernet interface to your network using a standard Ethernet cable.



**NOTE:** For 1U units with copper media-based Fast Ethernet (10/100/1000BaseT) ports, the DX appliance must be connected to a 10/100/1000BaseT full-duplex network port. The media settings on your switch for the port where the DX appliance is connected must match those for the DX appliance exactly.

For 2U units with fiber media-based Gigabit Ethernet ports, the DX appliance must be connected to a Gigabit switch with the media settings configured to autoselect.

---

3. (Optional) If you intend to install additional DX appliances for redundancy, connect the DX appliance's second Ethernet interface to your network.
4. Power-up the DX Appliance based on the model you have:

*For 1U DX Appliance Models:*

- a. Connect the supplied power cord to the power supply on the back of the DX appliance.
- b. Flip the power switch to the "on" position. The LED on the front of the DX appliance will glow when the DX appliance has power, and the LED on the power supply will glow green.

*For 2U DX Appliance Models with Dual Power Supply:*

Connect the supplied power cord to the power supply on the back of the DX appliance. The DX appliance's dual power supply has no power switch. Connecting a hot power cord to the DX appliance will turn it on and begin the boot process.

The LED on the front of the DX appliance will glow brightly when the DX appliance has power, and the LED on the power supply will glow green.



**NOTE:** The power supply will emit a long startup beep if there is no power to the second power supply. Pressing the red buzzer reset button to the left of the plug will terminate the beep. This is normal.

---



**NOTE:** It may take the DX appliance up to two minutes to boot; allow several minutes before proceeding with configuration.

---

## Chapter 5

# Performing Initial Configuration of the DX Appliance

This chapter provides the steps necessary to configure your DX Application Acceleration Platform once you have initially installed the system. It also discusses how to change the default administrator account password once you are finished with the configuration.

It includes the following topics:

- Connecting to the DX Appliance with a Terminal or Terminal Emulator on page 99
- Logging-In for the First Time on page 102
- Read and Agree to the License Agreement on page 103
- Answer the Configuration Questions on page 103
- Changing the Default Administrator Account Password on page 105

## Connecting to the DX Appliance with a Terminal or Terminal Emulator

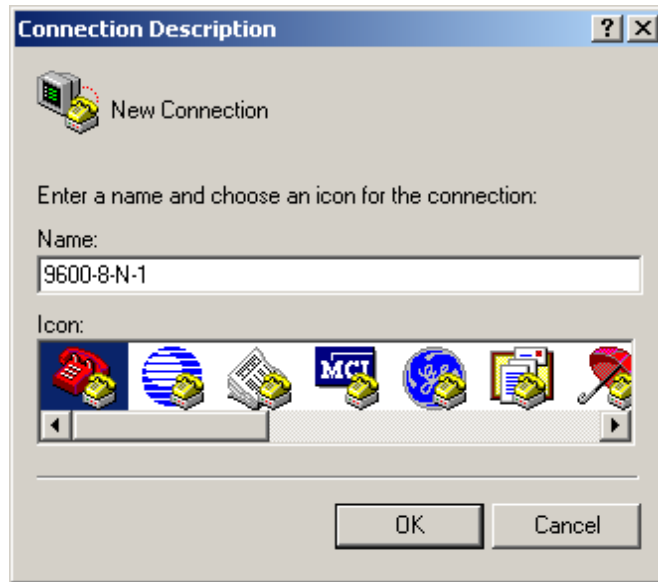
---

If you are using a terminal emulator, be sure that the emulator is configured with the settings listed as:

- Bits per second: 9600
- Data bits: 8
- Parity: none
- Stop bits: 1
- Flow control: none
- Smooth-Scroll: disabled

You must create a connection to use Windows Hyper Terminal. Your first configuration screen should look like the one shown in Figure 41.

**Figure 41: Hyper Terminal Connection Description Dialog Box**



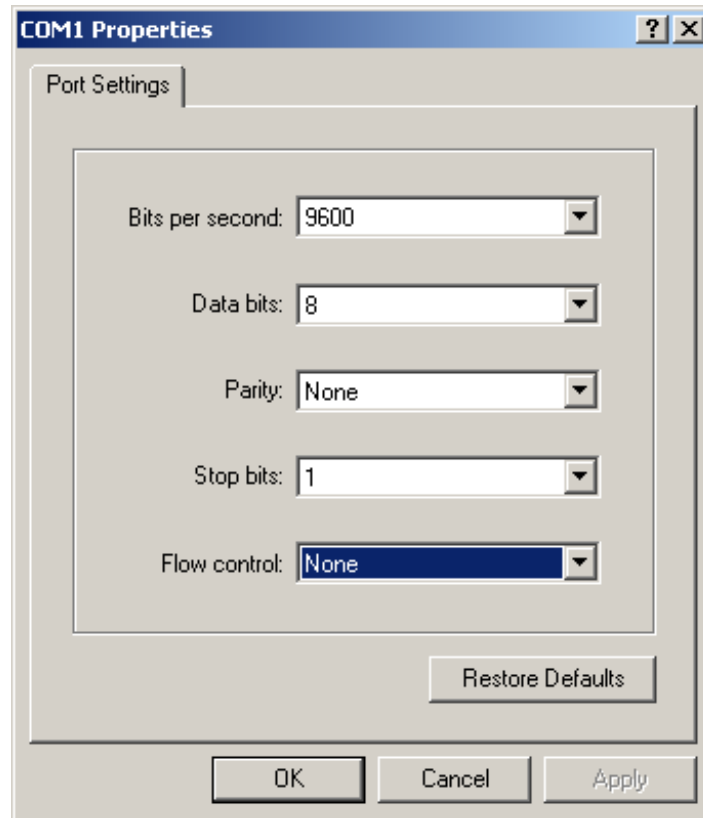
1. Enter a name that will be easy to identify. Hyper Terminal will then ask you which serial port you will be using as shown in Figure 42.

**Figure 42: Hyper Terminal Connection Dialog Box**



- The last step in creating a connection is to configure the communication port properties as shown in Figure 43. Configure the communication parameters as shown.

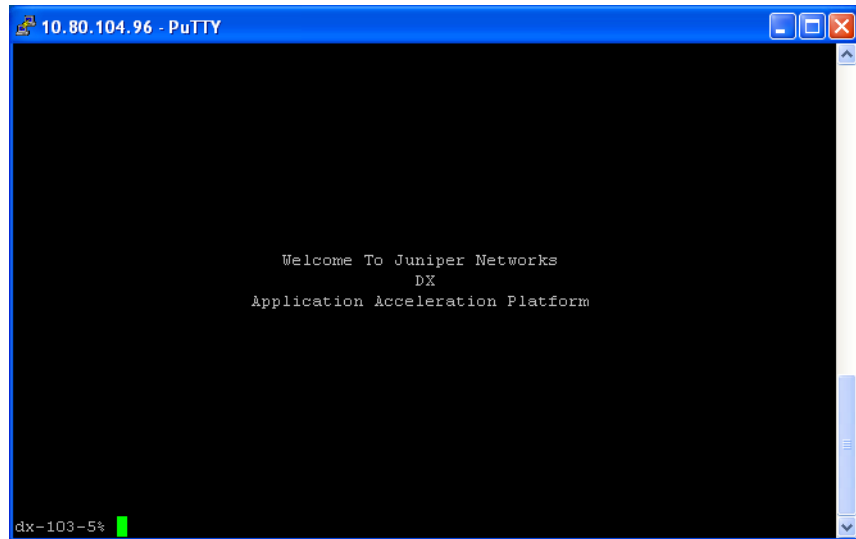
**Figure 43: Hyper Terminal Port Configuration Dialog Box**



**NOTE:** If you are using Windows Hyper Terminal, after clicking the OK button to confirm your settings in the dialog box as shown in Figure 43, you may still need to click the CALL button or select CALL from the CALL pull-down menu to establish the connection.

3. Open the terminal connection to the DX appliance and press ENTER to log-in. You will see a screen similar to the one shown in Figure 44.

**Figure 44: The DX Appliance First-Time Configuration Screen**



If you do not see the screen as shown in Figure 44 and are unable to open a connection to the DX appliance:

1. Be sure that you have given the DX appliance enough time to boot up.
2. If you are using Windows Hyper Terminal, be sure to use CALL to establish a connection after entering the terminal settings. Even if it says CONNECTED in the lower left-hand corner of the Hyper Terminal window, you may not be connected until you use CALL.
3. Try pressing ENTER again to log-in.
4. Double-check that the null modem cable is connected to the COM 1 port of the PC.
5. Double-check that your terminal emulator is configured as previously described.

## Logging-In for the First Time

---

If you have not previously set the username and password for the DX appliance, they will be set to their default values:

- Username: admin
- Password: admin

Log into DX appliance using the appropriate username and password. Continue onto the next step once you have logged-in.



## Read and Agree to the License Agreement

---

Before you can continue with first-time configuration, you must agree to the License Agreement that appears when you first boot the DX appliance. Use the space bar to display each page of the License Agreement until you reach the end. When prompted, type `yes` and press the ENTER key.

## Answer the Configuration Questions

---

The Juniper Networks First-Time Configuration program utility will ask you to provide values for twelve (12) basic configuration parameters required (refer to Table 2 on page 96) to get the DX appliance up and running in your network.

Table 3 on page 104 shows the questions that the DX appliance will ask you along with an explanation of each item. Items shown in brackets (e.g., [172.17.0.2]) are the factory defaults provided to serve as examples for your input. You must provide valid settings for the DX appliance to function in your network. Omit the brackets ([ ]) when typing your input.



**NOTE:** If you make a mistake as you go through the first-time configuration, press CTRL-C and then press ENTER to quit. Then, to re-enter the first-time configuration program, type the command `config` at the DXSHELL prompt and press ENTER.

---

**Table 3: Questions from the DX appliance First-Time Configuration Utility**

First-time Configuration Questions
<p>IP Address [192.168.0.2]: Set the IP address of this DX appliance.</p> <p>Netmask [255.255.255.0]: Set the Netmask (subnet mask) of this DX appliance.</p> <p>Fully-qualified host name []: dx.juniper.net Set the public name of this DX appliance that will be set in DNS records.</p> <p>Default route [192.168.0.1]: Set the Default route (gateway) for this DX appliance.</p> <p>DNS Domain []: juniper.net Set the DNS Domain (domain suffix) where this DX appliance is installed.</p> <p>Primary Nameserver [192.168.0.3]: Set the Primary Nameserver for this DX appliance.</p> <p>Do you want to run the Web Administration Server? [N]: Typing Y will allow you to monitor and configure the DX appliance through a Web browser by entering the address of the DX appliance and the default Web Admin Port 8090 in your browser (e.g., http://192.168.0.168:8090).<sup>1</sup> Access to the Web Administration Manager is password protected and can be turned off at any time.</p> <p>Do you want to allow administration access via ssh? [Y]: Type Y for Yes or N for No. Typing Y will allow you to monitor and configure the DX appliance through a secure Secure Shell (SSH) terminal session. This can be turned off at any time.</p> <p>Do you want to allow administration access via telnet? [N]: Type Y for Yes or N for No. Typing Y will allow you to monitor and configure the DX appliance remotely via telnet. This can be turned off at any time.</p>

1. It is possible to configure the WebUI administrator to listen on an IP (10.0.20.0, for example) and use port 8090. At the same time, a cluster of target hosts may be configured to use the same IP and port (10.0.20.0:8090). When a configuration change is made that requires a restart of the multiplexing engine, a WebUI administrator page could be displayed. To prevent this from occurring, you should not use the administrator port as a cluster port.

After answering all the first time configuration questions as shown in Table 3, you are finished configuring the DX appliance. You should see the following message:

```

Configuration complete.
Writing configuration.
Done.
dx%

```

You are at the DXSHELL command line. The DXSHELL prompt will display the hostname that you assigned to the DX appliance using the First-Time Configuration Utility, followed by the “%” sign (dx% in our example) the next time that you log in.

More information on configuring particular aspects of the DX appliance is presented in the chapters that follow.

## Changing the Default Administrator Account Password

---

For security reasons, as soon as you have configured your DX appliance, you should immediately change the password for the default administrator “admin”. Instructions for doing this are shown in sections, “Managing Users” on page 113. If, for any reason you cannot log onto any of the administrator accounts, you can reset the “admin” administrator password to its default value using the procedure described in “Resetting the admin User Password” on page 120.



## Part 3

# Configuration Information and Procedures

This part of the *Installation and Administration Guide for DXOS* provides some high-level task flows that show how to configure the DX appliance for a particular application or how to use a particular DX platform feature. This part also provides detailed procedures for configuring the various features of your DX appliance.

These topics can be found in the following chapters:

- Chapter 6, “DX Appliance Configuration Flows” on page 109
- Chapter 7, “Administering Your DX Platform” on page 113
- Chapter 8, “Integrating the DX Appliance into Your Network” on page 151
- Chapter 9, “Configuring Server Load Balancing” on page 179
- Chapter 10, “Setting Up the DX Appliance for SSL Traffic” on page 189
- Chapter 11, “Configuring Health Checking” on page 221
- Chapter 12, “Configuring ActiveN” on page 239
- Chapter 13, “Setting up the DX Appliance for “Sticky” Traffic” on page 251
- Chapter 14, “Configuring HTTP(S) Authentication” on page 253
- Chapter 15, “Configuring HTTP(S) Logging” on page 269
- Chapter 16, “Configuring the Forward Proxy Accelerator” on page 285
- Chapter 17, “Configuring the 3G Cache” on page 289
- Chapter 18, “Configuring OverDrive Application Rules” on page 297
- Chapter 19, “Configuring Global Server Load Balancing” on page 335
- Chapter 20, “Configuring Failover” on page 357
- Chapter 21, “Tuning the DX Appliance for Enterprise Applications” on page 373



## Chapter 6

# DX Appliance Configuration Flows

The chapter contains workflows for configuring the DX Appliance for Web and non-Web server application acceleration applications. Links to corresponding configuration procedures are included.

The following flows are included:

- “Setting up the DX Appliance for Server Load Balancing” on page 109
- “Securing Non-Web Applications through SSL” on page 110
- “Migrating Web Applications to Secure Web” on page 111

Assumption for all of the following configuration flows:

- You have installed and configured the network settings on your DX appliance. If not, see Chapter 4, “Installing Your DX Appliance” on page 95 and Chapter 5, “Performing Initial Configuration of the DX Appliance” on page 99.

## Setting up the DX Appliance for Server Load Balancing

Figure 4 shows the steps required to configure and activate the SLB service on the DX appliance. Optional steps are provided for reference. Perform the steps in order, using the links to detailed instructions for assistance.

**Table 4: SLB Configuration Tasks**

Step	Task	Comments
1	Add an SLB Group.	
2	Add the IP address and port number on which to listen for traffic.	For one-arm deployments, these must be on the same subnet as any NIC on the DX.
3	Add one or more target hosts to the group.	
4	(Optional) Configure Network Address Translation (NAT).	
5	(Optional) Configure a load balancing policy (default is round-robin)	
6	(Optional) Configure Quality of Service parameters.	
7	(Optional) Configure health checking parameters.	

**Table 4: SLB Configuration Tasks** Table continued on next page

Step	Task	Comments
8	(Optional) Configure session timers.	
7	(Optional) Configure failover.	
8	Activate the SLB service.	

For more information about server load balancing, see the following:

- Chapter 2, “Application and DX Product Concepts” on page 25
- Chapter 9, “Configuring Server Load Balancing” on page 179

## Securing Non-Web Applications through SSL

Table 5 shows the steps required to configure the DX appliance for secure communications using SSL. Configuring SSL also enables the DX appliance to offload SSL processing from existing equipment. Optional steps are provided for reference. Perform the steps in order, using the links to detailed instructions for assistance.

**Table 5: SSL Configuration Tasks**

Step	Task	Comments
1	Add a Forwarder.	
2	Add the IP address and port number on which to listen for traffic.	For one-arm deployments, these must be on the same subnet as any NIC on the DX.
3	Add one or more target hosts to the Forwarder.	Juniper recommends that the listen and target ports be the same.
5	Enable the listen side of the SSL connection.	
6	Create or import a certificate and key.	
7	Enable the target side of the SSL.	
8	Create or import a certificate and key.	You may use the democert and demokey files to get started.
9	(Optional) Configure client persistence (or “stickiness”)	
10	(Optional) Enable HTTP to HTTPS conversion.	
11	(Optional) Configure client authentication.	
12	Save and activate the Forwarder service and SSL connection.	

For more information about forwarders and SSL, see the following:

- Chapter 2, “Application and DX Product Concepts” on page 25
- Chapter 10, “Setting Up the DX Appliance for SSL Traffic” on page 189



## Migrating Web Applications to Secure Web

Table 6 shows the steps required to configure the DX appliance as a Web application front end (AFE) using SSL for secure communications. You are not required to configure SSL when you configure the DX as an AFE, but it provides additional security for your Web applications without impacting their performance.

Optional steps are provided for reference. Perform the steps in order, using the links to detailed instructions for assistance.

**Table 6: Application Front End and Secure Web Configuration Tasks**

Step	Task	Comments
1	Add a Cluster.	
2	Add the IP address and port number on which to listen for traffic.	For one-arm deployments, these must be on the same subnet as any NIC on the DX.
3	Add one or more target hosts to the Forwarder.	Juniper recommend that the listen and target ports be the same.
5	Enable the listen side of the SSL connection.	
6	Create or import a certificate and key.	
7	Enable the target side of the SSL.	
8	Create or import a certificate and key.	You may use the democert and demokey files to get started.
9	(Optional) Configure client persistence (or “stickiness”)	
10	(Optional) Enable HTTP to HTTPS conversion.	
11	(Optional) Configure client authentication.	
12	Save and activate the Cluster service and SSL connection.	

For more information about the Cluster service and SSL, see the following:

- Chapter 2, “Application and DX Product Concepts” on page 25



## Chapter 7

# Administering Your DX Platform

This chapter describes common administration tasks performed on the DX Application Acceleration Platform. It contains the following topics:

- “Managing Users” on page 113
- “Obtaining a License Key” on page 124
- “Using the Administrator Audit Trail” on page 128
- “Configuring System Event Logging and Notification” on page 129
- “Managing Your DX Appliance Configuration” on page 131
- “Configuring the Login Banner” on page 144
- “Upgrading the DX Application Acceleration Platform Software” on page 146

## Managing Users

---

Administrators are the only users that can add new users and change users' attributes, as described in the following sections:

- “Adding a User” on page 114
- “Changing the User's Password” on page 115
- “Clearing a User's Role” on page 116
- “Assigning Local or Remote Access Rights” on page 116
- “Deleting a User” on page 117
- “Viewing User Information” on page 117
- “Making Global Changes to User Accounts” on page 117
- “Exporting and Importing User Accounts” on page 119
- “Resetting the admin User Password” on page 120
- “Administrator Remote Authentication” on page 121

## Adding a User

You can add users with the WebUI or the CLI.

Follow these steps for adding a new user using the CLI:

1. To add a new user, type:

```
dx% add user
```

or

```
dx% add user <username>
```

2. Enter the new username: `<username>` (the DX will prompt you for the username if it is not provided). The system response will be similar to this:

```
dx% add user fred
```

```
User fred has been added. Please perform the following
to complete the addition of this user:
```

- set a password
- enable the user
- assign a role (optional)

```
dx%
```

Before one or more roles are assigned to a new user, a new user will have very limited rights and can only access the following commands:

```
cls          show hostname
exit         show loginbanner
help        show redirector
history     show server
ping        show support
set password show ua
show cluster show version
show commands who
show forwarder whoami
```

3. Specify the password for the new user by typing:

```
dx% set user <username> password
```

```
New password:
```

```
Enter the new password again:
```

```
Password changed for user <username>.
```

For example:

```
dx% set user fred password
```

```
New password:
```

```
Enter the new password again:
```

```
Password changed for user fred.
```

4. Enable the new user by typing:

```
dx% set user <username> enabled
```

For example:

```
dx% set user fred enabled
dx% User fred is now enabled.
```



A user cannot be enabled unless a password has been assigned. If you try to enable a user without assigning a password, you will receive the error message “Cannot enable user <username> because that user has no password.”

5. Assign one or more roles to the user as shown:

```
dx% set user <username> role <role1 role2 ...>
```

The role can be one of the following: administrator, security\_administrator, security\_operator, network\_administrator, network\_operator, or user.

For example:

```
dx% set user fred role network_administrator
Role network_administrator has been assigned to user fred.
```

```
dx% set user fred role network_operator security_operator
Role network_operator has been assigned to user fred.
Role security_operator has been assigned to user fred.
```

In the second example, user “fred” has been assigned access rights associated with both the network\_operator and security\_operator roles. This is useful in an organization where an operator has administration responsibilities for both the network and security services.

6. Verify the user has been added correctly by viewing the user’s information.

```
dx% show user <username>
```

For example:

```
dx% show user fred
User      Class      Status      Roles
----      -
fred      local      Enabled     network_administrator, network_operator,
security_operator
```

## Changing the User’s Password

If a user forgets his or her password, you can assign a new password.

To change a user’s password, type:

```
dx% set user <username> password
```

For example:

```
dx% set user fred password
New password: Enter a password
Retype new password: Retenter same password
Password changed for user fred.
dx%
```

No characters are echoed on password input, and the user's name is displayed on the final confirmation.

### **Clearing a User's Role**

You can remove an access role for a user when their responsibilities change.

To clear one or more roles for a given user, type:

```
dx% clear user <username> role <role1 role2 ...>
```

The role can be one of the following: administrator, security\_administrator, security\_operator, network\_administrator, network\_operator, or user.

For example:

```
dx% clear user fred role security_administrator
Role security_administrator has been removed from user fred.
```

### **Assigning Local or Remote Access Rights**

You can specify whether a user has local access to your DX appliance or if that user has remote access. This is distinct from a user's role, which provides permission to access particular features and services of the DX appliance. Local access is assigned by default when you add a user.

To assign or change local or remote access rights for a user, type:

```
dx% set user <username> [local|remote]
```

For example:

```
dx% set user fred remote
dx% show user fred
User      Class      Status      Roles
----      -
fred      remote     Enabled     network_administrator
```



When you assign remote access to a user, their existing role or roles are maintained. If no role was assigned, it is assigned the user role.

## Deleting a User

You can remove a user from the system at any time.

To delete a user, type:

```
dx% delete user <username>
```

For example:

```
dx% delete user fred
Are you sure you want to delete user fred (y/n)? [n] y
User fred deleted.
```

## Viewing User Information

You can display information for a particular user or for all users in the system.

To display information for a particular user, type:

```
dx% show user <username>
```

For example:

```
dx% show user fred
User      Class      Status      Roles
----      -
fred      local      Enabled     network_administrator, network_operator,
security_operator
```

To display information for all users, type:

```
dx% show user
```

For example:

```
dx% show user
User      Class      Status      Roles
----      -
admin     local      Enabled     administrator
fred      local      Enabled     network_administrator, network_operator,
security_operator
tom       local      Enabled     administrator
dick     local      Disabled    network_operator
harry    local      Enabled     (none)
```

## Making Global Changes to User Accounts

When desired, you can make changes to parameters for all user accounts at once, rather than one user at a time. Specifically, you can enable or disable all users, assign or remove roles to or from all users, or delete all users. When performing any of these tasks, all user accounts are modified with the exception of:

- The default account—the user with username “admin”.
- The user with the administrator role that is making the changes.

## Enabling and Disabling All Users

To enable or disable all users, type:

```
dx% set user all [enabled | disabled]
```

For example:

```
dx% set user all enabled
Are you sure you want to set all users to 'enabled' (y/n)? [n] y
User fred is now enabled.
User tom is now enabled.
User dick is now enabled.
User harry is now enabled.
```

## Assigning Roles to All Users

You can assign one or more roles to the existing roles for all users.

To assign roles to all users, type:

```
dx% set user all role <role1 role2 ...>
```

The role can be one of the following: administrator, security\_ administrator, security\_operator, network\_ administrator, network\_operator, or user.

For example:

```
dx% set user all role security_administrator
Are you sure you want to change all users' roles (y/n)? [n] y
Role security_administrator has been assigned to user fred.
Role security_administrator has been assigned to user tom.
Role security_administrator has been assigned to user dick.
Role security_administrator has been assigned to user harry.
4 users changed.
```

```
dx% set user all role network_operator network_administrator
Are you sure you want to change all users' roles (y/n)? [n] y
Role network_operator has been added to user fred.
Role network_administrator has been added to user fred.
Role network_operator has been added to user tom.
Role network_administrator has been added to user tom.
Role network_operator has been added to user dick.
Role network_administrator has been added to user dick.
Role network_operator has been added to user harry.
Role network_administrator has been added to user harry.
4 users changed.
```

## Clearing Roles from All Users

You can clear one or more roles from the existing roles for all users.

To clear roles assigned to all user, type:

```
dx% clear user all role <role1 role2 ...>
```

The role can be one of the following: administrator, security\_ administrator, security\_operator, network\_ administrator, network\_operator, or user.



For example:

```
dx% clear user all role security_administrator
Are you sure you want to change all users' roles (y/n)? [n] y
Role security_administrator has been removed from user fred.
Role security_administrator has been removed from user tom.
Role security_administrator has been removed from user dick.
Role security_administrator has been removed from user harry.
4 users changed.
```

```
dx% clear user all role network_operator network_administrator
Are you sure you want to change all users' roles (y/n)? [n] y
Role network_operator has been removed from user fred.
Role network_operator has been removed from user tom.
Role network_administrator has been removed from user tom.
Role network_operator has been removed from user dick.
Role network_administrator has been removed from user dick.
Role network_operator has been removed from user harry.
Role network_administrator has been removed from user harry.
4 users changed.
```

### Deleting all users

You can remove all user accounts on the DX appliance.

To delete all users, except for the default account, type:

```
dx% delete user all
```

For example:

```
dx% delete user all
Are you sure you want to delete all users (y/n)? [n] y
User fred deleted
User tom deleted.
User dick deleted.
User harry deleted.
```

To delete all user accounts, except the default account, and reset the DX appliance to its factory default settings, type the following commands while logged in as the default user:

```
dx% delete user all
dx% reset config
```

### Exporting and Importing User Accounts

You can export and import user account information for backup purposes or to match user accounts across multiple DX appliances. Exporting and importing requires:

- Access to the command line
- A Trivial File Transfer Protocol (TFTP) or Secure Copy (SCP) server
- A user account file

### Creating the User Account File

A user account file consists of the list of DXSHELL commands required to completely recreate the user accounts. You can use a text editor to customize the account information, removing or commenting out commands. Lines beginning with a pound sign (#) are comments and are ignored.

To view the commands needed to recreate the user accounts, type:

```
dx% display users
# Juniper Networks Config Version 5.1.B27
delete user all
set user admin password $1$0vCsR$q/eG9BZtP.MAU3cXEmesc0
set user admin enabled
```

### Exporting User Accounts

To export user accounts, use one of the following commands:

```
dx% export users tftp://<tftpservername>/<accountsfilename>
dx% export users scp://<scpservername>/<accountsfilename>
```

### Importing User Accounts

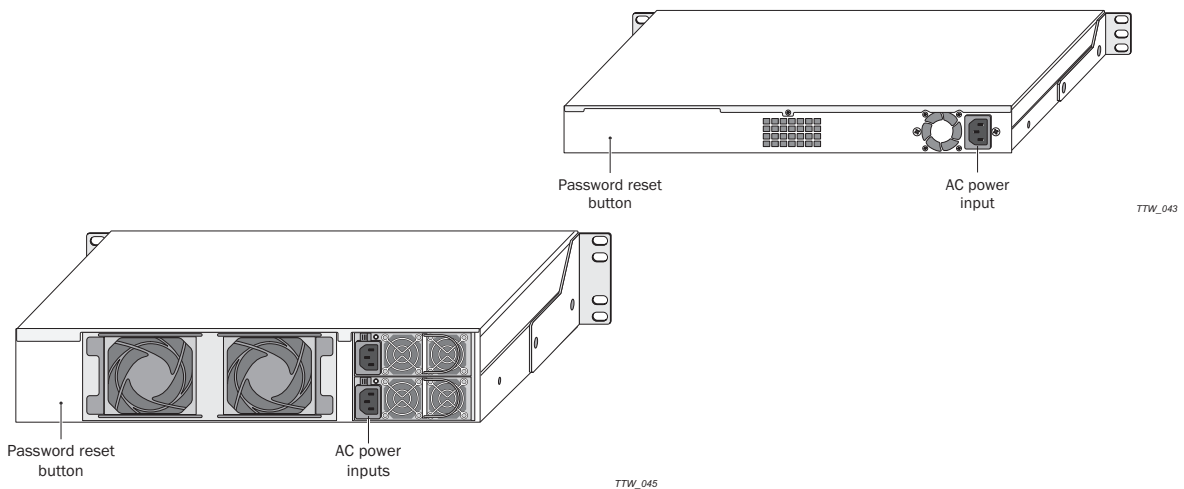
To import user accounts, use one of the following commands:

```
dx% import users tftp://<tftpservername>/<accountsfilename>
dx% import users scp://<scpservername>/<accountsfilename>
```

### Resetting the admin User Password

If you forget or lose the password for your DX Application Acceleration Platform, you can reset the password to the default value. Insert a paper clip into the PASSWORD RESET button access point on the rear of the DX. Refer to Figure 45.

**Figure 45: Resetting the DX Appliance Password**



When you feel the paper clip come in contact with a button directly inside the box, hold it for four seconds. This action causes the following behavior:

- The default account is enabled and its password is reset to its default value.
- Any open administrative sessions are closed.
- SSH, Telnet, and WebUI access to the DX appliance is disabled.

This action does not affect any other user accounts or the appliance configuration settings. You do not need to repeat the First-Time Configuration Program. However, to set a new password you do need to connect to the DX through the console port. You will be prompted for the default username (**admin**) and password (**admin**), and once you have entered these, you can set a new password with the following command:

```
dx% set password
```

### **Administrator Remote Authentication**

Administrator Remote Authentication allows a properly-enabled administrator to log onto and administer the DX using the Command Line Interface (CLI) from anywhere in the world. The connection uses a secure protocol (DAP and RADIUS) for remote authentication.

There are two classes of users (administrators of the DX): local and remote. By default, when a new user is added, his class is set to local. The class of a user is set to remote using the DXSHELL command:

```
dx% set user <user> class <local | remote>
```

Remote authentication is only performed for users whose class is remote.

For all the users, the assigned roles are stored locally on the DX. Because of this, all users, local or remote, have to be added on the DX. For a remote user, a password does not have to be set on the DX because the authentication is handled by the authentication server.

The default role for the remote users is “user.” If no role is specifically set for a remote user, the default role is used. The default role can be changed using the command:

```
dx% set admin remotearch userrole <role>
```

The login class is used to differentiate between a local and a remote user. Currently LDAP and RADIUS are supported for remote authentication; the default protocol is RADIUS. It can be changed using the command:

```
dx% set admin remotearch protocol <ldap | radius>
```

When a user tries to login, through the console, telnet, SSH, and WebUI, the DX uses this logic to authenticate:

- If the user is a local user, authentication takes place locally as before.
- If the user is a remote user:
  - If remote authentication is not enabled, the user login is refused
  - If the required current protocol (LDAP or RADIUS) configuration is not present, the user login is refused.
  - If remote authentication is enabled and all of the required current protocol (LDAP or RADIUS) configuration is present:
    - LDAP or RADIUS server 1 is contacted for authentication
    - If there is a communication error with server 1, server 2 is contacted for authentication.
    - If authentication does not succeed, user login is refused.
    - If authentication succeeds:
      - A remote authorization user role is assigned to the user.
      - If not, the default role is assigned.

### Remote Authentication Configuration Commands

These commands are used to configure Administrator Remote Authentication.

#### Set Commands

To enable or disable Administrator Remote Authentication, type the command:

```
dx% set admin remotemauth status <enabled | disabled>
```

To set the authentication protocol to use for Administrator Remote Authentication, type the command:

```
dx% set admin remotemauth protocol <ldap | radius>
```

To set the default role for remote users, type the command:

```
dx% set admin remotemauth userrole <role>
```

The default role is “user”.

To set the class attribute of a user, type the command:

```
dx% set user <user> class <local|remote>
```

To set the RADIUS server password, type the command:

```
dx% set admin remotemauth radius server key <key>
```

To set the IP address for RADIUS server 1, type the command:

```
dx% set admin remotearch radius server 1 ip <ip>
```

To set the port for RADIUS server 1, type the command:

```
dx% set admin remotearch radius server 1 port <port>
```

To set the IP address for RADIUS server 2, type the command:

```
dx% set admin remotearch radius server 2 ip <ip>
```

To set the port for RADIUS server 2, type the command:

```
dx% set admin remotearch radius server 2 port <port>
```

To set the Distinguished Name (DN) of the node in the LDAP Directory Information Tree, under which the users have to be searched, type the command:

```
dx% set admin remotearch ldap basedn <base-dn>
```

To set the attribute name that uniquely identifies the user in LDAP database, type the command:

```
dx% set admin remotearch ldap uid <uid>
```

To set the Distinguished Name of the LDAP admin user, type the command:

```
dx% set admin remotearch ldap bind userdn <user-dn>
```

The DX authenticates itself with the LDAP servers using this user DN.

To set the password for the LDAP admin user, type the command:

```
dx% set admin remotearch ldap bind password <password>
```

To set the IP address for LDAP server 1, type the command:

```
dx% set admin remotearch ldap server 1 ip <ip>
```

To set the port for LDAP server 1, type the command:

```
dx% set admin remotearch ldap server 1 port <port>
```

To set the IP address for LDAP server 2, type the command:

```
dx% set admin remotearch ldap server 2 ip <ip>
```

To set the port for LDAP server 2, type the command:

```
dx% set admin remotearch ldap server 2 port <port>
```

### **Show Commands**

To display the current status of remote authentication, type the command:

```
dx% show admin remotearch status
```

To display the current remote authentication protocol, type the command:

```
dx% show admin remoteauth protocol
```

To display the default user role for remote users, type the command:

```
dx% show admin remoteauth userrole
```

To display the RADIUS server key, type the command:

```
dx% show admin remoteauth radius server key
```

To display the IP Address and port for RADIUS server 1, type the command:

```
dx% show admin remoteauth radius server 1
```

To display the IP Address and port for RADIUS server 2, type the command:

```
dx% show admin remoteauth radius server 2
```

To display the LDAP base-dn, type the command:

```
dx% show admin remoteauth ldap basedn
```

To display the LDAP uid, type the command:

```
dx% show admin remoteauth ldap uid
```

To display the LDAP admin user-dn, type the command:

```
dx% show admin remoteauth ldap bind userdn
```

To display the LDAP admin password, type the command:

```
dx% show admin remoteauth ldap bind password
```

To display the IP Address and port for the LDAP server 1, type the command:

```
dx% show admin remoteauth ldap server 1
```

To display the IP Address and port for the LDAP server 2, type the command:

```
dx% show admin remoteauth ldap server 2
```

## Obtaining a License Key

---

The DX Application Acceleration Platforms have a built-in permanent license that allows access to all of the standard Server Load Balancer and SSL termination features. To take advantage of the HTTP acceleration, transport connection multiplexing, adaptive content processing (using the Overdrive Application Rules and 3G Caching), or global server load balancing, you need one or more additional licenses. See “DX Product Licensing Options” on page 17 for details about which licenses you may need.

To obtain a permanent license for these more advanced features, you need the following:

- A Juniper Customer Support Center (CSC) User ID and Password
- The device serial number (displayed on back of device)
- An Authorization Code used to generate the license for the DX platform and each optional feature

The Authorization Code is provided in the *Juniper Right To Use* document that is E-Mailed to the address specified on the Purchase Order (Figure 46).

**Figure 46: Example of the Juniper Right to Use Certificate**

Juniper Networks, Inc.  
1194 N. Mathilda Avenue  
  
Sunnyvale, CA 94089  
United States

Date Issued: 05 AUG 2005  
Order Number: 1067299  
Quote Line: 1

Customer PO: 0002640  
Part Number: 100-10000000-1-10  
Part Description: Adaptive content processing module

---

**Juniper RTU (Right to Use) Certificate**

**Instructions:**

New Juniper products typically ship with a temporary unlimited license that expires after a period of time if a permanent license key is not installed (see product literature for exact expiration period).


To obtain a permanent license key for new products or upgrade license keys for already deployed products, please visit [https://www.juniper.net/generate\\_license](https://www.juniper.net/generate_license) to convert the below Authorization Codes into license keys.

Once you have obtained the license keys they can be loaded onto your hardware to unlock your purchased features.

If you have any questions about the license generation process, please contact your reseller or distributor.

Item#	RTU Serial Number	Authorization Code
1	RTU00000 000000002	00000-unin-00000-00000
2	RTU00000 000000003	00000-unin-00000-00000

Authorization Code needed to generate the permanent license





**NOTE:** Software upgrades retain the Permanent License keys (see Upgrading the DX Application Acceleration Platform Software on page 146).

### **Obtaining a Juniper Customer Support Center (CSC) User ID and Password**

Before you can obtain a permanent license, you must have a Juniper Customer Support Center (CSC) User ID and Password. There are two ways to obtain these:

- Call Juniper Customer Care at 1-800-638-8296 (United States) or + 1-408-936-1572 (outside the United States)
- Register online at the Customer Support Center:

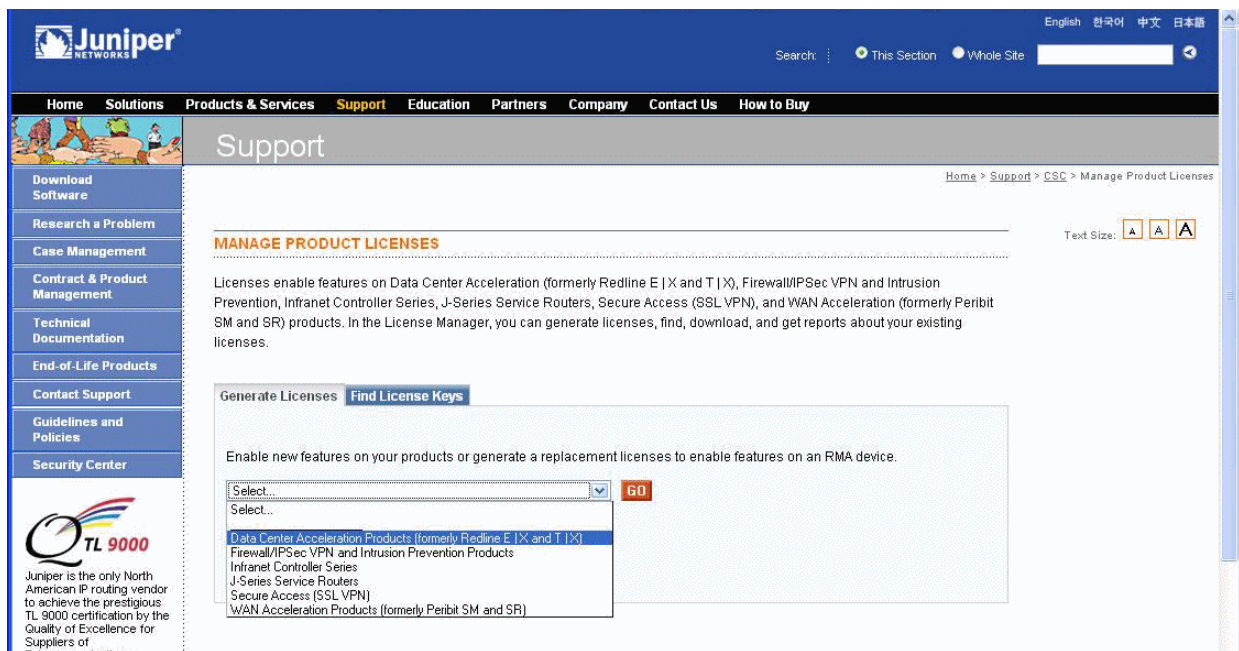
[https://www.juniper.net/generate\\_license](https://www.juniper.net/generate_license)

## Obtaining a Permanent License

To obtain a permanent license:

1. From a standard Web browser, go to the following Web site:  
[https://www.juniper.net/generate\\_license](https://www.juniper.net/generate_license)
2. Log in using your assigned User ID and Password.
3. Select the "Data Center Acceleration Products (formerly Redline E|X and T|X)" link (Figure 47) and click on the **Go** button.

**Figure 47: Manage Product Licenses Screen**



The Generate License Keys screen is displayed (Figure 48).



Figure 48: Generate License Key Screen

English 한국어 简体 日本語

Search: This Section Whole Site

Home Solutions Products & Services Support Education Partners Company Contact Us How to Buy

Support

Home > Support > CSC > Manage Product Licenses > Generate Licenses - Data Center Acceleration Products

**GENERATE LICENSE KEYS : DATA CENTER ACCELERATION PRODUCTS (FORMERLY REDLINE E | X AND T | X)**

Generate Licenses for Data Center Acceleration Products (formerly Redline E | X and T | X) using Serial Numbers and Authorization Codes.

\* Indicates required items

**NOTICE: DXOS 5.0 and Lower**  
You must contact [Juniper Customer Care](#) for License Key generation for devices running DXOS 5.0 and lower

Device Serial Number

Authorization Code  [Enter More Authorization Codes](#), for the same device  
For e.g., aBC1-D2eF-3dGh-5jJK

**GENERATE** **CANCEL**

[Generate License Keys for multiple Data Center Acceleration Products \(formerly Redline E | X and T | X\)](#) by uploading a Microsoft Excel(\*.xls) spreadsheet containing Authorization Codes

Copyright © 1998-2004, Juniper Networks, Inc. All Rights Reserved. [Trademark Notice](#) [Privacy](#) Juniper your Net. Feedback

4. Enter the serial number for your DX Application Acceleration Platform.
5. Enter your Authorization code.
6. Click on the **Generate** button to display the license key in a new window.

The license key should be similar to the following:

```
-----BEGIN JUNIPER LICENSE KEY-----
3418fa1db0e0ae0552cb79c472a076b15a9a4b51d0e2f6a54fcfa97e4b04b20f
29f4149978330387d102e076805e884bcf0f14d023db999b79651e140c732431
f3b3b815d3cf3eb593060e917c458defe8267da03a3ee3101d99c9becd34643d
6184fc028ff719bcd451f87ad431f90c28d6c68e85d105443edfbfe772d7df8b
426f3cd08ba32863c37ba856139af4169d7102f53aabb3f688a8b171c3446e9
f0819b23dd4f0bea49c9ae3b1ecb9feef5361ca3a9
-----END JUNIPER LICENSE KEY-----
```

7. Copy the key from the browser window and paste it into a text document.

You are now ready to install the license key.

## Installing the DX License Key

If you are upgrading from a version of the software earlier than 3.1, you will need to install the DX license key. If you are upgrading from version 3.1 or later, the previously installed license key is preserved, and you do not need to reinstall it.

1. If you have not already done so, obtain the license key specific to the serial number of your DX Application Acceleration Platform as detailed in “Obtaining a Permanent License” on page 126. Each license key is tied to the serial number, and works ONLY on the specified DX.
2. From an attached console, enter:
 

```
dx% capture license
```
3. Copy and paste the license key from your test file into the console.
4. Enter the `write` command to save the changes to your configuration.

The features associated with the license you installed should now be available. You do not need to restart the DX appliance.

## Using the Administrator Audit Trail

---

The Audit Trail provides a log of all activities performed on the DX Application Acceleration Platform . Specifically, it provides the following information:

- Timestamp
- Location of the source
- Username
- Access method: Web User Interface (WebUI) or DXSHELL
- Changes and activities performed



By default, the audit trail is not enabled. You must configure and enable logging for entries to be added to the audit log.

---

## Syntax of the Log Entries

The audit trail log uses the following syntax:

```
<Timestamp> <Location> <Username> <Tool> <Change>
```

where:

- **<Timestamp>** : The time of the change, displayed in [YYYY-MM-DD HH:MM:SS (TZ Offset)] format. For example: [2004-10-04 22:17:09 (+0700)].
- **<Location>** : The information of the system that made the change. It can either be the IP address of the workstation, or the reserved word “console” if the change is made from the serial console.

- **<Username >** : The username of the user that made the change. When changes are made by a module internal to the DX, the username listed is “RLN Internal”.
- **<Tool >** : Indicates if the change was from DXSHELL, the Web User Interface (WebUI), or an internal module. The word “DXSHELL” indicates that the change was made from the command line interface using the serial console, SSH or Telnet. The word “WebUI” indicates that the change was made from the WebUI. The word “system” indicates that the change was made by a DX.
- **<Change >** : Brief description of what was changed.

To display the audit trail, use the following command:

```
dx% show log audit
```

You can press any key to scroll through the log, or press **Esc** to return to the command line before reaching the end of the log.

### **Enabling and Disabling Logging of show Commands**

The logging of **show** commands is disabled by default. It can be enabled and subsequently disabled using DXSHELL or the Web User Interface (WebUI).

To enable logging of **show** commands using DXSHELL:

```
dx% set admin audit showcmd enabled
```

To disable logging of **show** commands using DXSHELL:

```
dx% set admin audit showcmd disabled
```

All commands executed on DXSHELL are logged when a **write** operation is performed.

To view whether logging of **show** commands is currently enabled or disabled, type:

```
dx% show admin audit showcmd
Show cmds admin logging: enabled
```

## **Configuring System Event Logging and Notification**

---

To monitor system performance and status, the DX platform keeps a log of a number of system events. You can also configure the DX platform to report certain classes of events using E-Mail or logging them to an external syslog logging machine.

There are four ways to record DX system events:

- **console**: Shows log events on the system console
- **e-mail**: Sends notifications of log events to specified E-Mail addresses
- **syslog**: Sends log events to an external syslog logging server

- memory: Records log events on the DX appliance

There are two levels of events:

- EMERG: Requires immediate attention
- ALERT: Informational in nature

If you set your alert level to ALERT, both EMERG and ALERT notices are generated. If you set your alert level to EMERG, only EMERG notices are generated. For a complete list of events for each level, refer to “List of Events” on page 433.

### Viewing Event Logging Configuration

You can view the configuration of event logging by typing the command:

```
dx% show admin log
Logging: disabled
Logging to:
  email: (none)
  memory: ALERT
  syslog: (none)
Email 'mailto' addresses:
  mailto1:
  mailto2:
```

### Viewing the System Event Log

You can view the system event log by typing the command:

```
dx% show log system
```

The syntax of the system log is as follows:

```
[timestamp][eventlevel][eventdescription]
```

where:

- [Timestamp]: The time the event occurred, displayed in [YYYY-MM-DD HH:MM:SS (TZ Offset)] format. For example: [2006-01-04 22:17:09 (+0700)].
- [EventLevel]: Indicates whether the event was an EMERG or ALERT level event.
- [EventDescription]: Brief description of action that triggered the event.

The following is an excerpt from a system log file:

```
[2006-02-01 18:43:40 (+0800)][ALERT][FO][Going Standby]
[2006-02-01 18:43:40 (+0800)][EMERG][FO][Be Standby to avoid conflict]
[2006-02-01 18:43:40 (+0800)][EMERG][FO][Peer (10.80.96.16) with smaller uptime]
[2006-02-01 18:43:34 (+0800)][ALERT][FO][Wait 1 cycle]
[2006-02-01 18:43:34 (+0800)][EMERG][FO][Peer (10.80.96.16) with smaller uptime]
[2006-02-01 18:43:34 (+0800)][ALERT][FO][10.80.96.16 is now reachable]
[2006-02-01 18:42:37 (+0800)][ALERT][FO][ether0 is up - send DP broadcast]
[2006-02-01 18:42:36 (+0800)][ALERT][LOG_][em0: Link is up 100 Mbps Full Duplex]
```

## Sample Configuration

The following example shows how to configure your DX appliance to receive e-mail notification of Layer 7 health check errors. It first enables event logging, then specifies an SMTP server to relay the notification. Next, the To and From e-mail addresses are specified followed by the event level appropriate for receiving Layer 7 health check errors.

```
dx% set admin log enabled
dx% set admin email server <IP address or hostname of SMTP
server>
dx% set admin email from <e-mail address>
dx% set admin log mailto1 <e-mail address>
dx% set admin log mailto2 <e-mail address>
dx% set admin log email ALERT
```



You can specify up to two e-mail addresses to receive event notifications.

## Managing Your DX Appliance Configuration

You can export and import an DX's configuration for backup purposes or to synchronize the configuration across multiple DX appliances. You must have access to DXSHELL and a TFTP or SCP server.

This section contains the following topics:

- “Exporting a Configuration” on page 131
- “Importing a Configuration” on page 132
- “Viewing the Contents of a Configuration File” on page 133
- “Editing a Configuration File” on page 133
- “Restoring the Factory Default Configuration” on page 135
- “Creating a DX Platform System Image” on page 135
- “Synchronizing Configurations Across Multiple DX Appliances” on page 138

### Exporting a Configuration

An exported DX configuration file consists of the list of DXSHELL commands required to completely recreate a configuration. Upon importing a configuration file, the DX platform runs the commands in the file to reproduce the configuration.

To export a configuration, enter one of the following commands:

```
dx% export config tftp://<tftpservername>/<configfilename>
dx% export config scp://<scpsservername>/<configfilename>
```

The `tftpservername` and `scpsservername` are a host name or an IP address. The `configfilename` is an absolute path for where you would like to export the configuration.

### Guidelines for Exporting a Configuration

Be aware of the following items when you are exporting a configuration:

- The directory specified for the filename must exist.
- We recommend that the configuration file name describes the function and identifies the version, for example, `dx_5.1.B28_ssl_server_2-25-2006`.
- The following settings are NOT exported:
  - Passwords
  - Commands that control the DX server's state
  - Commands that control the state of administrative services  
For example: SSH, Telnet and WebUI access
  - SSL Keys and Certificates  
See “Importing Existing Keys and Certificates” on page 197 or “Generating Keys and Certificates” on page 206 for more information.
  - User account Information  
See “Exporting and Importing User Accounts” on page 119 for more information.
  - Application rules
  - License keys  
See “Obtaining a License Key” on page 124 for more information.

### Importing a Configuration

To import a configuration:

1. Enter one of the following commands:

```
dx% import config tftp://<tftpservername>/<configfilename>
```

```
dx% import config scp://<scpsservername>/<configfilename>
```

2. Apply and save the new configuration:

```
dx% write
```

### Guidelines for Importing a Configuration

Be aware of the following items when you are importing a configuration:

- If any errors are encountered in the configuration file, the DX platform generates an error message and stops the import process.
- SSL keys and certificates must already be installed on the DX.

### Viewing the Contents of a Configuration File

You can view the contents of the DX appliance's current configuration file at any time to verify configuration changes or to determine the correct file to export or import.

To view the active configuration file, enter the following command:

```
% display config
```

For example, the beginning of a configuration file would be similar to the following:

```
dx% display config
# Juniper Networks Config Version 5.1.B27
copy config factory memory
set ether 0 ip 10.10.16.32
set ether 0 media autoselect
set ether 0 mtu 1500
set ether 0 netmask 255.255.255.0
set ether 1 ip 10.10.1.2
set ether 1 media 100baseTX full-duplex
set ether 1 mtu 1500
set ether 1 netmask 255.255.0.0
set hostname dx-2.jnpr.net
set route default 10.10.16.1
#set clock 2006.02.16 11:27:44
set timezone America/Los_Angeles
set ntp server 1 192.168.0.2
set admin syslog facility LOG_USER
set admin syslog host 1 port 514
set admin syslog host 2 port 514
set admin upgrade transport tftp
set failover discovery interface ether 0
set admin audit showcmd disabled
set admin cli sessionExpireTime 600
set admin log memory ALERT
...
```

### Editing a Configuration File

Configuration files can help you match common settings across multiple DX appliances. A freshly exported configuration file contains settings for attributes such as IP addresses that are particular to a single DX appliance.

You can edit the configuration file, removing distinct settings or commenting out commands, to create a general configuration file that your DX appliances can share. You can also create a partial configuration file to match a particular subset of settings across multiple DX appliances.

To edit a configuration file:

1. Export the configuration file, for example:

```
dx% export config tftp://192.168.0.11/juniper_sticky.conf
```

2. Open the configuration file in a text editor.
3. Enter pound signs (#) at the beginning of the command lines that you do not want to execute, or remove the lines altogether.

For example, if you do not wish to change the clock setting on the machine receiving the new configuration, look for the `set clock` command and add a pound sign in front:

```
#set clock 2006.02.16 11:27:44
```

4. Add any useful comments to the configuration file.

For example:

```
# This configuration file sets logging on all DX appliances.
```



A freshly exported configuration file begins with the **copy config factory memory** command. This command resets all settings to factory defaults before applying the commands in the configuration file. To avoid losing your configuration, DO NOT include this command in customized partial configuration files.

---

### Example: Partial Configuration for Sticky Load Balancing

You can create a partial configuration file to match sticky load-balancing settings across several DX appliances. Edit the existing configuration file on a DX appliance and then import this file onto the selected DX appliances.

This example uses a TFTP server address of 192.168.0.11 and a configuration file name of `juniper_sticky.conf`.

1. Export the configuration file from the DX appliance that contains the desired sticky load-balancing settings:

```
dx% export config tftp://192.168.0.11/juniper_sticky.conf
```

2. Open the file in a text editor and remove all commands not related to sticky load balancing (including the `copy config factory memory` command), leaving only the following commands in the file:

```
set cluster 1 sticky clientip distribution internet
set cluster 1 sticky clientip timeout 120
set cluster 1 sticky cookie expire 0
set cluster 1 sticky cookie mask ipport
set cluster 1 sticky method none
```

3. Save your changes.



4. On each of the DX appliances where you wish to apply the sticky load-balancing settings, enter the command:

```
dx% import config tftp://192.168.0.11/juniper_sticky.conf
```

### Restoring the Factory Default Configuration

To erase all custom settings and return to the factory default configuration:

1. Reset the configuration:

```
dx% reset config
```



If you are connected remotely to a DX appliance, you must enter valid network settings BEFORE applying the new configuration or you will no longer be able to connect to the DX appliance. If you leave the factory network settings in place, you will have to connect to the console port on the DX appliance to enter new network settings.

2. Save and apply the factory default configuration:

```
dx% write
```

### Creating a DX Platform System Image

You can create an image of the system on a DX appliance, including IP addresses, system files, and licenses using the System Snapshot feature. System Snapshot creates an effective backup of not only the configuration, but also the underlying operating system. It can also create base system images that can be used to clone new machines as needed.

After you have imported a system snapshot and rebooted the DX system, all services that were running on the original machine at the snapshot was taken will start on the machine that imported the system snapshot. System snapshot is imported into an unused partition. Importing a snapshot does not impact the running (active) partition.

Each DX has a manufacturing information file that contains the unit serial number, manufacturing date, model number and platform at the time of manufacturing. This information is not overwritten when importing a system snapshot.

Using System Snapshot, you can:

- Repair by replacement, providing more system uptime for a user. Field units that have failed can be quickly recreated using replacement hardware units and a system snapshot of the former unit.
- Recover from configuration mistakes. Administrators can revert to a known system snapshot should they ever want or need to return to a previous configuration.

- Secure your system configuration and licenses. All of the information that is exported during a system snapshot is encrypted. It is not casually visible if viewed off the DX appliance.



You must have administrator access to DXSHELL and an SCP server to create a system snapshot.

---

### Exporting a System Snapshot

To export the system snapshot, enter the following command:

```
dx% export snapshot system scp://<server>/<path>/<resource>
```

For example:

```
dx% export snapshot system scp://myarchive/usr/cvs/snap_juniper1
Creating...
myuser@myarchive's password:
Successfully exported snapshot.
```

### Importing a System Snapshot

To restore a previously saved system snapshot to a DX platform, enter the following command:

```
dx% import snapshot system scp://<server>/<path>/<resource>
```

For example:

```
dx% import snapshot system scp://myarchive/usr/cvs/snap_juniper1
```

During the import process, the system displays a variety of messages, some of which require a response:

- Verification of snapshot import

```
WARNING - This will import a snapshot and install it to the alternate
partition, overwriting its current contents.
```

```
After the import, you will be able to choose whether to use the snapshot or
currently active settings for your license and network configuration. You
can use the 'set boot' command to select the default boot partition once
this process is complete.
```

```
Would you like to continue (y/n)? [n]
```

- If you answer negatively, the process is stopped and the following message is displayed:

```
Import aborted.
```

- If you answer positively, the process continues and the following messages are displayed:

```
Receiving file /usr/cvs/snap_juniper1 from scp server myarchive...
myuser@myarchive's password: Enter password
Bytes received: 13054697
```

```

Verifying...
Decrypting...
Unpacking...
Installing.....
Verifying install.....
Doing post-install setup...
Done.
    From snapshot: Currently active license:
License:    INVALID Valid

```

The license from the snapshot is used by default. Would you like to use the currently active license instead (y/n)? [n]

[Installing currently active license to | Using snapshot license on] alternate partition.

```

    From snapshot:Currently active settings:
Hostname: dx-1.domain.com  dx-2.domain.com
Default route: 192.168.0.1 10.0.51.1

```

```

Ether 0 IP Address: 192.168.14.20 10.0.51.80
Ether 0 Netmask: 255.255.0.0 255.255.255.0
Ether 0 Media: 100baseTX full-duplexautoselect
Ether 0 MTU: 1500 1500

```

```

Ether 1 IP Address: 192.168.14.2110.0.51.81
Ether 1 Netmask: 255.255.0.0255.255.255.0
Ether 1 Media: autoselect autoselect
Ether 1 MTU: 1500 1500

```

- If there are more Ethernet cards on the snapshot than in the current machine, you see a dialog similar to this:

```

Ether 2 IP Address:192.168.14.20Ether 2 not present
Ether 2 Netmask:255.255.0.0n/a
Ether 2 Media:100baseTX full-duplexn/a
Ether 2 MTU:1500n/a

```

- If there are more Ethernet cards in the current machine than on the snapshot, you see a dialog similar to this:

```

Ether 2 IP Address:Ether 2 not present10.0.51.82
Ether 2 Netmask:n/a255.255.255.0
Ether 2 Media:n/a100baseTX full-duplex
Ether 2 MTU:n/a1500

```

The network settings from the snapshot are used by default. Would you like to use the currently active settings instead (y/n) ? [n]  
 [Installing currently active network settings to | Using snapshot network settings on] alternate partition.  
 Import snapshot successful. Use 'set boot' to activate the new partition, and 'reboot' to switch to it.

## ■ Stopping the server

In most cases, the server can continue to run when system snapshots are exported and imported. However, under certain conditions caused by memory constraints, the server may have to be stopped before exporting or importing the snapshot. In these cases, the user is prompted to stop the server before export or import, and to restart the server afterward.

The following example shows an export process in which the server must be stopped:

```
dx% export snapshot system scp://myarchive/usr/cvs/snap_juniper
WARNING - Because of memory constraints, the server will be stopped
        if the export continues.
Would you like to continue (y/n)?[y]
Running 'set server down'...
The EIX server was stopped.

Creating...
myuser@myarchive1s password:
Successfully exported snapshot.
The server is currently down.
Would you like to start it now (y/n)?[y]
Running 'set server up'... The EIX server was started.
```

## Synchronizing Configurations Across Multiple DX Appliances

Configuration synchronization provides a mechanism for an administrator to copy the settings from one DX appliance to other DX appliances in a pre-defined group. This can be a significant time-saver for administrators with two or more DXs.



Configuration Synchronization is NOT supported on the DX 3650-FIPS version of the DX Application Acceleration Platform .

---

Examples of information that is synchronized across the group are:

- SSL Certificates, keys, and passwords
- OverDrive application rules
- Username and password combinations

Other settings are unique to a particular machine within in the group and are not synchronized. These settings are called exceptions and currently include:

- Hostname
- Ethernet IP addresses, netmask, and other interface settings
- Default route
- Administration VIP and administration interface IP addresses
- Administration SOAP settings

A synchronization override file is used to manage values for these synchronization exceptions. It contains a list of synchronization addendums that specify user-entered information to be synchronized across machines in the group. This synchronization override file can be used during future configuration synchronizations.



If Configuration Synchronization is performed in a production environment without using the synchronization override file to manage the synchronization exceptions, unexpected network behavior can result.

---

Synchronization is achieved through the use of a SOAP server. Simple Object Access Protocol (SOAP) is an XML-based protocol for exchanging information over the Internet. Commands are provided to manage both the Synchronization Group and the SOAP Server. For configuration synchronization to work properly, SOAP must be configured and enabled on the group member unit(s), otherwise configuration synchronization will fail. SOAP does not need to be configured and enabled on the reference unit, and will not cause problems if it is.

### Configuration Synchronization Override File Format

The synchronization override file is a text file, and while its entries are fully managed by the configuration synchronization feature, it also can be edited manually. The following is an example of a synchronization override file:

```
# DO NOT DELETE THIS LINE -- SYNC OVERRIDE FILE SIGNATURE

# Description: Sample manual override command file for
#             configuration synchronization.
#
# Note: The first line of this file must be the text inside the double
#       quotes:
#       "# DO NOT DELETE THIS LINE -- SYNC OVERRIDE FILE SIGNATURE"
#
# Example:
#  1) sync group consists of two member appliances: dx1 and dx2
#  2) cluster "1" exists in the configuration
#
# Here are some typical commands that might be unique for each member:
#
dx1 | set cluster 1 listen vip 192.168.0.10
dx1 | set activeN failover nodeid 1
dx1 | set slb failover nodeid 1

dx2 | set cluster 1 listen vip 192.168.0.20
dx2 | set activeN failover nodeid 2
dx2 | set slb failover nodeid 2
```



The first line of this file must be the following text:

```
# DO NOT DELETE THIS LINE -- SYNC OVERRIDE FILE SIGNATURE
```

---

## Configuration Synchronization Commands

To synchronize the configuration settings across a group of DX appliances, type the command:

```
dx% sync group <name>
```

Before this command can be executed, both the Synchronization Group and the SOAP server must have been set up correctly. When adding members to the synchronization group, the local DX appliance must be added as a reference unit along with all remote DX appliances that need to receive the group configuration. For example:

```
dx% add sync group test
Created sync group "test"
dx% add sync group test member 10.0.10.100
Created sync group "test" member "10.0.10.100".
dx% add sync group test member dx-1
Created sync group "test" member "dx-1".
dx% set sync group test member
10.0.10.100 dx-1
dx% set sync group test member 10.0.71.100 password
New password:
Retype new password:
dx% add sync group test member dx-1
Created sync group "test" member "dx-1".
dx% set sync group test member dx-1 password
New password:
Retype new password:

dx% show sync group
Sync Group [test]
Description:
Override Filename:
Override Status: disabled
Timeout: 180
Member:                Port:        Username:      Password:
10.0.10.100            8070        admin          *****
dx-1                   8070        admin          *****

dx% sync group test
Using the current appliance as the reference, you are about to
synchronize the configuration settings on the following appliances:
  10.0.10.100 (reference)
  dx-1

All settings except the following will be synchronized:
  Hostname
  All ether settings
  Default route
  Admin VIP and interface settings
  Admin soap settings

You have specified the following manual override file:
  dx70_override (enabled)
Would you like to continue (y/n)? [y]
Synchronizing member 10.0.10.100 ...
Success (skipping sync with localhost).
Synchronizing member dx-1 ...
Success.
Synchronization for group "test" finished successfully.
```

## Synchronization Group Management Commands

For each of the commands, `<memberid>` is either a `<hostname>` or an `<ip>`.

To create a synchronization group, type the command:

```
dx% add sync group <name>
```

To add a member to the synchronization group, type the command:

```
dx% add sync group <name> member <memberid>
```

To set the username for a synchronization group member, type the command:

```
dx% set sync group <name> member <memberid> username <username>
```

The default username is "admin".

To set the password for a synchronization group member, type the command:

```
dx% set sync group <name> member <memberid> password
```

This will prompt you for a password. For example:

```
dx% set sync group <name> member <memberid> password
New password:
Retype new password:
dx%
```

No asterisks will be shown as the password is typed.

To add a description for a synchronization group, type the command:

```
dx% set sync group <name> description <description>
```

To rename a synchronization group, type the command:

```
dx% set sync group <name> name <newname>
```

To enable the use of the group override file, type the command:

```
dx% set sync group <name> override enabled
```

To disable the use of the group override file, type the command:

```
dx% set sync group <name> override disabled
```

The default is disabled.

To set the name for the group override file, type the command:

```
dx% set sync group <name> override filename <filename>
```

To show all of the settings for the synchronization group, type the command:

```
dx% show sync group
```

To show all of the settings for a particular synchronization group, type the command:

```
dx% show sync group <name>
```

To show the settings for a particular synchronization group member, type the command:

```
dx% show sync group <name> member
```

To show the description for a particular synchronization group member, type the command:

```
dx% show sync group <name> description
```

To show the override status for a particular synchronization group, type the command:

```
dx% show sync group <name> override
```

To show the override filename for a particular synchronization group, type the command:

```
dx% show sync group <name> override filename
```

To delete a synchronization group, type the command:

```
dx% delete sync group <name>
```

To delete a member from a synchronization group, type the command:

```
dx% delete sync group <name> member <memberid>
```

### **SOAP Server Management Commands**

To enable the Simple Object Access Protocol (SOAP) server, type the command:

```
dx% set admin soap up
```

The default is up.

To disable the SOAP server, type the command:

```
dx% set admin soap down
```

To set the port number for the SOAP server, type the command:

```
dx% set admin soap port <portnum>
```

The default port is 8070.

To set the SSL certfile filename for the SOAP server, type the command:

```
dx% set admin soap ssl certfile <filename>
```

The default file name is democert.



To set the SSL key file for the SOAP server, type the command:

```
dx% set admin soap ssl keyfile <filename>
```

The default file name is demokey.

To set the SSL key password for the SOAP server, type the command:

```
dx% set admin soap ssl keypass <password>
```

To show all of the configuration parameters for the SOAP server, type the command:

```
dx% show admin soap
```

To show the port number for the SOAP server, type the command:

```
dx% show admin soap port
```

To show all of the SSL configuration parameters for the SOAP server, type the command:

```
dx% show admin soap ssl
```

To show the SSL certfile filename for the SOAP server, type the command:

```
dx% show admin soap ssl certfile
```

To show the SSL key file filename for the SOAP server, type the command:

```
dx% show admin soap ssl keyfile
```

To show the SSL key file password for the SOAP server, type the command:

```
dx% show admin soap ssl keypass
```

## Configuring the Login Banner

---

Some users have corporate policies that require them to display a login banner to users when they are accessing corporate computer systems and networks. You can customize the welcome message that is displayed on either the Juniper Command Line or on the WebUI when a user logs in. Currently, when a user logs in using DXSHELL, the default message is displayed:

```
Welcome to Juniper Networks
DX
Application Acceleration Platform
```

The login banner could be changed to include any message that you would like to display. For example:

```
Unauthorized access to or use of this system is prohibited. All access
and use may be monitored and recorded.
```

The maximum length of the text string is 2000 characters. The banner allows for some printf-style substitutions, as follows:

```
%h hostname
%d date
%s system ("Juniper")
%v product version
%b product build id
%% show the percent character
```

When the banner display encounters one of these substitution strings, it extracts the information from the appropriate place in the operating system and displays it. This information cannot be changed by the user.

The banner cannot be exported. The banner is preserved when installing newer versions of software for the DX.

### Configuring the Login Banner from the Command Line Interface

The login banner can only be configured from the Command Line Interface. These commands are available to configure the banner:

**dx% capture loginbanner**

This command begins capture of the login banner. After typing this line has been entered, everything that you type (up to the 2000 character limit) will be captured as part of the banner to be displayed. Carriage returns can be included, and you can use copy and paste commands to make the capture process easier. End capture of the banner by typing a period on a blank line. This command can only be executed by a user with a role of "administrator."

**dx% display loginbanner**

This command displays the banner in its raw form. Substitution strings are shown in their normal form (%h) instead of the substitution form (hostname). The **display loginbanner** command can only be executed by a user with a role of "administrator."

**dx% delete loginbanner**

The `delete loginbanner` command deletes the banner. It can only be executed by a user with a role of “administrator.”

**dx% show loginbanner**

The `show loginbanner` command shows the banner with the appropriate substitutions. It is available to all users.

**Capturing a Login Banner**

Follow these steps to capture a login banner:

1. Type the `capture loginbanner` command:

**dx% capture loginbanner**

2. Enter the information that you want to display. End with a period on a blank line.

```
Unauthorized access to or use of this system is prohibited.
All access and use may be monitored and recorded.
%h
%d
%s
Put anything else that you want here . . .
.
```

3. Type the `show loginbanner` command to show the banner with the appropriate substitutions:

```
Unauthorized access to or use of this system is prohibited.
All access and use may be monitored and recorded.
MyFirstHost
4 July 2004
Juniper
Put anything else that you want here . . .
```

**Displaying the Login Banner in the Web User Interface**

The WebUI does not provide the capability to set the login banner. Instead, the text string that was set by the administrator using the `capture loginbanner` command is displayed as part of the WebUI login screen.

For example, if the administrator sets the login banner “Welcome to the World of Juniper Networks” using the command string:

```
dx% capture loginbanner
Welcome to the wonderful world of Juniper.
.
```

A new instance of the WebUI will display the following:



You can put HTML in your login banner, and it will display correctly on the WebUI. However, the DX does not parse out HTML code when displaying the banner on the Command Line Interface, so the HTML code will be displayed along with the desired banner.

## Upgrading the DX Application Acceleration Platform Software



When upgrading, the DX should not be handling live traffic, as the upgrade will interrupt the traffic flow and requires a reboot.

### Viewing Enabled DX Platform Features

A DX unit must have a license key installed to enable software features running on the system. The license key provides information about the hardware and the software features that the system is licensed to run. To see the features that are currently enabled in your system, type the command:

```
dx% show license
```

If you do not have a license key, or the license key is missing for the DX, follow the instructions in “Obtaining a License Key” on page 124.

## Upgrade Requirements

To upgrade a DX unit, you need:

- A TFTP or SCP server to hold the upgrade file
- An upgrade file (.pac) that corresponds to your DX model

The procedure for configuring a TFTP or SCP server varies from system to system. If you need to set up a TFTP or SCP server, consult the documentation for your operating system.

Upgrade files can be obtained from your Juniper Networks sales representative or reseller, or from the Juniper Networks Technical Support Web site.

## Preserving Your Configuration and Choosing a .pac File

1. Put the upgrade file on your TFTP or SCP server. By default, most unix-based TFTP servers expect files to be located in `/tftboot`.
2. Preserve the DXs configuration.

The steps required to preserve your configuration depends upon which software release your DX is running. To determine the version number of the software release your DX is running, use the command:

```
dx% show version
```

### For Version 2.1 and Greater

If your DX is running release 2.1 or later, your configuration will automatically be preserved when you upgrade.

### For Versions Prior to 2.1

If you are upgrading from a version prior to 2.1, you will have to re-configure the DX after upgrading, so be sure to write down your configuration before upgrading. To view a complete summary of your configuration, use the command:

```
dx% show config
```

3. Configure the DX to use your TFTP or SCP server.
  - a. Give the DX the IP address of your TFTP or SCP server:

```
dx% set admin tftp server <IP address of TFTP server>
```

- b. Tell the DX which file to retrieve from the TFTP server:

```
dx% set admin upgrade filename <name of upgrade .pac file>
```

4. Save the changes:

```
dx% write
```

5. Verify the TFTP configuration:

```
dx% show admin upgrade filename
dx% show admin tftp
```

### **Upgrading Using the *install* Command**

The `install` command preserves the current version of the firmware. With the `install` command, you keep the current working version on the active partition, while installing the newer version into the non-active partition. This allows you to test the new firmware and easily revert to the previous version if needed. Note that the `.pac` file for the `install` command is approximately 14 MBytes.

You must be logged in as a user with a role of `administrator` to perform the install procedure.

1. Ensure that the DX appliance is not actively handling traffic. by typing the command:

```
dx% set server down
```

2. View the setup to see the partition where the new firmware will be installed by typing:

```
dx% show boot
```

A sample output is shown as follows:

```
Boot 1 (cur, act) : Juniper Networks Accelerator DX 2.3.3 Wed Mar 12
20:35:50 GMT
Boot 2 : Empty partition
```

The current partition (`cur`) is the partition that is currently running. The active partition (`act`) is the one that will be used after the reboot. In this example, the current partition and the active partition are the same.

3. Copy the `install.pac` file to a TFTP or SCP server that is accessible by the machine being upgraded.
4. Configure the TFTP or SCP server and the name of the install file on the DX by typing:

```
dx% set admin tftp server <IP address or Hostname of the tftp server>
dx% set admin upgrade filename <install_filename.pac>
dx% write
```

Where `install_filename.pac` is the filename of the install file on the TFTP server.

5. Store a copy of the existing configuration for backup purposes by typing:

```
dx% export config tftp://<tftp_server>/dx_configuration
```

Where `tftp_server` is the IP address of the TFTP server and `dx_configuration` is the filename of the saved configuration on the TFTP server.

6. Install the new firmware by typing:

```
dx% install
```

This will download the .pac file from the TFTP or SCP server specified.

7. You will be prompted with the following warning. Enter **y** to continue.

```
WARNING - this will install firmware to the alternate partition, and will
overwrite its current contents. After the install, you will be able to
select the default boot partition. You will also be given the option to copy
your existing configuration to the new partition.
```

```
Would you like to continue (y/n)? [n]
```

8. After the install completes, set the boot partition for the next reboot by typing:

```
dx% set boot 2
```

Typing the **show boot** command will now show something like the output below. The active partition is the one where the DX will boot.

```
Boot 1 (current) : Juniper Networks Accelerator DX 2.3.3 Wed Mar 12 20:35:50
GMT
Boot 2 (active)  : Juniper Networks Accelerator DX 3.1.0 Fri June 9 20:35:56
GMT
```

9. Reboot the DX:

```
dx% reboot
```

10. Log into the DX using the default username **admin** and the password that you previously defined as the default password, before you installed the new release. If you did not change the default password, it is set to **juniper** by the factory.
11. Import your old configurations and restarting of services. You will be prompted to import your previous configuration settings with the following question:

```
"Would you like to import your existing configuration? [y|n]"
```

Select "yes" to import your previous configuration into the new install. Select "no" if you are not interested in importing your previous configuration at this time.

If you selected "yes", you will next be asked:

```
"Would you like to save your current configuration? [y|n]"
```

Select "yes" to save the configuration to the disk. Select "no" if the configuration shown is not the one you wanted.

Next you will be asked:

```
"Would you like to restart your services? [y|n]"
```

Select “yes” if you would like the services running in your previous configuration to be restarted in the new installation. Select “no” if you do not want any services started at this time. You can enable services later as an administrative user.

12. Save imported settings with the write command:

```
dx% write
```

13. Verify that the software was upgraded to the intended version by typing the command:

```
dx% show version
```

Make sure that the configuration is correct by typing the command:

```
dx% show config
```

14. Optionally, you may set up one or more users to administer the DX. Refer to the procedures in “Multi-Level Administrative Rights” on page 26.

For security reasons, the SSL keypass (pass phrase) is not copied over as part of the configuration file on the new partition after an upgrade. You can import the keypass by typing command:

```
dx% set cluster <n> listen ssl keypass <key password>
```



During an install, the configuration files are copied to the non-active partition. If you reboot to the alternate partition immediately, then the most recent configuration files are used, and no problems should be encountered.

---

However, you may choose to install the software, and then reboot the DX at a later time to limit network impact. If changes are made to the configuration between the install time and the reboot time, the configuration files on the alternate boot partition are no longer current. You must then do another install just before rebooting the unit in order to have updated configuration files on the alternate boot partition.



## Chapter 8

# Integrating the DX Appliance into Your Network

This chapter describes how to integrate your DX Application Acceleration Platform into your network, discussing the following topics:

- Overview on page 152
- Deploying the DX Appliance Behind an External Server Load Balancer (SLB) on page 153
- Integrating the DX Appliance into a Direct Server Return (DSR) Environment on page 154
- Client IP Transparency on page 155
- Source Network Address Translation on page 157
- Floating VIP on page 160
- Connection Binding and Microsoft's NTLM Authentication Protocol on page 161
- Connection Binding and Layer 7 Health Checking on page 162
- Reverse Route Return on page 162
- TCP Selective Acknowledgement on page 164
- Configuring a Virtual LAN on page 164
- Pausing a Target Host on page 167
- Using a Local IP for Target Host Communication on page 169
- Enabling Target Server Compression on page 170
- Instant Redirect on page 173
- “Configuring SNMP” on page 174
- “Configuring Support for the Juniper Secure Access SSL VPN” on page 175

## Overview

---

The last step in fully-integrating the DX Application Acceleration Platform into your network is to direct incoming Web requests to the DX rather than to your Web server.

## Cluster, Redirector, Forwarder, Cache, and ActiveN Group Naming Conventions

---

This feature allows you to name a Cluster, Redirector, Forwarder (“cluster” in its general sense), cache, or ActiveN group to enhance the usability of the DX. A default name will be assigned when a name is not provided. It will be most useful for medium to large customers that have multiple clusters and need easier identification (e.g., meaningful identifier instead of a number) for ease of management. In addition, this feature solves the problem of cluster renumbering when a cluster is deleted.

You can name a cluster, redirector, or forwarder at creation or after it is created. You can also rename an existing cluster, redirector, or forwarder. Names are subject to these restrictions:

- Names can be up to 32 characters long.
- The strings “all,” “cache,” and “NULL” are reserved names and must not be used as a cluster, cache, or ActiveN group names.
- Names are case-sensitive, except for the reserved names “all,” “cache,” and “NULL”. No variations of these words can be used.
- Names can be any valid character string and may be integer-only. The valid characters are:
  - @;,\$^&\*() = +! < > ,[]/\_.-0123456789
  - ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz
- The name cannot contain white space.
- When a cluster, redirector, or forwarder is created without a name specified, a name is automatically created. The name for this unnamed cluster follows the previous behavior as much as possible. Configuration exports from previous releases contain the number of the cluster in the add command, and the remaining cluster configuration commands in the export depend upon the implied identifier of 1,2,3, . . . Using the next available integer as the implied name for a cluster mimics the behavior in previous releases. This way, imports of configurations from previous releases continue to function.
- You can not create a new cluster, redirector, or forwarder if the specified name is already in use. The name space that is considered for name collisions is limited to the type of cluster being added, e.g., when adding a forwarder, the DX appliance will only examine the names of other forwarders for collisions. This allows a cluster, redirector, and forwarder to all share a name of “1.” This is needed for backwards compatibility.

- All references to cluster, redirector, and forwarder use a name instead of a numbered index. The ability to refer to a cluster by index will no longer be supported.

Some examples are:

```
% set cluster <N> ...' becomes '% set cluster <name> ...
% show cluster <N> ...' becomes '% show cluster <name> ...
% delete cluster <N> ...' becomes '% delete cluster <name> ...
% set redirector <N> ...' becomes '% set redirector <name> ...
% show redirector <N> ...' becomes '% show redirector <name> ...
% delete redirector <N> ...' becomes '% delete redirector <name> ...
% set forwarder <N> ...' becomes '% set forwarder <name> ...
% show forwarder <N> ...' becomes '% show forwarder <name> ...
% delete forwarder <N> ...' becomes '% delete forwarder <name> ...
```

- Integer-only names are assigned when no name is specified. The next available lowest integer is used for the assigned names. Example: if you add four clusters without names, the clusters “1”, “2”, “3”, and “4” will be created. If you then delete cluster “2,” the remaining clusters names will not change, leaving clusters “1”, “3”, and “4”. If you then add another cluster without a specified name, the assigned name will be “2” since this is the next lowest available integer. This is referred to as “filling the holes,” and is different from the previous behavior where after deleting cluster 2, the cluster numbers collapsed leaving clusters “1”, “2”, and “3”, and the new cluster's number would then be “4”. This is because all clusters are now referred to by name instead of index.
- The cluster name is included as part of the “add” command on a configuration export.
- The sort order for display of clusters (including tab completion) mimics “sort -n” behavior. This sorts the names according to arithmetic value for any and all leading numeric values in a name. Example: 23www will be listed before 3abc, and 9 will be listed before 11.

As an additional assistance for identification and purpose of clusters, redirectors, and forwarders, a “note” can be applied to individual clusters. This note is limited to 512 characters, and is expected to be free-form text but may not include new lines. This allows administrators to fully describe the cluster's usage, contact information, warnings, or any other pertinent information deemed necessary.

## Deploying the DX Appliance Behind an External Server Load Balancer (SLB)

**NOTE:** The DX has a built-in Server Load Balancer (SLB), and Juniper strongly recommends the use of the internal SLB over an external SLB to improve system performance and reliability. This information is provided for users that are retrofitting the DX behind an existing SLB.

If you use an external Server Load Balancer (SLB), you can use the SLB to direct traffic to the DX rather than the Web servers without interrupting the flow of traffic to your site. The simple deployment means that the DX can be gracefully introduced into and removed from service without interrupting traffic flow. This can be done both manually for maintenance needs, and automatically for hands-off failure recovery.

Follow these steps to integrate the DX into your network without any site downtime:

1. Ensure that the DX is serving pages from the target server. This can be done by accessing the DX through a Web browser and verifying that it is passing back pages from the target server.
2. Add the DX to the list of servers to which the server load balancer is directing traffic. In this configuration, the Web traffic flowing through the DX will be accelerated and the Web traffic flowing directly to the Web server will not be accelerated.
3. Verify that the DX is servicing some of the Web requests by looking at the DX server statistics. This can be done either through DXSHELL with the `show server stats 1` command or by looking at the DX Stats page in the WebUI.
4. Once you are comfortable that the DX is serving pages, re-direct all traffic bound for the Web server(s) to the DX.

## **Integrating the DX Appliance into a Direct Server Return (DSR) Environment**

---

### **Overview**

The DX can be easily deployed behind a Server Load Balancer (SLB) in DSR environments with a minimum amount of configuration. This simple deployment means that the DX can be gracefully introduced into and removed from service without interrupting traffic flow. This can be done both manually for maintenance needs, and automatically for hands-off failure recovery.

### **What is Direct Server Return (DSR)?**

DSR allows Web servers to bypass the load balancer when responding to requests. With DSR, the Web server sends HTTP responses directly to the requesting client, hence the name “Direct Server Return”.

### **Why use DSR?**

Because HTTP responses (i.e., page data, images, etc.) are much greater in size than requests, using DSR greatly reduces traffic flow through the load balancer.

### **How Does DSR Work?**

In a conventional, non-DSR environment, the SLB replaces the destination IP address in each client request packet with the IP address of the optimal target Web server.

With DSR, the load balancer does not modify the destination IP address of client request packets. Instead, the load balancer changes each request packet's destination MAC address to that of the target server. Each target Web server is configured with a loopback IP address that matches the SLB Virtual IP address (VIP). This allows the target host to accept request packets from the SLB and generate response packets that can be sent directly to the client without modification.

Note that only the SLB responds to Address Resolution Protocol (ARP) requests for the VIP to ensure that the router only forwards client requests to the SLB. Also, because only the MAC address is changed, the load balancer and its target servers must reside on the same layer 2 network.

### ***Inserting the DX Appliance into a DSR Environment***

1. Configure a cluster on the DX whose listen VIP matches the SLB VIP by typing the commands:

```
dx% add cluster
dx% set cluster <name> listen vip <VIP of SLB>
dx% set cluster <name> listen port 80
```

2. Add target Web servers that are currently a part of your DSR configuration to the cluster by typing the commands:

```
dx% set cluster <name> target host <ip:port for target host 1>
dx% set cluster <name> target host <ip:port for target host 2>
...
```

3. On the DX, enable DSR for this cluster with the command:

```
dx% set cluster <name> dsr enabled
```

4. Save and apply the changes with the command:

```
dx% write
```

5. Configure the SLB to forward client requests to the DX(s) instead of the pool of Web servers by specifying the actual interface IP address instead of the VIP address.
6. Configure the SLB to use the actual Web servers as a backup pool for the DX unit(s).

If the DX is taken out of service, the SLB will transparently direct traffic to the target Web servers, returning the site to its prior non-accelerated performance level until the SLB brings the DX back into the traffic flow.

**CAUTION:** Target servers should keep their loopback address configuration in order to allow them to handle DSR traffic should the DX be taken out of service.

### **Client IP Transparency**

---

DX Application Acceleration Platforms operate in a secure reverse-proxy mode. In this mode, all incoming client requests are terminated at the DX and multiplexed to a pool of pre-defined target hosts that serve the content. When the DX provides connection multiplexing, the Source IP (SIP) is replaced by the IP of the DX before the request is forwarded to the target host. This is required to provide the connection multiplexing capability within the DX.

However, this may create unintended side-effects:

- The target host logs do not have the client’s IP address any longer.
- Since all requests to the target host seem to originate from a single IP, the host may perceive the traffic as an attack and close the connection.

If you have an application that looks at the client’s IP address, there are two ways around this problem:

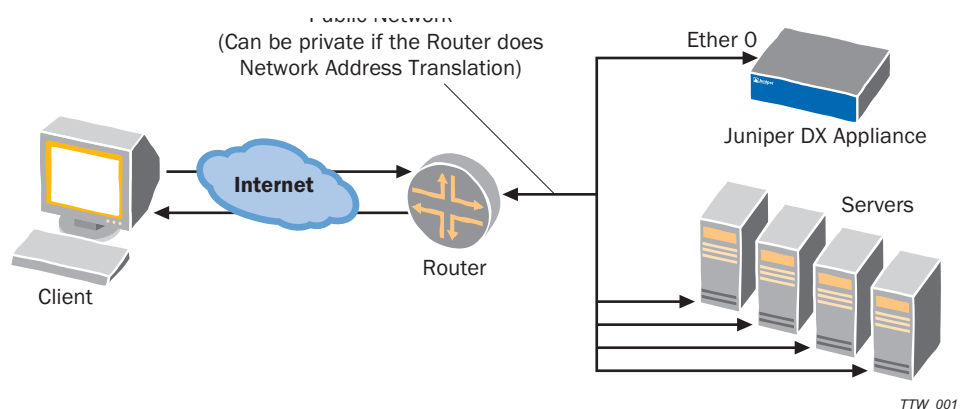
- Change your application to get the client’s IP address from the Juniper “clientipaddr” header instead of the source address.

or

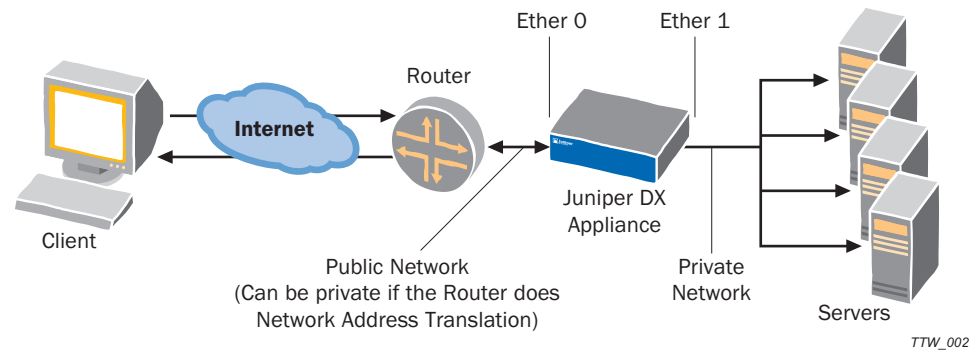
- Use the Client IP Transparency feature.
  - The Client IP Transparency feature allows you to enable or disable the client IP transparency capability for a cluster configuration.
- You will need to consider the following when Client IP Transparency is enabled:
  - The DX will no longer off-load the server (the DX does not pre-establish a session and multiplex client requests in the same persistent session).
- The target servers must use the DX as their default route.

The target hosts must be located on a local subnet directly accessible by the DX, and the clients must come from remote subnets. In the “One-Arm” topology (Figure 49), the DX Ether 0 port and the Web servers must be on one subnet, and the clients must be on other subnets. If there are only a handful of clients, this requirement can be circumvented by using static routes on the server for each client.

**Figure 49: One Arm Topology**



In “In-Line” mode (Figure 50), the DX Ether 1 port and Web servers must be on one subnet, and the DX Ether 0 port and clients must be on other subnets.

**Figure 50: In-Line Topology**

**NOTE:** Client IP Transparency does not support traffic originating from the target hosts and passing through the DX to any remote destination. Contact your Juniper Service Representative if you have any questions or concerns.

### Client IP Transparency Commands

Transparency is disabled by default. This allows the DX to operate in the normal manner. Enabling or disabling IP transparency will take effect only after a `write` operation.

The DXSHELL command used to set Client IP Transparency is:

```
dx% set cluster <name> transparency [enabled | disabled*]
```

The user must be Administrator or Network Administrator to use the `set` command.

The DXSHELL command used to show Client IP Transparency status is:

```
dx% show cluster <name> transparency
```

An Administrator, Network Administrator, Network Operator, or a user may use the `show` command.

### Source Network Address Translation

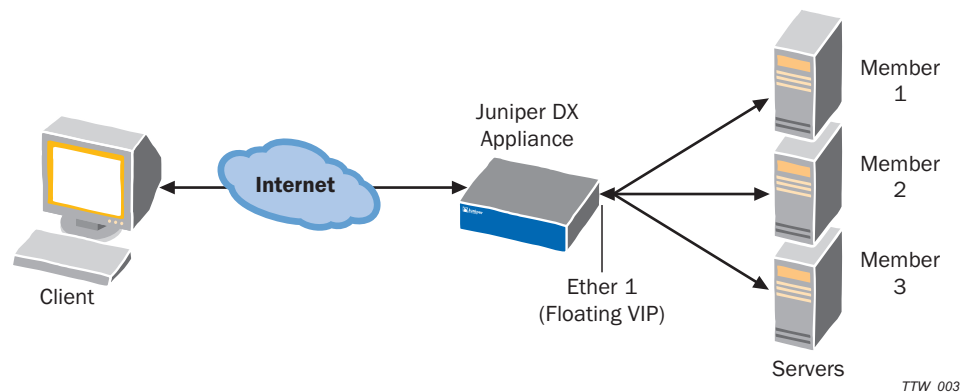
Source Network Address Translation (SNAT) provides external internet access to servers sitting behind a DX. This becomes critical when the server has the DX Application Acceleration Platform IP configured as the default gateway, as in the case of Client IP Transparency. SNAT translates the server's source IP address to the Virtual IP address of either the SLB or a cluster. The SNAT feature is akin to a simple DSL router. The reverse traffic is converted back to its original IP address and sent back to the server. Currently only many-to-one conversion is allowed.

## SNAT Operation

Currently three IP protocols: Transmission Control Protocol (TCP), User Datagram Protocol (UDP), and Internet Control Message Protocol (ICMP) are supported. When a DX receives a packet from a member matching the IP and net mask, it associates it with a session and the source IP and port are replaced with the VIP and a chosen port. The reverse is done on the return traffic from the extranet and connectivity to external network is achieved.

Figure 51 shows the basic operation of SNAT. Servers (called members here) have a floating VIP configured as their default gateway. When a packet matching the member IP and net mask is received on the DX, network translation is applied to the packet by replacing the source IP from the member IP to a VIP and sent via Ether 0, assuming that the default gateway of the DX exists on Ether 0. If that is not the case, then the interface address of the interface facing the default gateway is chosen. Also if the chosen VIP is not aliased on the Ether 0 Interface, then the interface IP address is chosen.

**Figure 51: Basic Operation of SNAT**



## SNAT Configuration Commands

### Add Commands

To add a new SNAT group, type the command:

```
dx% add snat group [name]
```

The name is optional. If a name is not provided, a name starting from 1 will be allocated.

### Delete Commands

To delete a SNAT group, type the command:

```
dx% delete snat group <name>
```



## Set Commands

To add a new member to a group, type the command:

```
dx% set snat group <name> newmember [name]
```

The name is optional. If a name is not provided, a name starting from 1 will be allocated.

To set the VIP for the group, type the command:

```
dx% set snat group <name> vip <IP>
```

To set a group member's IP address, type the command:

```
dx% set snat group <name> member ip <IP>
```

To set the member mask, type the command:

```
dx% set snat group <name> member mask <MASK>
```

To set the maximum number of connections, type the command:

```
dx% set snat maxconn
```

The default is 1000 connections. The minimum is 1 and maximum is 1000.

To set the maximum idle time, type the command:

```
dx% set snat idletime
```

The default is 500 seconds. The minimum is 1 second and maximum is 24 hours.

## Clear Commands

To remove a member from the group, type the command:

```
dx% clear snat group <name> < member | all >
```

## Show Commands

To display all information related to the Source Network Address Translation (SNAT) configuration., type the command:

```
dx% show snat
```

To display the maximum number of connections, type the command:

```
dx% show snat maxconn
```

To display maximum idle time, type the command:

```
dx% show snat idletime
```

To display group information, type the command:

```
dx% show snat group [name | all]
```

To display members in group, type the command:

```
dx% show snat group <name> member < name | all >
```

To display the VIP for a group, type the command:

```
dx% show snat group <name> vip
```

To display the IP address for a member of a group, type the command:

```
dx% show snat group <name> member <name> ip
```

To display the netmask for a member of a group, type the command:

```
dx% show snat group <name> member <name> mask
```

## Floating VIP

---

When using Client IP Transparency, the server sees the IP address of the actual client. To allow the responses to go through the DX, the server must have one of the DX IP addresses configured as its default gateway. This configuration has a problem in that if the server is on a different network than that of the client, the DX is forced to use the IP address of the interface facing the server (Ether 1, for example).

When a failover occurs in this condition, the server still has its default gateway pointed to the old active unit (which is currently down or passive). This causes the traffic to go to the wrong unit. The solution to this problem is to use a “Floating VIP.” A floating VIP is a VIP that floats between two units in failover and always remains on the active unit. ActiveN failover must be configured to use a floating VIP.

A floating VIP is used on the interface facing the servers so that the floating VIP is always aliased in the active unit. This way the server always finds the correct unit as the default gateway. The floating VIP must be based in the same subnet as the server and failover must be enabled.

To add a floating VIP, type the command:

```
dx% add floatingvip <ip>
```

To delete a floating VIP, type the command:

```
dx% delete floatingvip <ip | all>
```

To show all of the floating VIPs, type the command:

```
dx% show floatingvip
```

## Connection Binding and Microsoft's NTLM Authentication Protocol

---

The DX improves application server capacity by multiplexing requests over a few persistent connections to the server farm to conserve the target servers' resources. In some environments, it is necessary to bind a connection from the user to the target server instead of allowing user requests to use an arbitrary connection to the target server. Multiplexing of connections may potentially allow an authenticated connection to be used by non-authorized users, violating the security policy.

Environments that use the NT Lan Manager protocol (NTLM) for authentication to Microsoft Proxy Servers require connection binding. NTLM is a proprietary protocol that authenticates connections rather than users or requests. Therefore, multiplexing connections to the target server must be disabled to avoid violating the NTLM authentication scheme.

### Configuring Connection Binding

The connection binding feature provides the option of binding a connection from a single client to a target server. Connection binding is off by default, and can be enabled on a cluster-by-cluster basis.

1. To enable client to target server connection binding:

```
dx% set cluster <name> connbind enabled
```

In addition, you should configure the following for optimum performance.

2. Enable client IP-based client “stickiness” (refer to “Setting up the DX Appliance for “Sticky” Traffic” on page 251 for additional information).
3. Ensure that the Web server keeps connections alive by setting a long connection time. The suggested value is five minutes or more.
4. To disable the following factory-set server settings:

- a. Disable the addition of an HTTP warning header by typing:

```
dx% set cluster <name> factory h w disabled
```

- b. Disable adding or appending to the HTTP Via header by typing:

```
dx% set cluster <name> factory h v disabled
```

- c. Close the connection to the target server when a 304 response is received by typing:

```
dx% set cluster <name> factory h tc3 disabled
```

## Connection Binding and Layer 7 Health Checking

---

When L7 health checking is enabled and the target servers are NTLM-enabled, the expected HTTP return code of the health check should be set to 401 instead of the default of 200. Because the health check connections from the DX to the target servers are not NTLM authenticated connections, health check requests return 401 “Unauthorized” instead of 200 “OK”. The DX can make sure that the Web server is up and running, but access to content is denied due to the non-authenticated connection.

To set the expected return code to 401, enter the command:

```
dx% set cluster <name> health returncode 401
```

## Reverse Route Return

---

With “Reverse Route Return”, the DX automatically adds routes when packets come back from a node that does not already appear in the DX’s routing table. The problem reverse route return solves is that it is possible to lose packets when there is more than one gateway and the default gateway is not where the packet originated. Reverse route return allows response packets to be sent to the router that originally sent the request packets. This is done automatically, without the user having to manually configure static routes (a very time-consuming, error prone procedure).

For example, assume that there are two routers in the network (R 1 and R 2) and R 1 is the default gateway. If the DX receives a packet from R 2 and there are no routes configured for the particular destination, the response will be routed towards R 1. The information that the request (or original packet) came from R 2 instead of R 1 is not included. Reverse route return remembers the path where the request was originated and enables the DX to send the packet back to the router from which it was received.

### Behavior

Normally, when a packet arrives on aDX Application Acceleration Platform from a node, it is not guaranteed that the response to the packet will go back to the same node. This is because of the way routing works in the operating system. Routing does not “remember” the node from which it received the packet. Instead the routing module decides where the packet should go by using current entries its routing table. In cases where there is no explicit routing entry for the destination, the packet is sent to the default gateway.

If the original packet did not arrive from the default gateway, but instead arrived from a different route, the response may not reach the actual destination. One way to counter this problem is to have the user configure static routes to the destination manually, but this method is not only time-consuming, but also error prone. Also at times it may not be possible to predict the path that a packet will take before arriving at the DX.

The solution to this problem is whenever a packet is received in the system, it is checked for following cases:

- Did the packet originate from another network?
- The incoming packet is from the DX's default gateway.
- There is not a static route configured on the DX for the source IP of the incoming packet.

If all the above conditions apply to the incoming packet, then a route is created to the destination with the next-hop being the node from which the DX appliance received the packet. It is similar to adding a route manually from the command line, except that here the route does not stay indefinitely.

The DX times-out the routes based on the activity. If a route is not used in the number of seconds specified by the `set server reversepath timeout` command, it is removed. This removes stale routes, freeing up memory in the DX. The user can also limit the number of routes that can be added by this method using the `set server reversepath maxroutes` command.

When adding routes, the next-hop is derived from the ARP table and matched against the source hardware address (MAC address) in the packet. There can also be a case where there is not an ARP entry for the MAC address. This is usually the case if the system has recently been rebooted and all the ARP information has been reset. In this case, the DX will change the destination MAC address when it sends the response packets out.

### **Reverse Route Return Commands**

The following commands are provided for configuring the reverse route return feature. This command enables or disables the feature. The default value is disabled.

```
dx% set server reversepath < enabled | disabled >
```

To configure the maximum number of routes that can be added, use the command:

```
dx% set server reversepath maxroutes < number >
```

The minimum number is one, the maximum is 500, and the default value is 20

To configure the maximum timeout value for the entries added, use the command:

```
dx% set server reversepath timeout < secs >
```

The routes will be deleted after this interval of inactivity. The minimum value for timeout is one second, the maximum is 5000 seconds, and the default is 45 seconds.

Any settings made by the `set server reversepath` command will only take effect following the `write` command.

To display the current configuration of the reverse route return feature, use the command:

```
dx% show server reversepath
```

To display the current maximum number of routes that are allowed, use the command:

```
dx% show server reversepath maxroutes
```

To display the current timeout value, use the command:

```
dx% show server reversepath timeout
```

To display the current entries created in the system, use the command:

```
dx% show server reversepath entries
```

To clear an entry created by reverse route return, use the command:

```
dx% clear server reversepath entry < ip >
```

## TCP Selective Acknowledgement

---

Multiple packet losses from a window of data can have a catastrophic effect on TCP throughput. TCP uses a cumulative acknowledgment scheme where received segments that are not at the left edge of the receive window are not acknowledged. This forces the sender to either wait a round-trip time to find out about each lost packet, or to unnecessarily retransmit segments that have been correctly received. With this cumulative acknowledgment scheme, multiple dropped segments generally cause TCP to lose its ACK-based clock, reducing overall throughput.

Selective Acknowledgment (SACK) is a strategy that corrects this behavior in the face of multiple dropped segments. With selective acknowledgments, the data receiver informs the sender about all segments that have arrived successfully, so the sender need only retransmit the segments that have actually been lost.

The DX Application Acceleration Platform supports Selective Acknowledgment, and it is always on; no configuration is needed.

## Configuring a Virtual LAN

---

A Virtual LAN (VLAN) is a network of computers that behave as if they are connected to the same physical network even though they may actually be located on different segments of a LAN. VLANs are configured through software rather than hardware, which makes them extremely flexible. One of the biggest advantages of VLANs is that when a computer is physically moved to another location, it can stay on the same VLAN without any hardware reconfiguration.

The DX uses VLAN tagging to differentiate packets belonging to different VLANs in a multi-VLAN environment. VLAN tags are also useful when two switches with multiple VLANs configured are connected in tandem. This enables the switch to transfer packets safely between the switches and not spill it out of the VLAN.

The mechanics of VLAN tagging are described by IEEE 802.1q Standards. The VLAN tag is a two-byte value inserted between the hardware layer header (an Ethernet header in our case) and the network header (IP or ARP header). A VLAN tag is identified mainly by the 12-bit ID that is part of the VLAN header. It can have a value ranging from 1 to 4095.

The DX inserts VLAN headers in the outgoing packets based on the destination address. It can also insert VLAN tags based on a range of destination addresses.

## Behavior

To understand the behavior of the VLAN feature, you should understand these terms:

- TPID: The Tag Protocol Identifier is set to 0x8100 to identify the frame as a IEEE 802.1q tagged frame.
- PRIO: The Frame Priority field is used to prioritize the traffic.
- CFI: The Canonical Format Indicator is a one-bit flag that indicates that the MAC address is in canonical format.
- VID: The VLAN ID is a two-byte header added in between the Ethernet header and the network layer header. It contains the VLAN tag, which is a value between 1 - 4095. (It is a 12-bit number, where 0 and 4096 are reserved values.)

There are different ways in which the VLAN tagging feature can be implemented. The classical method involves creating virtual interfaces. Each VLAN tag is associated with a virtual interface that has its own subnet and IP address. All the packets that originate from this interface will have the VLAN ID. Each virtual interface is linked with a physical parent interface. For example, this type of system could have a configuration like this:

```
vlan0: flags=8843<UP,BROADCAST,RUNNING,SIMPLEX,MULTICAST> mtu 1500
inet 172.21.12.34 netmask 0xffff0000 broadcast 172.21.255.255
ether 00:02:b3:b2:cb:51
vlan: 10 parent interface: em0
```

In this case, vlan0 is a virtual interface that has an IP address of 172.21.12.34. The VLAN ID associated with this virtual interface is 10 and its parent interface is em0.

This method is very effective if the VLANs are separated at subnet boundaries. The basic guideline of this method say “if you are sending packet from this X interface, the packet will have Y VLAN tag”. While this architecture is well-defined and tested, it is very inflexible. It does not allow you to have two target machines that are in same subnet to be in different VLANs.

To counter this issue, the DX appliance changes the philosophy from “all packets from this interface will have VLAN ID X” to “all the packets going from or to this IP address will have VLAN ID X.” In other words, if the packet contains this IP as either the source or destination IP, then the tag is inserted. In case there is a conflict between the tag for the source IP and the tag for destination IP, the destination IP will take precedence. Subnet boundaries do not affect the VLAN tags.

## VLAN Commands

These commands are used to configure the VLAN feature.

### Set Commands

To add a VLAN tag, type the command:

```
dx% set vlan ip < ip > < tag >
```

or

```
dx% set vlan range < startip-endip > < tag >
```

A tag added with a specific IP address takes precedence over a range. For example, if you add:

```
dx% set vlan range 192.168.10.100-192.168.10.200 10
```

and

```
dx% set vlan ip 192.168.10.34 456
```

the tag will have a VLAN ID of 456 instead of 10, even though IP 192.168.10.34 falls in the specified range.

### Clear Commands

To delete a VLAN entry when added as a single entry, type the command:

```
dx% clear vlan ip <ip | all>
```

For example:

```
dx% clear vlan ip 192.168.56.7
```

To delete a VLAN entry when added as a range, type the command:

```
dx% clear vlan range <startip-endip | all>
```

For example:

```
dx% clear vlan range 192.168.56.4-192.168.56.10
```

Note that this command works only on entries that were added as ranges.

To clear an entry based on the tag value:

```
dx% clear vlan tag <tag>
```

For example:

```
dx% clear vlan tag 10
```



### Show Commands

To display VLAN tags in the system, type the command:

```
dx% show vlan
```

To show VLAN entries based on ranges in the system, type the command:

```
dx% show vlan range <start-end | all>
```

To display VLAN entries based on IP addresses, type the command:

```
dx% show vlan <ip|all>
```

## Pausing a Target Host

---

Target Host Pausing allows you to move target host servers in and out of rotation as live Web servers without forcing a restart of the DX appliance. Target host pausing allows you to place a particular target host into a “soft” or a “hard”-paused condition:

- Soft pausing halts all new client traffic to the target host, but allows all existing “in-use” traffic to continue indefinitely.
- Hard pausing halts all new client traffic to the target host, and terminates all existing in-use traffic.

This action is performed on the command line and takes effect immediately after you press the enter key. If you then issue a **write** command, the paused condition is saved in the configuration. Otherwise, it remains only a runtime change; restarting the server or rebooting the DX appliance will cause that behavioral change to be lost. A paused server can be taken out of the paused state by issuing an **unpause** command.

To avoid potential race conditions that result in undefined and/or aberrant behavior, the pause/unpause commands only take local effect on the in-memory configuration when the server is down. The user is then notified on the command line accordingly. For example:

```
dx% set server down
dx% set cluster 1 target host 192.168.14.223:80 hardpaused
Server could not be contacted. Change applied only to in-memory config.
```

Because the target host pausing works on a per-cluster level, but target hosts are singularly represented within the server, pausing a target host suffers from similar anomalies as Layer 7 health checking. If the same target host is shared across two clusters, then the paused (or unpaused) condition will apply to all clusters that include that target host. This is based upon which condition is present first in the configuration file for any situation where the configuration file is **read** (server up/down, server crash, reboot). An example should clarify this.

**Assumptions**

- Cluster 1 has target host 192.168.10.100:80
- Cluster 2 has target host 192.168.10.100:80
- The server is running.

**Scenario 1**

- Action: Customer pauses the target host 192.168.10.100:80 for Cluster 2.
- Result: Target host 192.168.10.100:80 is paused for both Clusters 1 and 2.

**Scenario 2**

Action: same as Scenario 1, except the operator does a “write”, “server down”, or “server up”.

- Result: Target host 192.168.10.100:80 is paused for both clusters 1 and 2 before **write**. After **server up**, the target host 192.168.10.100:80 is not paused since the paused condition is not configured for cluster 1 and it was the first cluster in the configuration.

If the first cluster in the list has a shared target host set as paused, then all other clusters will have it set as paused when the server is restarted. When a target host is marked as paused for any cluster, it is paused within the server for all clusters, without regard to ordering.

When a cluster has all of its target hosts in some combination of “down” or “paused,” then the cluster is marked down and the Global Application Failover (GAF) functionality comes into play (either “blackhole,” “finclient,” or “redirect”). The target host condition should not be considered a state of the target host, but a behavior; a target host can be both “up” and “paused”, or both “down” and “paused.”

**Target Host Pause Commands**

The commands for target host pausing are as follows.

To enable “hard” pausing, type the command:

```
dx% set cluster N target host X hardpaused
```

To enable “soft” pausing, type the command:

```
dx% set cluster N target host X softpaused
```

To disable pausing (either hard or soft), type the command:

```
dx% set cluster N target host X unpaused
```

## Using a Local IP for Target Host Communication

---

A local IP address can be configured to be used for communication with the target hosts. The configuration is per cluster and forwarder. The same local IP address is used for communication with all the target hosts in a cluster or forwarder. The source IP for all the connections (e.g., client requests, health check requests) initiated by the DX to the target hosts will be this local IP.

By default, the local IP address is not configured. When the local IP address is not configured, the local IP address used is the IP address of the interface through which target host communication takes place.

The following rules apply when using a local IP address:

- Local IPs cannot be one of the interface IPs
- Local IPs cannot be one of the cluster/forwarder VIPs
- Local IPs cannot be the administrator VIP
- Local IPs has to be on the same subnet
- Local IPs will also be used as the source IP for health check connections

When the target hosts in a cluster or forwarder and the local IP are not in the same subnet, the traffic from the DX to the target hosts and target hosts to the DX might follow different routes. It is preferable and the feature is most useful, when they are in the same subnet.

When a target host is in more than one cluster, the local IP used for communication with the target host is the local IP of the first cluster that has the target host and the local IP configured. If no cluster has the local IP configured, the interface IP is used.

When a target host is in more than one forwarder, the local IP used for communication with the target host is the local IP of the first forwarder that has the target host and local IP configured. If no forwarder has the local IP configured, the interface IP is used.

### Local IP Configuration Commands

To set the local IP to be used for communication with all the target hosts in a cluster, type the command:

```
dx% set cluster <name> target localip <ip>
```

To set the local ip to be used for communication with all the target hosts in a forwarder, type the command:

```
dx% set forwarder <name> target localip <ip>
```

To remove the local IP setting for the cluster, type the command:

```
dx% clear cluster <name> target localip
```

To remove the local IP setting for the forwarder, type the command:

```
dx% clear forwarder <name> target localip
```

To display the local IP setting for the cluster, type the command:

```
dx% show cluster <name> target localip
```

To display the local IP setting for the cluster/forwarder, type the command:

```
dx% show forwarder <name> target localip
```

## Enabling Target Server Compression

---

During normal operation, the DX automatically compresses responses sent to Web browser clients, but does not accept compressed responses received from the target servers. With Target Server Compression enabled, the DX can process compressed HTTP responses from a target server, improving the overall performance when:

- The DX is used to perform forward proxy acceleration
- Target servers are located at a large distance from the DX
- Applications running on the target servers utilize specific encodings

The DX receives and processes the response from the target server depending on the configured target server compression mode. The DX can also request an encoding method from the target server based on the configuration of target compression encoding.

### Target Server Compression Mode

Target server compression has three operational modes. In all three modes, the DX requests compressed responses from the target server in a Cluster. The responses from the target server may or may not be compressed. Selecting “None” disables the feature.

In *standard* mode, the DX performs standard processing and Page Translator Content (PTC) application rules are applied, if configured. The encoding may be converted to an alternate method depending on compression settings. Refer to “Configuring OverDrive Application Rules” on page 297 for additional information about application rules.

In *target* mode, the DX forwards the response from the target without performing any processing. No PTC application rules are applied, even if configured, and no encoding modifications are made. This mode is useful when the target server knows which encoding, if any, is best for the client browser.

In *target enhanced* mode, the DX processes only unencoded responses. PTC application rules are applied to unencoded responses. This mode is useful when you want the target server to determine the encoding method, but want the unencoded response to be processed.

See Table 7 for a summary.

**Table 7: DX Behavior Based on Configured Compression Mode**

Compression Mode	Standard Processing	PTC Application Rules Applied	Encoding Modified
None	X		
Standard	X	X	X
Target			
Target Enhanced	unencoded responses only	unencoded responses only	

### Target Compression Encoding

Target compression encoding has two options used to configure the encoding method to request from the target server.

With the *standard* option, the DX requests gzip and deflate compression encoding from the target server. This option only applies when target server compression mode is configured as standard.

With the *browser* option, the DX requests browser-supplied compression encoding from the target server. This option gives the target server exact information about the encodings supported by the client's browser. This is useful when target server compression mode is configured as target or target encoding. The target server can then send the response in the browser-supplied encoding without the DX modifying the response.

Table 8 shows the allowable configuration combinations.

**Table 8: Configuration Combinations**

Encoding Options/Compression Mode	None	Standard	Target	Target Enhanced
Standard		X		
Browser			X	X

### Configuring Target Server Compression with DXSHELL

This section shows the commands used to control target server compression.

#### Set Commands

To set the Target Server Compression mode, type the command:

```
dx% set cluster <name> compression targetcompression mode [none | standard | target | target_en]
```

See Target Server Compression Mode on page 170 for details about the parameters of this command.

To set the Target Server Compression encoding method, type the command:

```
dx% set cluster <name> compression targetcompression encoding
[standard | browser]
```

See Target Compression Encoding on page 171 for details about the parameters of this command.

The **set** commands require the user to have either an Administrator or Network Administrator role to execute them.

### Show Commands

To show the configured target server compression mode, type the command:

```
dx% show cluster <name> compression targetcompression mode
```

To show the configured target server compression encoding method, type the command:

```
dx% show cluster <name> compression targetcompression encoding
```

To show target server compression statistics, use the commands:

```
dx% show cluster <name | all> stats http
dx% show cluster <name> target host <host> stats http
dx% show server stats http
```

To show the historical statistics for target server compression, use the commands:

```
dx% show cluster <name> stats history http target decompression
[performed | failure] [hour | day | month | year]

dx% show cluster <name> target host <host> stats history http
target decompression [performed | failure] [seconds | minutes]

dx% show server stats history http target decompression
[performed | failure] [hour | day | month | year]
```

## Configuring Target Server Compression with the WebUI

Target server compression can be configured using the WebUI:

1. Select the Services tab on the WebUI dashboard.
2. Select the Cluster name for which you want to configure target server compression.
3. Expand the Compression section.
4. Select a target server compression mode from the drop-down list.
5. Optionally, select a target compression encoding from the drop-down list.
6. Collapse the Compression section.
7. Click Save Settings to save your changes.

You can view the number of target server compression requests that are made and the number that have failed for the DX appliance overall and per target server. This information is displayed for the appliance overall on the Statistics tab for the DX Server under Decompression. It is displayed a particular target server on the Statistics tab > Cluster > Select Target Host address > under Decompression.

## Instant Redirect

---

Instant Redirect is a simple mechanism used to divert traffic from a cluster where all target hosts are down (i.e., a “dead” cluster) to an active cluster somewhere else in the network (world). The instant redirect feature allows a user to configure the cluster to respond with a redirect (HTTP 302 reply) instead of operating in the customary blackhole mode. When the cluster is completely down (due to all target hosts being down), new connections arriving at the cluster will have a 302 response sent to them immediately (the DX appliance does not even wait for the request to arrive). The response is made in HTTP 1.0 fashion with the connection being closed after sending out the response. This allows the DX appliance to respond at very high speed and rapidly reflect traffic to the new destination.

The instant redirect feature is configured using the command:

```
dx% set cluster <name> listen targetdown [blackhole|finclient|redirect <url>]
```

where:

- **blackhole** refers to the current behavior of dropping all packets sent to the cluster that has all of its target hosts down.
- **finclient** refers to the historical behavior of allowing the client to connect and then subsequently closing down the connection with a FIN.
- **redirect <url>** refers to the new behavior of redirecting clients with an HTTP 302 reply to the new location specified in the <url>. The URL is specified as follows:

```
http://<server>[:port][/path/resource]
```

To view the current configuration, use the command:

```
dx% show cluster <name> listen targetdown
```



**NOTE:** The Instant Redirect feature only works with HTTP clusters, not HTTPS.

---

## Configuring SNMP

---

### Configuring the SNMP Agent Parameters

The following steps are used to set up the SNMP agent:

1. Enable the SNMP service by typing:

```
dx% set admin snmp up
```

2. Define the System location by typing:

```
dx% set admin snmp location <location>
```

or

```
dx% set admin snmp location snmp q/a lab, rack 4
```

3. Define the System contact by typing:

```
dx% set admin snmp contact <contact>
```

or

```
dx% set admin snmp contact John Smith
```

### Configuring the SNMP Agent for Sending Traps

The following steps are used to set up the SNMP agent to send traps:

1. Define the trap host by typing:

```
dx% set admin snmp trap host [1 | 2] ip <ip address>
```

or

```
dx% set admin snmp trap host 1 ip 205.178.13.100
```

2. Define the community string for the trap host by typing:

```
dx% set admin snmp trap host [1 | 2] community <community string>
```

or

```
dx% set admin snmp trap host 1 community my_community
```

3. Define the SNMP version for the trap host. The agent supports both version 1 and version 2 formats.

```
dx% set admin snmp trap host [1 | 2] version [1 | 2]
```

or

```
dx% set admin snmp trap host 1 version 1
```



4. Enable sending of generic traps by typing:

```
dx% set admin snmp trap generic [enabled | disabled]
```

or

```
dx% set admin snmp trap generic enabled
```

5. Enable sending of Enterprise-specific traps by typing:

```
dx% set admin snmp trap enterprise [enabled | disabled]
```

or

```
dx% set admin snmp trap enterprise enabled
```

6. OPTIONAL. Enable or disable sending of Authentication Failure traps by typing:

```
dx% set admin snmp trap authfailure [enabled | disabled]
```

or

```
dx% set admin snmp trap authfailure enabled
```

7. OPTIONAL. Define the threshold for connections count by typing:

```
dx% set admin snmp trap threshold connection <1-100%>
```

or

```
dx% set admin snmp trap threshold connection 95
```

## Configuring Support for the Juniper Secure Access SSL VPN

---

The DX application acceleration platform supports the Juniper Secure Access SSL VPN (SA) solution. SSL VPNs use SSL/HTTPS to transport private data across the public Internet. Using an SSL VPN, the connection between the mobile user and the internal resource happens through a Web connection at the application layer.

SA provides access to various aspects of the customer network using the following methods:

- Core—access is provided to core functions in the network using SSL and HTTPS connections.
- Secure Application Manager (SAM) and Java Secure Application Manager (JSAM)—access is provided to client/server applications using SSL and HTTPS connections.
- Network Connect (NC)—access is provided to the network at IP layer using Network Connect Protocol (NCP) and optimized NCP.

The default DX cluster configuration provides SSL and HTTP acceleration for the Core and SAM access methods. Additional configuration for NC and JSAM access methods is needed. This configuration places the DX into forwarder mode, and any further data from the client is simply forwarded to the target. This is restricted to the client TCP connection that was forwarded. Any new connections are handled with normal cluster processing.

### **Configuring SA Compatibility**

To configure the DX for compatibility with NC and JSAM SA access methods:

Using the WebUI:

1. Do one of the following:
  - Click the Services tab > Click the Cluster on which to enable SA compatibility >
  - Click the Cluster on which to enable SA compatibility from the Cluster Health area on the Dashboard > Cluster menu
2. Expand the Advanced section.
3. Click the Enabled radio button next to Secure Access Compatibility.
4. Click Save Settings.

Using the CLI, enter the following `set cluster` command:

```
dx% set cluster <name> sacompat enabled
```

### **Viewing the SA Compatibility Configuration**

You can view whether the SA compatibility feature is enabled or disabled using the `show cluster <name> sacompat status` command. To view details of the configuration or specific URL configurations, use the following `show cluster` commands:

```
dx% show cluster <name> sacompat advanced  
dx% show cluster <name> sacompat advanced url [1|2|3]
```

### **Modifying the SA Compatibility Configuration**

You can enable and disable the SA compatibility feature. You can modify the specified URLs, and you can clear the URL values.

To disable the SA compatibility feature, use the `set cluster <name> sacompat disabled` command:

By default the URL used when SA compatibility is enabled is /dana/j. You can modify this URL and optionally specify up to two additional URLs to use for forwarded data.



**NOTE:** In general practice, the default setting for URL1 is sufficient; however the option to modify the default and configure additional URLs is provided for SA requests that require specific treatment.

---

To modify the URL values, enter the appropriate command:

```
dx% set cluster <name> sacompat advanced url 1 <url>
dx% set cluster <name> sacompat advanced url 2 <url>
dx% set cluster <name> sacompat advanced url 3 <url>
```

The second and third URLs are empty by default.

To restore the URLs to their default values, use the **set cluster <name> sacompat advanced defaults** command.

To clear the URL configuration, use the **clear cluster <name> sacompat advanced url <1|2|3>** command.



## Chapter 9

# Configuring Server Load Balancing

This chapter describes how to configure the Server Load Balancer service on the DX Application Acceleration Platform. Accepting all default behaviors, you can configure the DX platform to perform basic server load balancing functions. You can customize the behavior of the SLB service to improve the performance of the SLB and to support a more sophisticated network configuration.

This chapter includes the following topics:

- “Configuring a Basic Server Load Balancer” on page 179
- “Customizing the SLB Service” on page 180
- “Pausing a Target Host” on page 184
- “Deleting an SLB Group” on page 185
- “Statistics” on page 185

For more detail about all of the commands available, see the *CLI Reference Guide for DXOS*.

## Configuring a Basic Server Load Balancer

---

To configure the SLB service on the DX platform for basic operation, you need only create an SLB group and add a target host (for example an application server) to the group.

To configure an SLB using DXSHELL:

1. Add an SLB group:

```
dx% add slb group [name] <ip:port>
```

The name of the group is optional. A name is generated for the SLB group if you do not enter one yourself. The IP address and port are the virtual IP (VIP) on which the SLB group listens to (receives) incoming traffic from an application or protocol that is being load balanced.

2. Add one or more target hosts to the SLB group using their IP addresses:

```
dx% set slb group <name> target host <ip:port>
```

A target host is a single server that is part of a group.

3. Enable the SLB:

```
dx% set slb enabled
```

4. Save your configuration changes:

```
dx% write
```

The DX platform is now configured with SLB service. You may configure another SLB service (repeat this procedure) or customize the configuration for better performance (see the next section, “Customizing the SLB Service” on page 180).

## Customizing the SLB Service

---

You can improve the overall performance of the DX platform acting as a SLB by tuning the SLB service to better meet your network configuration and needs. The following sections provide configuration information for customizing your SLB services:

- “Configuring Network Address Translation (NAT)” on page 180
- “Configuring the SLB Group Communication Protocol” on page 181
- “Configuring the SLB Group Load-Balancing Policy” on page 181
- “Configuring SLB Health Checking” on page 181
- “Configuring Client to Server Sticky” on page 182
- “Configuring Quality of Service (QoS)” on page 183
- “Configuring SLB Session Parameters” on page 183

For information about how to configure failover for the SLB service, see “Configuring Failover” on page 357.

### Configuring Network Address Translation (NAT)

NAT is configured at the SLB group level. Full- and Half-NAT options are available:

- With Full-NAT, address translation is performed on both the source address and the destination address. The interface IP address is used as the source IP address to connect to the target host. There are no modifications required to the target hosts. This is the default setting.
- With Half-NAT, address translation is only performed on the destination address. In this configuration, the target host must have its default route pointing to the SLB.

To configure NAT for an SLB group using DXSHELL:

1. Specify full- or half-NAT:

```
dx% set slb group <name> nat <half|full*>
dx% write
```

2. Optionally, when full-NAT is configured, specify the range of ports on which the address translation is to occur. Port values are specified between zero and 65535. The default start port is 1024. The default end port is 8000.

```
dx% set slb group <name> nat port start <number>
dx% set slb group <name> nat port end <number>
dx% write
```

### **Configuring the SLB Group Communication Protocol**

The communication protocol is configured at the SLB group level. TCP and UDP options are available. TCP is the default option.

To configure the SLB group protocol using DXSHELL, enter the following command:

```
dx% set slb group <name> protocol <udp|tcp*>
dx% write
```

### **Configuring the SLB Group Load-Balancing Policy**

By default an SLB group uses the round robin load-balancing policy. You can modify this setting for all of your SLB groups or for a particular SLB group.

To modify this setting using DXSHELL, enter the following command:

```
dx% set slb group <name|all> policy <roundrobin*|leastconns|
backupchainrevert|weightedroundrobin|maxconns|
weightedleastconns>
dx% write
```

Refer to “Load Balancing Policies” on page 30 for descriptions of these load balancing options.

### **Configuring SLB Health Checking**

Health checking in the form of a valid connection check is performed automatically for all target hosts in an SLB group. This feature cannot be disabled, but you can override the global settings for how often a target host is checked and the number of times it attempts to connect to that target host. Optionally, you can configure SMTP health checking.

To fine tune health checking for SLB globally or a specific SLB group using DXSHELL:

1. Specify the interval for health check requests when the target host is in various administrative states:

```
dx% set slb healthcheck interval <down <seconds> | syn <seconds>|  
up <seconds>>
```

```
dx% set slb group <name> healthcheck interval <down <seconds> |  
syn <seconds>|up <seconds>>
```

The default when a target host is down is 10 seconds. When the target host is up, the default is 20 seconds. For TCP SYN , the default is 5 seconds.

2. Specify the maximum number of health checks to perform. The default is three tries.

```
dx% set slb healthcheck maxtries <number>
```

```
dx% set slb group <name> healthcheck maxtries <number>
```

3. Enable SMTP health checking for one or all SLB groups. This is disabled by default.

```
dx% set slb group <name|all> healthcheck smtp enabled
```

4. Save your configuration changes.

```
dx% write
```

## Configuring Client to Server Sticky

The “sticky” feature is configured on an SLB group level. When the sticky feature is enabled, all requests received by the DX appliance from a particular client are directed to a specified target host. This feature is disabled by default.

To configure client sticky for an SLB group using DXSHELL:

1. Enable the sticky feature:

```
dx% set slb group <name> sticky enabled
```

2. Optionally, specify the maximum number of minutes (1 to 43200) allowed between consecutive client requests that are bound to the same target host, for one or all groups:

```
dx% set slb sticky timeout <minutes>  
dx% set slb group <name | all> sticky timeout <minutes>
```

The default is 120 minutes. This overrides the SLB global timeout.



- Optionally, specify whether a client IP is bound to (follows) the same target host as another cluster, forwarder, or SLB group (the default is none). This creates a sticky group that can load balance client applications with multiple protocol flows (such as TCP and UDP) to the same target host.

```
dx% set slb group <name> sticky leader <none*|<cluster|forwarder|
slb group> <name>>
```

- Save your configuration changes.

```
dx% write
```

### Configuring Quality of Service (QoS)

The DX platform can attach QoS information to the traffic sent to the target hosts and/or the return traffic sent to the client. In particular, you can configure the Differentiated Services Code Point/Type of Service (DSCP/ToS) values on this traffic. The QoS information is configured on an SLB group level, and is not attached by default.

To configure QoS for an SLB group using DXSHELL, mark the traffic flow with the QoS parameters you want applied:

```
dx% set slb group <name> <listen|target> qos mark outgoing
[dscp|tos|none*]
```

For example, if you want DSCP and ToS values attached to the traffic returning to the client, enter the following:

```
dx% set slb group <name> listen qos mark outgoing dscp
dx% set slb group <name> listen qos mark outgoing tos
dx% write
```

If you want to attach QoS information to the traffic sent to the target hosts in your SLB group as well, enter the following:

```
dx% set slb group <name> target qos mark outgoing dscp
dx% set slb group <name> target qos mark outgoing tos
dx% write
```

### Configuring SLB Session Parameters

You can specify globally, or by SLB group, the amount of time the DX appliance waits to close an SLB session.

Three purge criteria can be used to end a session:

- **ackwait**: Three way TCP handshake has not completed within specified time (default is 6 seconds).
- **active**: No active sessions are present within the specified time (default is 90 seconds).
- **closewait**: Sessions are terminated by the client (default is 12 seconds).

To set the global purge timeouts for SLB sessions, enter:

```
dx% set slb session timeout <ackwait | active | closewait>
    <seconds>
dx% write
```

To set the SLB group purge timeouts for SLB sessions, enter:

```
dx% set slb group <name> session timeout <ackwait | active |
    closewait> <seconds>
dx% write
```

You may also send a reset to either the client or server when an active session is purged to indicate that the connection has been terminated. This can be configured for all SLB services on the DX appliance or for a particular SLB group, as follows:

```
dx% set slb advanced reset <client | server> <disabled | enabled>
dx% write

dx% set slb group <name> advanced reset <client | server>
    <disabled | enabled>
dx% write
```

## Pausing a Target Host

---

Target Host Pausing allows you to move target host servers in and out of rotation without forcing a restart of the DX appliance. Target host pausing allows you to place a particular target host into a “soft” or a “hard”-paused condition:

- Soft pausing halts all new client traffic to the target host, but allows all existing “in-use” traffic to continue indefinitely.
- Hard pausing halts all new client traffic to the target host, and terminates all existing in-use traffic.

This action is performed on the command line and takes effect immediately after you press the enter key. If you then issue a **write** command, the paused condition is saved in the configuration. Otherwise, it remains only a runtime change; restarting the server or rebooting the DX appliance will cause that behavioral change to be lost. A paused server can be taken out of the paused state by issuing an **unpause** command.

To hard pause a selected target host or all target hosts within an SLB group:

```
dx% set slb group <name|all> hardpaused
```

To soft pause a selected target host or all target hosts within an SLB group:

```
dx% set slb group <name|all> softpaused
```

To disable pausing (either hard or soft) for one or all target hosts within an SLB group:

```
dx% set slb group <name|all> unpaused
```

## Deleting an SLB Group

---

You can delete SLB groups using the DXSHELL. Using the keyword `all` deletes all of the SLB groups. When deleting an SLB group, all the target hosts that were added under the group are also deleted.

To delete an SLB group:

```
dx% delete slb group <name|all>
```

You can also delete one or more of the Target Hosts assigned to an SLB Group. To do this, use the following command:

```
dx% clear slb group < name > target host <targetip|all>
```

## Statistics

---

These commands are used to view statistics for the Server Load Balancer. They provide a way to monitor the performance of an SLB and to address any issues found. The SLB statistics can be viewed at an overall level, group level, or target host level.

### Overall Statistics

You can view basic SLB statistics or a number of other categories of SLB statistics using the following commands:

```
dx% show slb stats  
dx% show slb stats errors  
dx% show slb stats healthcheck  
dx% show slb stats memory  
dx% show slb stats sticky  
dx% show slb stats ftp  
dx% show slb stats tftp
```

Except for memory and current session statistics, the statistics displayed are cumulative.

Table 5 lists the statistics displayed for each command.

**Table 5: SLB Statistics Displayed with show slb stats Commands**

Command	Statistic Displayed	Description
show slb stats	Client bytes	Number of bytes received from all clients using all SLB services.
	Client packets	Number of packets received from all clients using all SLB services.
	Server bytes	Number of bytes received from all target hosts provided in response to all clients using SLB services.
	Server packets	Number of packets received from all target hosts provided in response to all clients using all SLB services.
	Current session: Active Embryonic Finwait	Number of active, embryonic, and finwait sessions using all SLB services.
	Total connections	Total number of connections attempted by the SLB services.
	Successful connections	Total number of connection attempts made by SLB services that were successful.
	Failed connections	Number of connection attempts made by SLB services that failed.
	All of the following statistics	
show slb stats errors	NAT port unavailable	Number of times the port used for NAT is unavailable.
	Target host unavailable	Number of times target hosts have been unavailable.
	Target host lost	Number of times an established connection was lost to any target host.
	Connection timeouts: Active Embryonic Finwait	Number of times active, embryonic, and finwait connections did not connect within the necessary time.
	Connection established failed	Number of times a connection could not be established.
	Resets from clients	Number of times a client reset a connection to the DX.
	Resets from target hosts	Number of times a target host reset a connection to the DX.
	Transmission errors	Number of errors due to transmission problems, such as NIC failures, buffer overload, and so forth.

**Table 5: SLB Statistics Displayed with show slb stats Commands (continued)**

Command	Statistic Displayed	Description
show slb stats healthcheck	Total probes	Total number of times health checking has been performed.
	Total responses	Total number of responses received from health check probes.
	Total timeouts	Total number of times a health check probe did not receive a response in the designated amount of time.
	Success since change	Number of times health check probes have been successful since the last change of state (from down to up or up to down).
	Failures since last change	Number of times health check probes have failed since the last change of state (from down to up or up to down).
	Transitions: Down to up Up to down	Number of times all SLB services have been brought up or down.
	Time since last change	Amount of time, in hh:mm:ss format, since the last status change has occurred.
show slb stats memory	Current usage: NAT memory NAT ports Session memory	Amount of memory used by the NAT feature, the NAT ports, and all current SLB sessions.
	Memory allocation failed	Number of times memory could not be allocated for an SLB service.
show slb stats sticky	Inserts	Number of entries added to the Client IP sticky table.
	Retrieves	Number of entries retrieved from the Client IP sticky table.
	Paused target hosts	Number of times attempted to retrieve entries from target hosts that were in the paused state.
	Down target hosts	Number of times attempted to retrieve entries from target hosts that were down.
show slb stats ftp	Commands: PORT PASV EPSV EPRT	Number of PORT, PASV, EPSV, and EPRT commands received through FTP requests. For information about these codes, refer to RFC 959.
	Responses PASV EPSV	Number of PASV and EPSV responses sent. For information about these codes, refer to RFC 959.
show slb stats tftp	Write request (WRQ)	Number of WRQ requests received in TFTP requests.
	Read request (RRQ)	Number of RRQ requests received in TFTP requests.

### Corresponding show slb stats Commands

For each show command listed, a corresponding `clear slb stats` command is available:

```
dx% clear slb stats
dx% clear slb stats errors
dx% clear slb healthcheck
dx% clear slb stats memory
dx% clear slb stats sticky
dx% clear slb stats ftp
dx% clear slb stats tftp
```

Executing these commands clears the counters for all the statistics in the respective category.

### Group Statistics

SLB service statistics can be displayed by SLB group. The same statistics are shown for each category of statistics as are displayed with the overall SLB statistics command. Corresponding `clear slb group stats` commands are available.

To view the group statistics using the following commands:

```
dx% show slb group <name> stats
dx% show slb group <name> stats errors
dx% show slb group <name> stats healthcheck
dx% show slb group <name> stats memory
dx% show slb group <name> stats sticky
dx% show slb group <name> stats ftp
dx% show slb group <name> stats tftp
```

Except for memory and current session statistics, the statistics displayed are cumulative.

### Target Host Statistics

SLB service statistics can be displayed by target host within an SLB group. The same statistics are shown for each category of statistics as are displayed with the overall SLB statistics command. Corresponding `clear slb group target host stats` commands are available.

To view the group statistics using the following commands:

```
dx% show slb group <name> target host <ip:port> stats
dx% show slb group <name> target host <ip:port> stats errors
dx% show slb group <name> target host <ip:port> stats healthcheck
dx% show slb group <name> target host <ip:port> stats memory
dx% show slb group <name> target host <ip:port> stats sticky
dx% show slb group <name> target host <ip:port> stats ftp
dx% show slb group <name> target host <ip:port> stats tftp
```

Except for memory and current session statistics, the statistics displayed are cumulative.

## Chapter 10

# Setting Up the DX Appliance for SSL Traffic

This chapter describes setting up the DX for Secure Socket Layer (SSL) traffic, discussing the following topics:

- Before You Begin on page 190
- Step-by-step Configuration Examples on page 190
- Importing Existing Keys and Certificates on page 197
- Importing from iPlanet on page 205
- Generating Keys and Certificates on page 206
- SSL Ciphersuite Details on page 209
- Forcing Clients to use HTTPS with Cluster Redirection (Auto SSL) on page 212
- Configuring SSL Client Authentication on page 213
- Specifying Your Own List of SSL Ciphersuites on page 210

## Before You Begin

---

If you are installing the DX in a testing environment where valid key and certificate files are not needed, the DX comes with “dummy” key and certificate files named `demokey` and `democert`, respectively.

If you are installing the DX in a production environment, make sure you have valid key and certificate files in base-64 encoded format. Instructions for importing these files from a variety of environments, as well as converting them to base-64, appear in “Importing Existing Keys and Certificates” on page 197.

When importing key files from different environments, occasionally they will need to be converted using the OpenSSL software. For information on this program, see the OpenSSL Web pages at:

<http://www.openssl.org/>

## Step-by-step Configuration Examples

---

**NOTE:** An (\*) before the command prompt indicates that the configuration has been changed but not written (SAVED).

### Possible SSL Cluster Configurations with the DX Appliance

There are four possible SSL Cluster Configurations. Each is discussed in an example that follows.

LISTEN: SSL Disabled	LISTEN: SSL Enabled
TARGET: SSL Enabled	TARGET: SSL Disabled
LISTEN: SSL Enabled	LISTEN: SSL Disabled
TARGET: SSL Enabled	TARGET: SSL Disabled

### SSL Configuration Examples: Listen: Enabled and Target: Disabled

These instructions guide you through the process of setting up a DX with SSL “enabled” on the listen side and “disabled” on the target side. This section assumes you have already captured your key and certificate files. Refer to the line-by-line explanations of these commands in “Basic Conventions and Terms” on page 34.

1. Set the listen configuration:

```
dx% set cluster 1 listen port 443
(*) dx% set cluster 1 listen vip 10.100.2.63
(*) dx% write
dx% set cluster 1 listen ssl protocol sslv23
(*) dx% set cluster 1 listen ssl certfile cert
(*) dx% set cluster 1 listen ssl keyfile key
New password:
(*) dx% set cluster 1 listen ssl ciphersuite all
(*) dx% set cluster 1 listen ssl enabled
(*) dx% write
```



- Optionally, if the key in the key file is encrypted with a password, set the password so that the DX appliance can decrypt the key:

```
(*) dx% set cluster 1 listen ssl keypass
(*) dx% write
```

- Set the target configuration:

```
dx% set cluster 1 target name mywebserver.juniper.net
(*) dx% clear cluster 1 target host all
(*) dx% set cluster 1 target host 10.100.1.37:80
(*) dx% set cluster 1 target ssl disabled
(*) dx% write
```

- Start the server:

```
dx% set server up
(*) dx% write
```

- Enable the convert302 protocol option.

With the convert302 protocol option enabled, the DX converts the HTTP 302 responses from the target server from HTTP to HTTPS for the client.

```
dx% set cluster <name> convert302protocol enabled
```



**NOTE:** If you need to redirect requests from a secure server back to the non secure server, you should not enable this option.

---

You should now have SSL on the listen side and clear on the target side. Try opening a browser and going to <https://10.100.2.63/> to test the configuration.

### **SSL Configuration Examples: Listen: Disabled and Target: Enabled**

These instructions guide you through the process of setting up a DX with SSL “disabled” on the listen side and “enabled” on the target side. This section assumes you have already captured your key and certificate files.

- Set the listen configuration:

```
dx% set cluster 1 listen port 80
(*) dx% set cluster 1 listen vip 10.100.2.63
(*) dx% set cluster 1 listen ssl disabled
(*) dx% write
```

- Set the target configuration:

```
dx% set cluster 1 target name mywebserver.juniper.net
(*) dx% clear cluster 1 target host all
(*) dx% set cluster 1 target host 10.100.1.37:80
(*) dx% write
dx% set cluster 1 target ssl protocol sslv23
(*) dx% set cluster 1 target ssl ciphersuite all
(*) dx% set cluster 1 target ssl timeout 1440
(*) dx% set cluster 1 target ssl enabled
(*) dx% set server factory cscf enabled
(*) dx% write
```

3. OPTIONAL: If the Web server certificates are invalid and being used for testing:

```
dx% set server factory cscf disabled
(*) dx% write
```

4. Start the server:

```
dx% set server up
(*) dx% write
```

You should now have clear on the listen side and SSL on the target side. Open a browser and go to <http://10.100.2.63/> to test the configuration.

### **SSL Configuration Examples: Listen: Enabled and Target: Enabled**

These instructions guide you through the process of setting up a DX with SSL “enabled” on the listen side and “enabled” on the target side. This section assumes you have already captured your key and certificate files.

1. Set the listen configuration:

```
dx% set cluster 1 listen port 443
(*) dx% set cluster 1 listen vip 10.100.2.63
(*) dx% write
dx% set cluster 1 listen ssl protocol sslv23
(*) dx% set cluster 1 listen ssl certfile txcert
(*) dx% set cluster 1 listen ssl keyfile txkey
New password:
(*) dx% set cluster 1 listen ssl ciphersuite all
(*) dx% set cluster 1 listen ssl enabled
(*) dx% write
```

2. Optionally, if the key in the key file is encrypted with a password, set the password so that the DX appliance can decrypt the key:

```
(*) dx% set cluster 1 listen ssl keypass
(*) dx% write
```

3. Set the target configuration:

```
dx% set cluster 1 target name mywebserver.juniper.net
(*) dx% clear cluster 1 target host all
(*) dx% set cluster 1 target host 10.100.1.37:80
(*) dx% write
dx% set cluster 1 target ssl protocol sslv23
(*) dx% set cluster 1 target ssl ciphersuite all
(*) dx% set cluster 1 target ssl timeout 1440
(*) dx% set cluster 1 target ssl enabled
(*) dx% set server factory cscf enabled
(*) dx% write
```

4. OPTIONAL: If the Web server certificates are invalid and being used for testing.

```
dx% set server factory cscf disabled
(*) dx% write
```

5. Start the server:

```
dx% set server up
(*) dx% write
```

You should now have end-to-end SSL. Open a browser and go to <http://10.100.2.63/> to test the configuration.

### **SSL Configuration Examples: Listen: Disabled and Target: Disabled**

These instructions guide you through the process of setting up a DX with SSL “disabled” on the listen side and “disabled” on the target side. This section assumes you have already captured your key and certificate files.

1. Set the listen configuration:

```
dx% set cluster 1 listen port 80
(*) dx% set cluster 1 listen vip 10.100.2.63
(*) dx% set cluster 1 listen ssl disabled
(*) dx% write
```

2. Set the target configuration:

```
dx% set cluster 1 target name mywebserver.juniper.net
(*) dx% clear cluster 1 target host all
(*) dx% set cluster 1 target host 10.100.1.37:80
(*) dx% set cluster 1 target ssl disabled
(*) dx% write
```

3. Start the server:

```
dx% set server up
(*) dx% write
```

You should now have end-to-end clear. Try opening a browser and going to <http://10.100.2.63/>

### **SSL Forwarder Configuration**

The DX can be configured to act as an SSL Forwarder. In Forwarder mode, the DX performs the SSL encryption or decryption, and then forwards the HTTP or non-HTTP traffic directly to the server or client. In Forwarder mode, the client connection gets terminated at the DX, and the DX opens a new connection to the server. The DX then forwards HTTP and non-HTTP traffic transparently from the client to the server. This means that the DX never initiates termination of a connection; it is either the client or the server.

An SSL Forwarder offers these features:

- Forwards HTTP and non-HTTP traffic transparently from client to server
- Forwarder can be used for:
  - Offloading SSL on the client side for HTTP and non-HTTP traffic
  - Server side SSL for HTTP and non-HTTP traffic
  - End-to-end SSL
- Performs Layer 4 health checking for monitoring target hosts

- Provides I/O and SSL statistics (same as a cluster)
- Acts like a cluster with connection-binding “On” and pre-established (“Hot”) target connections equal ‘Zero’, and no HTTP handling.
- Honors all global factory settings applicable to I/O and SSL layers
- Supports DSR mode

### **Possible SSL Forwarder Configurations with the DX Appliance**

There are four possible SSL Forwarder Configurations. Each is discussed in an example that follows:

LISTEN: SSL Disabled	LISTEN: SSL Enabled
TARGET: SSL Enabled	TARGET: SSL Disabled
LISTEN: SSL Enabled	LISTEN: SSL Disabled
TARGET: SSL Enabled	TARGET: SSL Disabled

### **SSL Configuration Examples: Listen: Enabled and Target: Disabled**

These instructions will guide you through the process of setting up a DX as a Forwarder with SSL “enabled” on the listen side and “disabled” on the target side. This section assumes you have already captured your key and certificate files. Refer to the line-by-line explanations of these commands in “Basic Conventions and Terms” on page 34.

1. Set the listen configuration:

```
dx% set forwarder 1 listen port 443
(*) dx% set forwarder 1 listen vip 10.100.2.63
(*) dx% write
dx% set forwarder 1 listen ssl protocol sslv23
(*) dx% set forwarder 1 listen ssl certfile cert
(*) dx% set forwarder 1 listen ssl keyfile key
(*) dx% set forwarder 1 listen ssl keypass
New password:
(*) dx% set forwarder 1 listen ssl ciphersuite all
(*) dx% set forwarder 1 listen ssl enabled
(*) dx% write
```

2. Set the target configuration:

```
dx% set forwarder 1 target name mywebserver.juniper.net
(*) dx% clear forwarder 1 target host all
(*) dx% set forwarder 1 target host 10.100.1.37:80
(*) dx% set forwarder 1 target ssl disabled
(*) dx% write
```

3. Start the server:

```
dx% set server up
(*) dx% write
```

4. Enable the convert302 protocol option.

With the convert302protocol option enabled, the DX will convert the HTTP 302 responses from the target server from HTTP to HTTPS for the client.

```
dx% set forwarder <name> convert302protocol enabled
```

**NOTE:** If you need to redirect requests from a secure server back to the non-secure server, you should not enable this option.

You should now have SSL on the Listen side and clear on the Target side. Try opening a browser and going to <https://10.100.2.63/> to test the configuration.

### **SSL Configuration Examples: Listen: Disabled and Target: Enabled**

These instructions will guide you through the process of setting up a DX as a Forwarder with SSL “disabled” on the listen side and “enabled” on the target side. This section assumes you have already captured your key and certificate files.

1. Set the listen configuration:

```
dx% set forwarder 1 listen port 80
(*) dx% set forwarder 1 listen vip 10.100.2.63
(*) dx% set forwarder 1 listen ssl disabled
(*) dx% write
```

2. Set the target configuration:

```
dx% set forwarder 1 target name mywebserver.juniper.net
(*) dx% clear forwarder 1 target host all
(*) dx% set forwarder 1 target host 10.100.1.37:80
(*) dx% write
dx% set forwarder 1 target ssl protocol sslv23
(*) dx% set forwarder 1 target ssl ciphersuite all
(*) dx% set forwarder 1 target ssl timeout 1440
(*) dx% set forwarder 1 target ssl enabled
(*) dx% set server factory cscf enabled
(*) dx% write
```

3. OPTIONAL: If the Web server certificates are invalid and being used for testing:

```
dx% set server factory cscf disabled
(*) dx% write
```

4. Start the server:

```
dx% set server up
(*) dx% write
```

You should now have SSL clear on the Listen side and enabled on the Target side. Open a browser and go to <https://10.100.2.63/> to test the configuration.

### **SSL Configuration Example, Listen: Enabled, Target: Enabled**

These instructions will guide you through the process of setting up a DX as a Forwarder with SSL “enabled” on the listen side and “enabled” on the target side. This section assumes you have already captured your key and certificate files.

1. Set the listen configuration:

```
dx% set forwarder 1 listen port 443
(*) dx% set forwarder 1 listen vip 10.100.2.63
(*) dx% write
dx% set forwarder 1 listen ssl protocol sslv23
(*) dx% set forwarder 1 listen ssl certfile txcert
(*) dx% set forwarder 1 listen ssl keyfile txkey
(*) dx% set forwarder 1 listen ssl keypass
New password:
(*) dx% set forwarder 1 listen ssl ciphersuite all
(*) dx% set forwarder 1 listen ssl enabled
(*) dx% write
```

2. Set the target configuration:

```
dx% set forwarder 1 target name mywebserver.juniper.net
(*) dx% clear forwarder 1 target host all
(*) dx% set forwarder 1 target host 10.100.1.37:80
(*) dx% write
dx% set forwarder 1 target ssl protocol sslv23
(*) dx% set forwarder 1 target ssl ciphersuite all
(*) dx% set forwarder 1 target ssl timeout 1440
(*) dx% set forwarder 1 target ssl enabled
(*) dx% set server factory cscf enabled
(*) dx% write
```

3. OPTIONAL: If the Web server certificates are invalid and being used for testing:

```
dx% set server factory cscf disabled
(*) dx% write
```

4. Start the server

```
dx% set server up
(*) dx% write
```

You should now have end-to-end SSL. Open a browser and go to <https://10.100.2.63/> to test the configuration.

### SSL Configuration Example, Listen: Disabled, Target: Disabled

These instructions will guide you through the process of setting up a DX as a Forwarder with SSL “disabled” on the listen side and “disabled” on the target side. This section assumes you have already captured your key and certificate files.

1. Set the listen configuration:

```
dx% set forwarder 1 listen port 80
(*) dx% set forwarder 1 listen vip 10.100.2.63
(*) dx% set forwarder 1 listen ssl disabled
(*) dx% write
```

2. Set the target configuration:

```
dx% set forwarder 1 target name mywebserver.juniper.net
(*) dx% clear forwarder 1 target host all
(*) dx% set forwarder 1 target host 10.100.1.37:80
(*) dx% set forwarder 1 target ssl disabled
(*) dx% write
```

3. Start the server:

```
dx% set server up
(*) dx% write
```

You should now have end-to-end clear. Try opening a browser and going to <http://10.100.2.63/>.

## Importing Existing Keys and Certificates

---

If you already have certificates and keys, you can transfer them to the DX. This section shows how to import keys and certificates from:

- Apache mod\_ssl
- ApacheSSL
- IIS 4.0
- IIS 5.0
- iPlanet

Key and certificate file names cannot contain spaces, and must be compatible with the server operating system. When prompted either to name a key or certificate file or check the name of a key or certificate file, ensure that the names follow these conventions. Keys and certificates must be base-64 encoding to work with the DX.

**NOTE:** If you are using a global certificate, you will need to install a chain certificate (Intermediate Certificate) so that browsers can trust your certificate. Your Trusted Root Certificate Authority can provide this intermediate certificate.





```
.
dx% list file
txcert
txkey
dx%
```

The DX now has a certificate and key with which to perform SSL transactions.

## Importing from ApacheSSL

The key and certificate locations are listed in the \$PACHESLROOT/conf/httpd.conf file. The default key is \$PACHEROOT/certs/\*.key. The default certificate is \$PACHEROOT/certs/\*.crt. Make note of these names and locations.

To import these files to the DX, follow this example of copying and pasting the key and certificate files from the locations previously described.

```
dx% capture file txcert
Enter file. End with . on a blank line.
-----BEGIN CERTIFICATE-----
MIIDejCCAuOgAwIBAgIBADANBgkqhkiG9w0BAQQFADCBizELMAkGA1UEBhMCWFgx
EjAQBgNVBAGTCURFTU8gT05MWTESMBAGA1UEBxMJREVNTyBPTkxZMREwEAYDVQK
Ew1ERU1PIE90TFkxYjAQBgNVBAsTCURFTU8gT05MWTESMBAGA1UEAxMjREVNTyB
TkxZMRgwFgYJKoZIhvcNAQkBFglERU1PIE90TFkxHhNMDIwMzA1MjMzAxwhcN
MDIwMzA2MjMzAxwCBizELMAkGA1UEBhMCWFgxYjAQBgNVBAGTCURFTU8gT05M
WTESMBAGA1UEBxMJREVNTyBPTkxZMREwEAYDVQKQKew1ERU1PIE90TFkxYjAQB
gNVBAsTCURFTU8gT05MWTESMBAGA1UEAxMjREVNTyBPTkxZMRgwFgYJKoZIhvc
NAQkBFglERU1PIE90TFkxwZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAKRg
L5Z5tcp8HkubHFrpC1tub2CEANVBjsXfk/n8rIe/J1XCm2Gv1Q85Fk6pWh8P597
reMvM1XI9gQE/1xBaSEwJv4GuVPtfcGyG8PJmako00d/OkYsYH1ZJG7aIMmJB1
DA5iwZpZDvHmFIgT9EJ7nZAYE/Rb1p6dmJBNZYtOMaXAgMBAAGjgeswgegWHQ
YDVR00BBYFCCeMnFJOsgvF3B4HuaX9fBBDk9xMIG4BgNVHSMGgbAwga2AFCCeMn
FJOsgvF3B4HuaX9fBBDk9xoYGRpIGOMIGLMQswCQYDVQKGEwJYwDESMBAGA1UE
CBMjREVNTyBPTkxZMREwEAYDVQKQKew1ERU1PIE90TFkxYjAQBgNVBAGTCUR
FTU8gT05MWTESMBAGA1UEBxMjREVNTyBPTkxZMREwEAYDVQKQKew1ERU1PIE9
0TFkxGDAWBgkqhkiG9w0BCQEWCURFTU8gT05MWTESMBAGA1UEBmBBAAGI/MA0
GCSqGSIb3DQEBAUAA4GBAIG/L8dbydfkNbydH3wHcF5uUuLG5rajGzput7GrQ
EjKUmKEB+bI/VIRbPQC7wupTGzvWOF0iR7MsY64y5cbpMoGrfZ2qNgNKF+i6WL
lmTfh4+1tKiCMnhTRPMcszjvvgR1WhiVbsYqWBd0FwrkqAUapuUDwctaAxV2pw
Jos47IO
-----END CERTIFICATE-----
```

```
.
dx% capture file txkey
Enter file. End with . on a blank line.
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCkYC+WebXKfB5Lmxxa6Qtbbm9ghADVQSbF35P5/KyHvyZVwpth
r9UPORZ0qVofD+fe63jLzNVyPYEBP9cQWkhmCb+Br1T7X3BshvDyZgJKDthfz
pLGB5WSRu2iDjiQdQwOYlmaWQ7x5hSIE/Rce52QMhPOW9aenZiQTWLTjG1wIDAQ
ABAoGALFdiHvaAKromtgCGuqNpE+YL136kduKXYgN4+JPHuuq+nZ3cqpJzKCM
fOGMI055Hz20390ovPhOHQt4E1v1zyNiZmowcC7xQpdkUXEpCGJQcb2w09zcc
rouFEfK0j3EaxQsU1q1bfSsivNVFB1uryKSFC5ad8m5bTT1LiYDrFTOECQQDR
ck+4wj6xHEKPCfRmCRv8rfZ1BRKIRyudmUI3+j7a60J6S24Z+zSr16oYHDTK
5M6U2GhU1EXdyICTb20EqZwAkEay0T0jD+MjgPVQvr/sxsCOJXv+PkReTzszni
SaDEkBDx+rNwanUVFmdguTKRIRz6ZkzbA7VfT3iP3HwbJ1mFRwJAbBsnoQLJ3x
rqE/CccGo1Quf79QyoMyUhExh/AGuvM8j01TbH3qs11Zjcl9M/QJZ3Noa42ycp
JL+QA3Um/SgAQJBAJYuEC20LOBMzVS1RVA/5zgfNG064snqteVdEavaxL3JEE
Vjmwz2yw2VNYMdmZ1WzdVSeQKxvUj3P3ms3GFpG8CQQCHom0+t9sh11ZtX1nnGbu
/CGK1LLzRX8QIK+/AFwRQfvJad763cc1qyYzNwBSxIeaBbpC0vjdq1DNcaX3aXup
1
-----END RSA PRIVATE KEY-----
```

```
.  
dx% list file  
txcert  
txkey  
dx%
```

The DX now has a certificate and key with which to perform SSL transactions.

### ***Importing from IIS 4 on Windows NT***

The certificate file is in the directory that was specified when the certificate was downloaded.

1. Double-click the certificate file to open the viewer
2. Click the DETAILS tab
3. Click COPY to file. The Certificate Manager Export Wizard opens. Click NEXT.
4. Select the "Base 64 encoded X509" radio button. Click NEXT.
5. Specify a file name and location. Click NEXT.
6. Click FINISH.
7. Click OK when you see the successful completion notice.
8. Exit the Certificate Manager Export Wizard.
9. Close the certificate viewer.

The keys are located within the Key Ring (the key manager program). Follow these instructions to export a key:

1. Click the START button, point to Programs > Windows NT 4.0 Option Pack > Microsoft Internet Information Server, and click Internet Service Manager. The Microsoft Management Console will open.
2. Navigate to the Web site using the object list.
3. Right-click the Web site object and click PROPERTIES in the shortcut menu.
4. Click the DIRECTORY SECURITY tab.
5. Click EDIT in the Secure Communication panel.
6. Click KEY MANAGER.
7. Click the key to export.
8. In the Key menu, point to Export Key, and click BACKUP FILE.
9. Read the security warning and click OK.
10. Select a file location and enter a file name.

11. Click SAVE.
12. Exit the Internet Service Manager.

## Exporting Key and Certificate Files to the DX Appliance:

### Exporting the certificate

The IIS 4.0 certificate can be exported as “base64 encoded X509” format. Simply open the base-64 encoded file in an appropriate text editor and copy its contents to the clipboard. Then, at the DX command prompt, type `capture file txcert`, and paste the certificate information that you copied. Make sure to end the new file with a period on a blank line by itself. Note that you do not need to name the key file “txkey” (the name can be anything you choose).

```
dx% capture file txcert
Enter file. End with . on a blank line.
-----BEGIN CERTIFICATE-----
MIIDEjCCAuOgAwIBAgIBADANBgkqhkiG9w0BAQQFADCBiZELMAkGA1UEBhMCWFgx
EjAQBgNVBAgTCURFTU8gT05MWTESMBAGA1UEBxMJREVNTyBPTkxZMRlWkEAYDQK
Ew1ERU1PIE90TFkxEjAQBgNVBAgTCURFTU8gT05MWTESMBAGA1UEAxMJREVNTyBP
TkxZMRgwFgYJKoZIhvcNAQkBFglERU1PIE90TFkwHhcNMDIwMzA1MjMzAxwHcN
MDIwMzA2MjMzAxwJCBiZELMAkGA1UEBhMCWFgxEjAQBgNVBAgTCURFTU8gT05M
WTESMBAGA1UEBxMJREVNTyBPTkxZMRlWkEAYDQKEw1ERU1PIE90TFkxEjAQBgNV
BAgTCURFTU8gT05MWTESMBAGA1UEAxMJREVNTyBPTkxZMRgwFgYJKoZIhvcNAQkB
FglERU1PIE90TFkwZ8wDQYJKoZIhvcNAQEBBQADgY0AMIGJAoGBAKRgLSZ5tcp8
HkubHFrpC1tub2CEANVBjsXfk/n8rIe/JlXCm2Gv1Q85Fk6pWh8P597reMvM1XI9
gQE/1xBaSEwJv4GuVPtfcGyG8PJmAko00d/OkYsYHlZJG7aIMmJB1DA5iWZpZDvH
mFIgT9EJ7nZAyE/Rb1p6dmJBNZYt0MaXAgMBAAGjgeswgegwhQYDVR00BBYEFCCe
MnFJOsgvF3B4HuaX9fBBDk9xMIG4BgNVHSMGegbAwwa2AFCCeMnFJOsgvF3B4HuaX
9fBBDk9xYoYGRpIGOMIGLMQswCQYDQKGEwJYwDESMBAGA1UECBMJREVNTyBPTkxZ
MRlWkEAYDQKHEw1ERU1PIE90TFkxEjAQBgNVBAoTCURFTU8gT05MWTESMBAGA1UE
CxMJREVNTyBPTkxZMRlWkEAYDQKDEw1ERU1PIE90TFkxGDAWBgkqhkiG9w0BCQEW
CURFTU8gT05MWTESMBAGA1UEBhMCWFgxZGZput7GrQEjKUmKEB+bI/VIRbPQC7wupTGzv
WOF0iR7MsY64y5cbpMoGrfZ2qNgNKF+i6Wl1mTfh4+1tKiCMnhTRPMcszjvvgRlW
hivbsYqWBdOFwrkqAUapuUDwctaAxV2pwJos47IO
-----END CERTIFICATE-----
.
```

The certificate is now on the DX.

### Exporting the Key

First the IIS 4.0 key (iis4key.key) needs to be converted to the DX format. Copy the file from the IIS machine to a UNIX machine in order to convert your key to base-64 encoded format. To do this, locate the key file and execute the following commands:

```
unix% hd iis4key.key | head
```

This will perform a hex dump and display the key file on-screen. Now find the byte pattern “30 82” in the key file, which should be located before the “private-key” text. Strip off everything before the “30 82” using the following commands:

```
unix% dd skip=1 bs=xx < iis4key.key > iis4key.key2
unix% openssl rsa -inform NET -in iis4key.key2 -out iis4key.b64
```

In the argument “**bs=xx**”, “**xx**” is the number of bytes you are stripping out. A byte is a two-digit pair of numbers. For example, “12 34 56 78” equals 4 (four) bytes, so you would enter “**bs=4**”. Typically the number of bytes will be around 30 (thirty). You now have a key in base-64 (iis4key.b64) encoding that can be used with the DX.

Open the base-64 encoded file in a text editor and copy the contents. Then, at the DX command prompt, type the command **capture file txkey**, press RETURN, and paste the contents of the file as follows. Make sure to end the new file with a period on a blank line by itself. Note that you do not need to name the key file “txkey” (the name can be anything you choose).

```
dx% capture file txkey
Enter file. End with . on a blank line.
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCkYC+WebXKfB5Lmxxa6Qtbbm9ghADVQSbF35P5/KyHvyZVwpth
r9UPORZOqVofD+fe63jLzNVyPYEBP9cQwkhMCb+Br1T7X3BshvDyZgJKDtHfzpGL
GB5WSRu2iDjiQdQw0YlmaWQ7x5hSIE/Rce52QMhPOW9aenZiQTWwLTjG1wIDAQAB
AoGALFdiHvaAKromtgCGuqNpE+YL136kduKXYgN4+JPHuuq+nZ3cqpJzKCMfOGMI
055Hz20390ovPh0HQ4E1v1zyNiZmowcC7xQpdkUXEpCGJQcb2w09zcqrouFEfK0
j3EaxQsU1q1bfSsiNVFB1uryKSFC5ad8m5bTT1LiYDrFTOECQQRck+4wj6xHEKP
CfRmCRv8rfZ1BRKIRyudmUI3+j7a60J6S24Z+zSr16oYHDTK5M6U2GhU1EXdyICt
b20EqZwxAkEAY010jD+MjqPVQvr/sxsCOJXv+PkReTzszniSaDEKbZdx+rNWanUV
FmdguTKRIrZ6ZkzbA7VFT3iP3HwbJ1mFRwJAbBsnoQLJ3xrqE/CccGo1Quf79Qyo
MyUhExh/AGuvM8j01TbH3qs11Zjc19M/QJZ3Noa42ycpJL+QA3Um/SgakQJBAJYu
eC20LOBMzVS1RVA/5zgfNG064snqteVdEavaxL3JEEVjmwz2yw2VNyMdmZ1WzdV
SeQKxvUj3P3ms3GFpG8CQCqHom0+t9sh11ZtX1nnGbu/CGK1LLzRX8QIK+/AFwRQ
fvJaD763cc1qyYzNwBSxIeaBbpC0vjdq1DNcaX3aXup1
-----END RSA PRIVATE KEY-----
.
```

Now verify that you have the certificate and key files:

```
dx% list file
txcert
txkey
dx%
```

The DX now has a certificate and key for SSL transactions.

## Importing from IIS 5 on Windows 2000

Follow these steps to export a certificate and key from IIS 5 on Windows 2000.

1. Click the START button, point to Programs > Administrative Tools, and click Internet Service Manager. Alternately, open the Internet Service Manager in the Administrative Tools folder in the Control Panel.
2. Right-click the Web site object and click PROPERTIES in the shortcut menu.
3. Click the DIRECTORY SECURITY tab.
4. Click VIEW CERTIFICATE in the Secure Communications panel. The Certificate Viewer appears.
5. Click the DETAILS tab.
6. Click Copy to file. The Certificate Export Wizard appears. Click NEXT.

7. The Export Private Key panel appears.
8. Choose “YES, EXPORT THE PRIVATE KEY” option. Click NEXT.
9. The Export File Format panel appears.
10. Choose the PERSONAL INFORMATION EXCHANGE - PKCS#12 (PFX) option and any optional choices desired. Click Next.
11. The Password panel appears. Type in the password and confirm the password text boxes. Click NEXT.
12. The File to Export panel appears.
13. Type the path and file name in the File name text box or click Browse to select a location manually. Click NEXT.
14. Completing the Certificate Export Wizard panel appears.
15. Click FINISH.

Now the IIS 5.0 cert and key (iis5certkey.pfx) must be converted to base-64 encoded format. Move the files to a server that has OpenSSL installed and use the following command:

```
unix% openssl pkcs12 -nodes -in iis5certkey.pfx
```

This will print the file which contains the certificate and key on-screen. Scan the file for the relevant certificate and key information. The certificate information will look like:

```
-----BEGIN CERTIFICATE-----
MIIDejCCAu0gAwIBAgIBADANBgkqhkiG9w0BAQFADCBizELMAKGA1UEBhMCWFgx
EjAQBgNVBAGTCURFTU8gT05MWTESMBAGA1UEBxMjREVNTyBPTkxZMREWEAYDVQK
Ew1ERU1PIE90TFkxEjAQBgNVBAsTCURFTU8gT05MWTESMBAGA1UEAxMjREVNTyBP
TkxZMRgwFgYJKoZiIhvcNAQkBFglERU1PIE90TFkWHhcnMDIwMzA1MjMzAxwHcN
MDIwMzA2MjMzAxwJCBizELMAKGA1UEBhMCWFgxEjAQBgNVBAGTCURFTU8gT05M
WTESMBAGA1UEBxMjREVNTyBPTkxZMREWEAYDVQKQEW1ERU1PIE90TFkxEjAQBgNV
BAsTCURFTU8gT05MWTESMBAGA1UEAxMjREVNTyBPTkxZMRgwFgYJKoZiIhvcNAQk
BglERU1PIE90TFkWGZ8wDQYJKoZiIhvcNAQEBBQADgY0AMIGJAoGBAKRgLSZ52tCP8
HkubHFrpC1tub2CEANVBjsXfk/n8rIe/J1XCm2Gv1Q85Fk6pWh8P597reMvM1XI9
gQE/1xBaSEwJv4GuVPtfcGyG8PJmAKo00d/OkYsYH1ZJG7aIMmJB1DA5iWZpZDvH
mFIgT9EJ7nZAYE/Rb1p6dmJBNZYtOMaXAgMBAAGjgeswgegwhQYDVR00BBYFCCE
MnFJ0sgvF3B4HuaX9fBBDk9xMIG4BgNVHSMGegbAwga2AFCCeMnFJ0sgvF3B4HuaX
9fBBDk9xoYGRpIGOMIGLMQswCQYDVQGEwJYWDESMBAGA1UECBMjREVNTyBPTkxZ
MRIWEAYDVQKHEw1ERU1PIE90TFkxEjAQBgNVBAoTCURFTU8gT05MWTESMBAGA1UE
CxMjREVNTyBPTkxZMREWEAYDVQKQEW1ERU1PIE90TFkxGDAWBgkqhkiG9w0BCQEW
CURFTU8gT05MwYIBADAMBgNVHRMERTADAQH/MA0GCSqGSIb3DQEBBAAUAA4GBAIg/
L8dbydfkNbydH3wHcF5uUuLG5rajGzput7GrQEjKUmKEB+bI/VIRbPQC7wupTGzv
WOF0iR7MsY64y5cbpMoGrfZ2qNgNKF+i6WL1mTfh4+1tKiCMnhTRPMcszjvvgR1W
hivbsYqWBdOFwrkqAUapuUDwctaAxV2pwJos47IO
-----END CERTIFICATE-----
```

And the key will look like:

```
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCkYC+WebXKfB5Lmxxa6Qtbbm9ghADVQSbF35P5/KyHvyZVwpth
r9UPORZ0qVofD+fe63jLzNVyPYEBP9cQWkhMcb+Br1T7X3BshvDyZgJKDtHfzpzGL
GB5WSRu2iDjiQdQwOYlmaWQ7x5hSIE/RCe52QMhP0W9aenZiQTWwLTjG1wIDAQAB
AoGALFdiHvaAKromtgCGuqNpE+YL136kduKXYgN4+JPHuuq+nZ3cqpJzKCMFOGMI
055Hz20390ovPhOHQt4E1v1zyNiZmowcC7xQpdkUXEpCGJQcb2w09zczrouFEfK0
j3EaxQsU1q1bfSsiNVFB1uryKSFC5ad8m5bTT1LiYDrFT0ECQQDRck+4wJ6xHEKP
CfRmCRv8rfZ1BRKIRyudmUI3+j7a60J6S24Z+zSr16oYHDTK5M6U2GhU1EXdyICt
b20EqZwxAkEAy010jD+MjqPVQvr/sxsCOJXv+PkReTzszniSaDEKbZdx+rNwanUV
FmdguTkRIrZ6ZkzbA7Vft3iP3HwbJ1mFRwJAbBsnoQLJ3xrqE/CccGo1Quf79Qyo
MyUhExh/AGuvM8j01TbH3qs11Zjc19M/QJZ3Noa42ycpJL+QA3Um/SgakQJBAJYu
eC20LOBMzVS1RVA/5zgfnG064snqteVdEavaxL3JEEVjmwz2yw2VNyMdmZ1WzdV
SeQKxvUj3P3ms3GFpG8CQCqHom0+t9sh11ZtX1nnGbu/CGK1LLzRX8QIK+/AFwRq
fvJaD763cc1qyYZNWBSxIeaBbpC0vjdq1DNcaX3aXup1
-----END RSA PRIVATE KEY-----
```

Leave the window open and note the location of this information. You will copy and paste it onto the DX in the next step.

### Exporting Key and Certificate Files to the DX Appliance

Open a SSH connection to the DX. Copy and paste the key and certificate information you just noted into the DX using the following steps

At the DX command prompt, type the following command `capture file txcert`, then paste the certificate information. Make sure to end the new file with a period on a blank line by itself. Note that you do not need to name the key file “txcert” (the name can be anything you choose).

```
dx% capture file txcert
Enter file. End with . on a blank line.
-----BEGIN CERTIFICATE-----
MIIEdejCCAu0gAwIBAgIBADANBgkqhkiG9w0BAQQFADCBiZELMAKGA1UEBhMCWFgx
EjAQBgNVBAGTCURFTU8gT05MWTESMBAGA1UEBxMjREVENTyBPTkxZMRiWEAYDVQQK
Ew1ERU1PIE90TFkxEjAQBgNVBAsTCURFTU8gT05MWTESMBAGA1UEAxMjREVENTyBP
TkxZMRgwFgYJKoZIhvcNAQkBFjERU1PIE90TFkWHhcNMDIwMzA1MjM1MzAxwHcN
MDIwMzA2MjM1MzAxwCBiZELMAKGA1UEBhMCWFgxEjAQBgNVBAGTCURFTU8gT05M
WTESMBAGA1UEBxMjREVENTyBPTkxZMRiWEAYDVQQKEw1ERU1PIE90TFkxEjAQBgNV
BAsTCURFTU8gT05MWTESMBAGA1UEAxMjREVENTyBPTkxZMRgwFgYJKoZIhvcNAQkB
FjERU1PIE90TFkGZ8wDQYJKoZIhvcNAQEBBQADGgY0AMIGJAoGBAKRGL5Z5t8cp8
HkubHFrpC1tub2CEANVBjSxXfk/n8rIe/J1XCm2Gv1Q85Fk6pWh8P597reMvM1XI9
gQE/1xBaSEwJv4GuVptfcGyG8PJmako00d/OkYsYH1ZJG7aIMmJB1DA5iWZpZDVh
mFIgT9EJ7nZAYE/Rb1p6dmJBNZYt0MaXAgMBAAGjgeswgegwHQYDVR00BBYEFCce
MnFJ0sgvF3B4HuaX9fBBDk9xMIG4BgNVHSMEgBAwga2AFCCeMnFJ0sgvF3B4HuaX
9fBBDk9xoYGRpIGOMIGLQswCQYDVQQGEwJYWDESMBAGA1UECBMjREVENTyBPTkxZ
MRiWEAYDVQQHEw1ERU1PIE90TFkxEjAQBgNVBAoTCURFTU8gT05MWTESMBAGA1UE
CxMjREVENTyBPTkxZMRiWEAYDVQQDEw1ERU1PIE90TFkxGDAWBgkqhkiG9w0BCQEW
CURFTU8gT05MWTESMBAGA1UEBhMCWFgxMA0GCSqGSIb3DQEBBAUAA4GBAIG/
L8dbydfkNbydH3wHcF5uUuLG5raJGzput7GrQEjKUmKEB+bI/VIRbPQC7wupTGzv
W0F0iR7MsY64y5cbpMoGrfZ2qNgNKf+i6WL1mTfh4+1tKiCMnhTRPMcszjvwgr1W
hivbsYqWBdOfwrkqAUapuUDwctaXv2pwJos47IO
-----END CERTIFICATE-----
```

Your certificate is now on the DX.

Now, at the DX command prompt, type the following command, `capture file txkey`, press ENTER, then paste the key information you noted previously. Make sure to end the new file with a period on a blank line by itself. Note that you do not need to name the key file “txkey” (the name can be anything you choose).

```
dx% capture file txkey
Enter file. End with . on a blank line.
-----BEGIN RSA PRIVATE KEY-----
MIICXQIBAAKBgQCkYC+WebXKfB5Lmxxa6Qtbbm9ghADVQsBf35P5/KyHvyZVwpth
r9UPORZ0qVofD+fe63jLzNVyPYEBP9cQWkhMCb+Br1T7X3BshvDyZgJKDtHfzpGL
GB5WSRu2iDjiQdQwOYlmaWQ7x5hSIE/RcE52QMhPOW9aenZiQTWWLTjG1wIDAQAB
AoGALFdiHvaAKromtgCGuqNpE+YL136kduKXYgN4+JPHuuq+nZ3cqpJzKCMFOGMI
055Hz20390ovPh0HQ4E1v1zyNiZmowcC7xQpdkUXEpCGJQcb2w09zcqrouFEfK0
j3EaxQsU1q1bfSsivNVFB1uryKSFC5ad8m5bTT1LiYDrFTOECQQRck+4wj6xHEKP
CfRmCRv8rfZ1BRKIRyudmUI3+j7a60J6S24Z+zSr16oYHDTK5M6U2GhU1EXdyICT
b20EqZwxAkEAY010jD+MjqPVQvr/sxsCOJXv+PkReTzszniSaDEKbZdx+rNwanUV
FmdguTkRiRZ6ZkzbA7VfT3iP3HwbJ1mFRwJAbBsnoQLJ3xrqE/CccGo1Quf79Qyo
MyUhExh/AGuvM8j01TbH3qs11Zjc19M/QJZ3Noa42ycpJL+QA3Um/SgakQJBAJYu
eC20LOBMzVS1RVA/5zgfNGO64snqteVdEavaxL3JEEVjmw2yw2VNyMdmZ1WzdV
SeQKxvUj3P3ms3GFpG8CQQCHom0+t9sh11ZtX1nnGbu/CGK1LLzRX8QIK+/AFwRQ
fvJaD763cc1qyYzNwBSxIeaBbpC0vjdq1DNcaX3aXup1
-----END RSA PRIVATE KEY-----
.
```

Now verify that you have the certificate and key files:

```
dx% list file
txcert
txkey
dx%
```

The DX now has a certificate and key with which to perform SSL transactions.

## Importing from iPlanet

The `pk12util` command available on the iPlanet server allows you to export certificates and keys from the internal database of iPlanet server and import them into the DX. By default, `pk12util` uses certificate and key databases named `cert7.db` and `key3.db`.

To export a certificate and key from the iPlanet server, perform the following steps:

1. Go to the `server_root/alias` directory containing the databases.

```
iplanet% cd server_root/alias
```

2. Add `server_root/bin/https/admin/bin` to your path.
3. Locate `pk12util` in `server_root/bin/https/admin/bin`.
4. Set the environment. For example:

```
On Unix: setenv LD_LIBRARY_PATH/server_root/bin/https/lib:${LD_LIBRARY_PATH}
On IBM-AIX: LIBPATH
On HP-UX: SHLIB_PATH
On NT, add it to the PATH
LD_LIBRARY_PATH server_root/bin/https/bin
```

You can find the path for your machine as `server_root/https-admin/start`.

1. Enter the `pk12util` command to view available options:

```
iplanet% pk12util
```

2. Perform the actions required. For example, in Unix you would enter:

```
iplanet% pk12util -o certpk12 -n Server-Cert [-d /server/alias] [-P https-test-host]
```

3. Enter the database password.

4. Enter the `pkcs12` password.

To import the SSL key and certificate into the DX, you must run the OpenSSL command on the file output from the `pk12util` utility as mentioned previously:

```
unix% openssl pkcs12 -in certpk12
```

This will print out the certificate and key in the base-64 encoded format, which you will then need to copy and paste onto the DX using the `capture file` command. The above example assumes that `certpk12` was the output from the `pk12util` command.

## Generating Keys and Certificates

---

### GEN KEY

Usage: `gen key <key_file>`

The “`gen key file`” command is short for “generate private key.” It generates a 1024-bit RSA private key. A filename **MUST** be specified, and the key is saved into that file.

Sample:

```
dx% gen key
save to filename [return for none]: my.key
keyfile passphrase (keypass) [return for none]:
Saved as my.key...
```

### GEN CSR

Usage: `gen csr`

This command is short for “generate certificate signing request.” It prompts the user for information (i.e., country, organization name, common name, state, city, etc.), then creates a certificate signing request based upon the key and the user’s input. The most important field is the “common name”, which must match the DNS name of the cluster’s listen address. The CSR should be sent to a Certificate Authority (like Verisign or Thawte) in exchange for an official certificate, which can then be imported into the DX via the `capture file`. A filename **MUST** be specified



and the CSR is saved to the file. The `list file` command can be used to view the CSR.

Sample:

```
dx% gen csr
```

```
Please supply the requested information to form the Distinguished Name (DN)
incorporated in your certificate.
```

```
You may accept the default value shown in brackets by pressing enter,
or force a field to be blank by entering a single '.' and pressing enter.
```

```
Please note: to prevent security errors, the Common Name field should
match the host name (fully-qualified domain name) that browsers address
this machine as.
```

```
Country name (2 letter code) [US]:
State or province name (full name) [California]:
Locality name (eg, city) []:
Organization (company) name []:
Organizational unit name []:
Common name (advertised host name) [dx.juniper.net]:
Email address []:
Certificate Request:
```

```
Data:
```

```
Version: 0 (0x0)
Subject: C=US, ST=California, CN=dx.juniper.net
Subject Public Key Info:
  Public Key Algorithm: rsaEncryption
  RSA Public Key: (1024 bit)
    Modulus (1024 bit):
      00:cf:c1:7e:d8:3c:68:be:26:9f:98:c0:07:d1:c9:
      fb:57:80:d8:17:28:20:27:74:24:f3:5a:df:13:0a:
      54:60:ba:39:5c:bf:8d:85:4e:56:14:b2:6c:26:03:
      5d:92:80:f6:0b:44:4d:cc:d4:a4:99:11:6d:ce:a2:
      bb:4c:b6:7d:24:75:ac:95:53:ae:2a:90:48:51:bf:
      51:68:15:39:f5:4b:2c:7c:5e:50:6b:5b:f5:4a:5e:
      d1:6f:60:a9:de:6e:96:ed:5c:95:e1:b0:33:97:b8:
      d8:4c:78:7c:e6:9d:dd:68:76:50:97:c5:99:0c:43:
      72:69:bc:9e:4e:ab:c7:a1:2b
    Exponent: 65537 (0x10001)
```

```
Attributes:
```

```
  a0:00
  Signature Algorithm: md5WithRSAEncryption
    64:90:e2:c1:7a:41:c0:fd:51:4b:2d:79:71:43:69:9f:1d:82:
    80:54:67:45:5b:48:b1:71:c2:c3:51:e2:94:d7:a3:66:45:94:
    05:24:37:cb:33:09:4f:cb:4b:7c:66:6f:af:ac:a3:47:7c:19:
    71:42:7d:26:c8:bd:fc:6e:b2:2b:99:d0:24:53:d2:77:27:13:
    4f:ff:59:ff:f1:6a:c5:0e:d1:35:27:f0:4c:63:dc:50:22:e8:
    29:88:4b:a0:70:f0:1f:16:d5:bc:61:43:60:8a:e0:ff:f8:f6:
    df:f9:73:8c:81:46:77:67:50:30:df:6f:b4:62:76:36:8e:60:
    3a:00
```

```
Saving as my.csr...
```

```
dx%
```

## GEN SSC

Usage: `gen ssc <key_file> <ssc_file>`

This command is short for “generate self-signed certificate.” It accepts a 1024-bit RSA private key, then prompts the user for information (i.e., country, organization name, common name, state, city, etc.), then generates a self-signed certificate based upon the key and the user's input. The most important field is the “common name,” which must match the DNS name of the cluster's listen address. This certificate can be used on the DX. The certificate will be “phony,” but it may be sufficient for a company's internal test lab. The filename MUST be specified, and the certificate is saved to that file.

Sample:

```
dx% gen ssc
save to filename [return for none]: my.ssc
keyfile passphrase (keypass) [return for none]:
Saved as my.ssc...
```

Please supply the requested information to form the Distinguished Name (DN) incorporated in your certificate.  
You may accept the default value shown in brackets by pressing enter, or force a field to be blank by entering a single '.' and pressing enter.

Please note: to prevent security errors, the Common Name field should match the host name (fully-qualified domain name) that browsers address this machine as.

```
Country name (2 letter code) [US]:
State or province name (full name) [California]:
Locality name (eg, city) []:
Organization (company) name []:
Organizational unit name []:
Common name (advertised host name) [dx.juniper.net]:
Email address []:
Certificate:
Data:
  Version: 3 (0x2)
  Serial Number: 0 (0x0)
  Signature Algorithm: md5WithRSAEncryption
  Issuer: C=US, ST=California, CN=dx.juniper.net
  Validity
    Not Before: Apr  8 17:48:29 2002 GMT
    Not After : Apr  8 17:48:29 2003 GMT
  Subject: C=US, ST=California, CN=dx.juniper.net
  Subject Public Key Info:
    Public Key Algorithm: rsaEncryption
    RSA Public Key: (1024 bit)
      Modulus (1024 bit):
        00:bc:64:f1:2c:a7:96:ee:d5:df:64:b2:f3:2a:a3:
        99:e0:c3:41:ba:97:3c:dc:6c:15:ba:dc:b0:bc:a2:
        5a:9c:78:12:1c:c7:22:ff:16:f2:d5:8a:8f:0b:2a:
        f8:02:f6:35:b1:5b:b5:9d:f9:35:4c:36:0d:6b:bc:
        0e:ce:0a:cd:6a:b9:bd:2e:db:e6:82:b6:c2:c8:6a:
        4c:5f:d7:e7:78:cc:8d:2e:22:c9:15:52:df:97:ae:
        71:e8:c8:c1:b3:4c:13:6d:d7:01:f7:1b:4a:4e:51:
        bf:dc:78:c1:1c:96:b2:da:33:fa:88:20:a0:5e:ec:
        9f:57:73:59:5a:90:4b:23:3b
      Exponent: 65537 (0x10001)
```

```

Signature Algorithm: md5WithRSAEncryption
89:d9:f0:a4:d0:ec:5c:cb:6a:24:28:19:8d:95:1d:ff:27:bc:
f2:0f:f7:97:d4:35:a1:75:6e:8a:28:ce:17:55:01:ed:36:95:
c6:28:01:24:11:46:23:ee:da:1d:f5:53:82:65:18:84:dd:99:
33:c5:9b:62:fa:af:d9:29:28:32:13:a0:47:3d:74:82:ec:d1:
04:98:cb:29:11:ac:6e:21:39:37:3f:a7:70:86:0b:30:43:32:
24:62:1d:40:d1:0c:d0:c5:cc:74:24:d7:47:2b:e9:7f:f6:fd:
5f:68:08:88:30:40:44:5e:07:5f:f3:e5:fc:ed:fd:c9:d3:e3:
a1:6b
Saving as my.ssc...
dx%

```

## SSL Ciphersuite Details

The following SSL ciphersuites are available on the DX.

**Table 1: SSL Ciphersuites**

Ciphersuite	Description
Common SSL Ciphers	The fastest ciphersuites from both the Strong and Export groups.
RC4-MD5	
RC4-SHA	
EXP-RC4-MD5	
EXP-RC2-CBC-MD5	
EXP1024-RC4-MD5	
EXP1024-RC2-CBC-MD5	
Strong SSL Ciphers	The highest-security ciphersuites that are suitable for use in USA.
RC4-MD5	
RC4-SHA	
DES-CBC3-MD5	
DES-CBC3-SHA	
AES256-SHA	
AES128-SHA	
IDEA-CBC-SHA	
IDEA-CBC-MD5	
Export SSL Ciphers	Lower-security ciphersuites that are suitable for export
EXP-RC4-MD5	
EXP-RC2-CBC-MD5	
EXP1024-RC4-MD5	
EXP1024-RC2-CBC-MD5	
DES-CBC-MD5	
DES-CBC-SHA	

**Table 1: SSL Ciphersuites (continued)**

Ciphersuite	Description
All SSL Ciphers	Strong and Export
RC4-MD5	
RC4-SHA	
DES-CBC-MD5	
DES-CBC-SHA	
DES-CBC3-MD5	
DES-CBC3-SHA	
AES256-SHA	
AES128-SHA	
IDEA-CBC-SHA	
IDEA-CBC-MD5	
EXP-RC4-MD5	
EXP-RC2-CBC-MD5	
EXP1024-RC4-MD5	
EXP1024-RC2-CBC-MD5	

### Specifying Your Own List of SSL Ciphersuites

You can specify a file containing a list of SSL ciphersuites to configure an SSL cluster or redirector.

#### Capturing a Cipherfile

The cipherfile can be captured using the `capture file` command. It should contain a list of ciphersuites that conform to the OpenSSL standard. A sample list looks like:

```
RC4-MD5:MEDIUM:!DH:HIGH:!EXPORT56:-AES256-SHA
```

These commands support this feature:

```
% set cluster 1 listen ssl cipherfile <filename>
% set cluster 1 listen ssl ciphersuite file
% show cluster 1 listen ssl cipherfile
% show cluster 1 listen ssl cipherlist

% set cluster 1 target ssl cipherfile <filename>
% set cluster 1 target ssl ciphersuite file
% show cluster 1 target ssl cipherfile
% show cluster 1 target ssl cipherlist
```

If the ciphersuite is not file, then the cipherfile is ignored.

If SSL is enabled and a write is done, then the DXSHELL will validate the cipherfile in the same way that OpenSSL validates a ciphersuite list. OpenSSL is very lenient, but if OpenSSL does not complain, then DXSHELL will not either. For example, if cipherfile is set to demokey, OpenSSL will allow it because the first line “-----BEGIN RSA PRIVATE KEY-----” has a valid “RSA” keyword in it.

The “`show. cipherlist`” commands are provided so the user can confirm the actual list of ciphersuites to be used. Showing the cipherlist will print out a detailed line for each ciphersuite, showing the name, version, key exchange, authentication, encryption, and hash method.

**NOTE:** The “`show. cipherlist`” commands have no tab-completion because there is no way to distinguish a cipherfile from any other file.

**NOTE:** There is no WebUI support for specifying a cipherfile.

Some sample commands to configure a cipherfile are:

```
% capture file myciphers
Enter file. End with . on a blank line.
RC4-MD5:MEDIUM:HIGH:!EXPORT56
.

% set cluster 1 listen ssl ciphersuite file
(*)% set cluster 1 listen ssl cipherfile myciphers
(*)% write
% show cluster 1 listen ssl cipherlist
Cipherlist:
RC4-MD5          SSLv3 Kx=RSA      Au=RSA  Enc=RC4(128)  Mac=MD5
RC4-MD5          SSLv2 Kx=RSA      Au=RSA  Enc=RC4(128)  Mac=MD5
AES128-SHA      SSLv3 Kx=RSA      Au=RSA  Enc=AES(128)  Mac=SHA1
IDEA-CBC-SHA    SSLv3 Kx=RSA      Au=RSA  Enc=IDEA(128) Mac=SHA1
RC4-SHA         SSLv3 Kx=RSA      Au=RSA  Enc=RC4(128)  Mac=SHA1
IDEA-CBC-MD5    SSLv2 Kx=RSA      Au=RSA  Enc=IDEA(128) Mac=MD5
RC2-CBC-MD5     SSLv2 Kx=RSA      Au=RSA  Enc=RC2(128)  Mac=MD5
AES256-SHA      SSLv3 Kx=RSA      Au=RSA  Enc=AES(256)  Mac=SHA1
DES-CBC3-SHA    SSLv3 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=SHA1
DES-CBC3-MD5    SSLv2 Kx=RSA      Au=RSA  Enc=3DES(168) Mac=MD5
%
```

## The SSL AppRules Feature

A variable has been added to the Application Rules to support this feature:

- `ssl_cipher_bits`

along with two new test operators:

- `less_than`
- `greater_than`

The new test operators will only work with the `ssl_cipher_bits` test variable and the `ssl_cipher_bits` test variable will only work with Request Sentry rules.

The general usage to form a complete test condition is as follows:

```
RS: ssl_cipher_bits <less_than|greater_than> "<bit_length>"
```

where the `<bit_length>` can be an integer value between 0 and 1024. Typical values in the real world would be 40, 56, or 128.

This test condition can be used as a means of redirecting clients that do not have sufficiently strong browsers to a Web page that would instruct them on how to download such a page. For example:

```
RS: ssl_cipher_bits less_than "128" then redirect
"http://browserupgrade.mysite.com/mysite/upgrade.html"
```

For additional information on Application Rules, see “Configuring OverDrive Application Rules” on page 297.

## Forcing Clients to use HTTPS with Cluster Redirection (Auto SSL)

---

The Cluster Redirection feature allows you to redirect requests from a browser to a new location or redirect requests using a different protocol (HTTP or HTTPS). Some examples of uses for this feature are:

- Redirecting all requests coming in via HTTP on port 80 to the *same page* using HTTPS on port 443
- Redirecting all requests coming in via HTTP on port 80 to a *new page* using HTTP on port 80
- Redirecting all requests coming in via HTTP on port 80 to a *new page* using HTTPS on port 443

To redirect client requests, the DX responds with the HTTP 302 “temporarily moved” response code in compliance with RFC 2616. The response also contains the new location in an HTTP Location header which both HTTP 1.0 and HTTP 1.1-compliant clients recognize.

**NOTE:** When using Auto SSL to redirect requests from HTTP:// to HTTPS://, any hard-coded HTTP links in the content page will get redirected to HTTP with an HTTP 302 redirect message. The 302 redirect message will be in clear using HTTP, so the browsers may see warnings which indicate that the page contains “secure and non-secure items.” The only non-secure items are the HTTP 302 redirect messages.

### EXAMPLE: Configuring Cluster Redirection to Redirect HTTP Requests to HTTPS

The following example shows how to use a Redirector to redirect all incoming HTTP requests on port 80 to be HTTPS requests on port 443 of the same VIP. The VIP in this example is 205.178.13.100.

1. Add a redirector with the same VIP as the cluster where you wish to redirect requests:

```
dx% add redirector
dx% set redirector 1 listen vip 205.178.13.100
```

2. Set the listen port for the redirector to 80:

```
dx% set redirector 1 listen port 80
```

3. Set the target port for the redirector to 443:

```
dx% set redirector 1 port 443
```

4. Set the redirector protocol to HTTPS. This will instruct the browser to use HTTPS when connecting to the redirected location:

```
dx% set redirector 1 protocol https
```

5. Set the host where the redirector will direct requests. You can enter either the fully-qualified domain name or the IP address of the host where requests should be redirected. You should NOT use an IP address if either of the following is true:

- The VIP is a private IP address.
- Multiple DXs are being load balanced by an upstream load balancer

```
dx% set redirector 1 host www.mysite.com
dx% set redirector 1 host 205.178.13.100
Enable the Redirector:
dx% set redirector 1 enabled
```

6. OPTIONAL: Configure the redirector to redirect requests to a custom URL. By default, clients are redirected to the same page initially requested at the new location.

If you would like to send the browser to a different page, such as a secure login page, you must set a custom URL and set the URL method to custom. The custom URL must be configured before the URL method.

```
dx% set redirector 1 customURL "/secure_login.html"
dx% set redirector 1 URLmethod custom
```

## Configuring SSL Client Authentication

---

### Overview

SSL client authentication lets the DX accept connections only from clients possessing the proper credentials; in this case, an SSL Client Certificate designated as valid. SSL client authentication can be enabled on a VIP/cluster basis. Clients will be unable to access restricted information unless they possess a valid SSL Client Certificate. Unauthorized clients do not have the opportunity to make an HTTP request before their connection is terminated.

To authenticate clients, the DX can use root certificates and corresponding certificate revocation lists (CRL) issued by well-known, trusted Certificate Authorities (CA), such as Verisign, Thawte, etc. or certificates and CRLs from an in-house CA.

In an Enterprise environment, it is likely that there will be an in-house CA that provides certificates for client authentication. The in-house CA can publish client certificates and intermediate CA certificates that allow other organizations within the Enterprise environment to act as their own CAs.

It is important to note that the DX DOES NOT perform the following tasks:

- The DX does not act as a CA.
- The DX does not generate its own CA certificate or CRLs for that certificate.
- The DX does not generate client certificates.

These items must be generated outside of the DX and imported to the DX for use in the client authentication process. More detailed information on how the DX stores and presents CA certificates and CRLs is outlined in the following sections.

### **Certificate Authority (CA) Certificate Presentation**

The DX can store and present (i.e., advertise) one or more valid CA certificates to the client during the SSL handshake. The advertised CA certificate(s) can either be root certificates, from a well-known trusted CA, in-house CA certificates, or intermediate CA certificates. The DX is capable of storing multiple CA certificates per VIP. This allows you to present one or more CA certificate to clients based upon the VIP for the client connection.

All certificates listed in the advertised CA certificate file must be in base64-encoded format. The following is an example of this format:

```
-----BEGIN CERTIFICATE-----
MIICpDCCAgoCAQEDQYJKoZIhvcNAQEEBQAwgBwxCzAJBgNVBAYTA1VTMRMwEQYD
VQIIEwpDYWxpZm9ybmlhMREwDwYDVQQHEWhDYW1wYmVsbnBDEZMBcGA1UEChMQUmVkbG1uZSBOZXR3b3JrczEUMBIGA1UECXMlRW5naW5lZXJpbmcxKjAoBgNVBAMTIUVuZ21uZWVyaW5nIEN1cnRpZm1jYXR1IEF1dGhvcml0eTEoMCMYGCsGSIb3DQEJARYZ
ZW5nY2FAcmVkbG1uZW5ldHdvcmtzLmNvbRcNMDIxMDMxMDExMjI4WhcNMDIxMTA3
MDExMjI4WjAUMBIGA1UECXMlRW5naW5lZXJpbmcxKjAoBgNVBAMTIUVuZ21uZWVyaW5nIEN1cnRpZm1jYXR1IEF1dGhvcml0eTEoMCMYGCsGSIb3DQEJARYZ
dGhvcml0eTEoMCMYGCsGSIb3DQEJARYZ
EzARBgNVBAgTCkNhbnB3JuaWExETAPBgNVBACTCENhbXBiZWxsMRkwFwYDVQQK
ExBSZWRSaW5lIE5ldHdvcmtzMSMwIQYDVQQLExpSb290IEN1cnRpZm1jYXR1IEF1
dGhvcml0eTEoMCMYGCsGSIb3DQEJARYZ
dXR3b3JpdHkxKDAmbGkqhkiG9w0BCQEWGjJsbmNhQHJ1ZGxpbnVuzXR3b3Jrcy5j
b22CAQEDQYJKoZIhvcNAQEEBQADgYEAhudjWq+t1tx0cJa63H36eQgBrew6QNNk
PtDdC5Lojhu9dETmR+GKza1YyyD0Kmz1/QIx4GFwthNRXoUYWxw/KWgayu1Gzru
JFbdQA004YiXYL9EeAWHxwhn0H+RHGtE+qjJF1YhXX31onnQKyvKsuKxfbG7Nmku
jrc42BglWu=
-----END CERTIFICATE-----
```

### **Trusted Certificate Authority (CA) Certificate Storage**

The DX can also maintain a list of CA certificates that are considered “trusted.” These trusted certificates are used to validate the certificate chain presented by the client. While the advertised list of certificates may only comprise a portion of the client’s certificate chain, the trusted list must comprise the entire certificate chain for successful client authentication.

A certificate chain is a list of certificates formed by referring to each issuer of a certificate. For example, if root CA “Trusted Certs, Inc.” issues an intermediate CA certificate to “Company X”. Company X, in turn, issues a client certificate to employee Alice, then a certificate chain is formed from the root CA to the intermediate CA to the employee.

If Alice presents a certificate to the DX that advertises Company X's intermediate CA certificate, then Alice can supply her client certificate for authentication. As part of the authentication process, the DX will walk the certificate chain all the way back to the root CA certificate validating each one along the way.



In order to accomplish this, the trusted CA certificate file must contain not only Company X's intermediate CA certificate, but also the root CA certificate of Trusted Certs, Inc. If a trusted CA certificate file is not specified by the user, then a default trusted list of certificates is used by the DX. This list is composed of all the major well-known CAs. Note that the list of trusted CA certificates does not include the client's certificates.

All certificates listed in the trusted CA certificate file must be in base64-encoded format.

### **Certificate Revocation List (CRL)**

The DX will terminate an SSL handshake if a client's certificate is present in a customer-specified Certificate Revocation List (CRL). One CRL may exist per entry in the trusted CA certificate file. A CRL is not required for activating SSL Client Authentication on a particular VIP.

The CRL must be in base64-encoded format. An example of this format is as follows (note that the header and trailer must match exactly as shown):

```
-----BEGIN X509 CRL-----
MIICpDCCAgoCAQEdQYJKoZIhvcNAQEEBQAwbwCzAJBgNVBAYTA1VTMRMwEQYD
VQIQIExpDYWxpZm9ybm1hMREwDwYDVQQHEwhDYW1wYmVsbnZEMzBGA1UEChMUMkVh
bG1uZSB0ZXR3b3JrczEUMBIGA1UECMLRW5naW51ZXJpbmcxKjAoBgNVBAMTlUUVu
Z21uZWVyaW5nIEN1cnRpZm1jYXR1IEF1dGhvcml0eTEoMCYGCsGSIb3DQEJARYZ
ZW5nY2FACmVkbG1uZW51dHdvcm1zLmNvbRcnMDIxMDMxMDEEMjI4WhcNMDIxMTA3
MDEEMjI4WjAUMBIQAQMDTAyMTAzMTAxMDAxM1qgggEEMIIBADCB/QYDVR0jBIH1
MIHygBSU0vjI1Dn+HXdpi22BMTpgBFLLrKGB1qSB0zCBODELMAKGA1UEBHMVVMx
EzARBgNVBAGTCKNhG1mb3JuawExETAPBgNVBACTCENhbXBiZWxsMRkwFwYDVQQK
ExBSZWRSaw51IE51dHdvcm1zMSMwIQYDVQQLExpSb290IEN1cnRpZm1jYXR1IEF1
dGhvcml0eTEvMCOGA1UEAxMmUmVkbG1uZSB0ZXR3b3JrcyBDZXJ0awZpY2F0ZSB0
dXR0b3JpdHkxKDAmBglkqhkIG9w0BCQEWGXJsbmNhQHJ1ZGxpbnVuzXR3b3Jrcy5j
b22CAQEdQYJKoZIhvcNAQEEBQAQdGyEAhudjWq+t1tx0cJa63H36eQgBRew6QNnK
PtDdC5Lojhu9dETmR+GKza1YyyDOKmz1/QIx4GFwthNRXoUYWXww/HWgayu1Gzru
NFbdQA004YiXYL9EeAWHXwhnOH+RHGtE+qjJF1YhXX31onnQKvYKsuKxfbG7Nmku
jrc42BgWuQ=
-----END X509 CRL-----
```

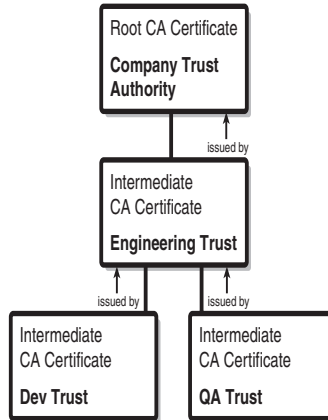
If a CRL is present, the client must satisfy the following three criteria in order to connect to the DX and make requests.

- The client must possess a private key corresponding to the client certificate.
- The client certificate's certificate chain must be valid.
- The client's certificate must not exist within any Certificate Revocation List (CRL) corresponding to any certificate in the certificate chain.

### **Example of Chain Certificates and CRLs**

Assume that you have a root CA known as “Company Trust Authority.” It is a root CA because its certificate is self-signed. That is, the issuer and the subject of the certificate are the same. Then let's assume that you create an intermediate CA called “Engineering Trust” whose certificate has been issued by the Company Trust Authority. Finally, let's assume that you have two additional intermediate CAs, “Development Trust” and “QA Trust” whose certificates have been signed by Engineering Trust. This effectively creates the certificate chain shown in Figure 52.

**Figure 52: SSL Certificate Chain**



Based upon this certificate chain, an organization can issue certificates to various clients. In this example, you can issue a certificate to employee Alice from the Development Trust CA, and a certificate to Bob from the QA Trust CA. Note that in this case, both client certificates have been signed by intermediate CAs.

If you want to configure a DX such that only those who have certificates from Development Trust are allowed access to the content available through that DX, then you would set up an advertised CA certificate list with the Development Trust CA certificate. However, our trusted CA certificate file would have entries for the Development Trust CA, the Engineering Trust CA, and the Company Trust Authority CA. Our arrangement would be something depicted in the following diagram. All entries are in base64-encoded format. Refer to Figure 53.

**Figure 53: SSL Advertised and Trusted Lists**

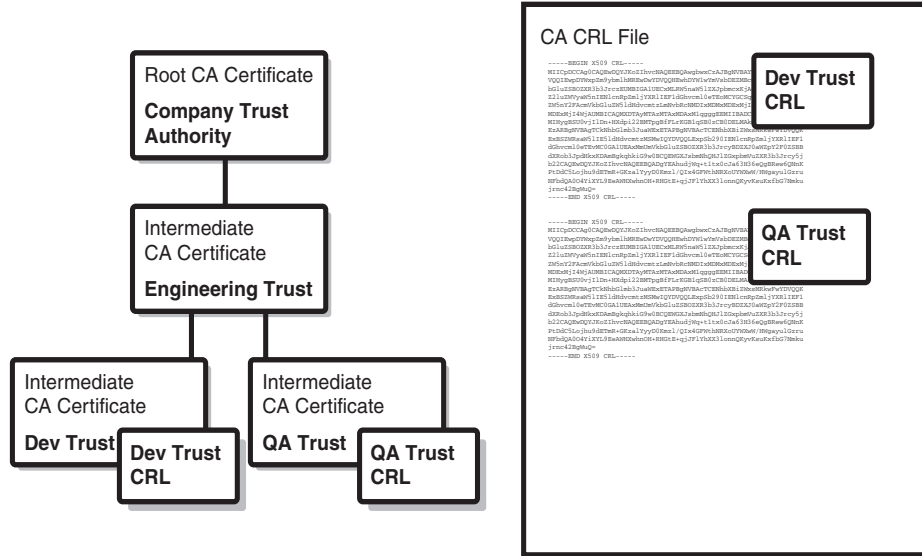


This arrangement is required to satisfy the successful client authentication SSL handshake criteria because the DX must be able to validate the certificate chain all the way back to a root CA that it considered trusted. Note that each entry in the certificate files will be in base64-encoded format.

Client certificates may need to be invalidated from time to time. For example, if an employee who is issued a client certificate leaves the company, you need to establish a mechanism whereby the invalidated certificate is stored in some way relative to the “trusted list” of CA certificates. This is precisely the purpose of the CRL.

When a CRL exists for a certificate in the certificate chain, that CRL is consulted to make sure that the client's certificate is not on that list. Note that entries in a CRL are only certificates that have been issued by that CA. For the previous example, you might have this arrangement with regard to CRLs. Note that the CRL entries are also in base64-encoded format. Refer to Figure 54.

Figure 54: SSL In-House Control



**DXSHELL Commands for SSL Client Authentication**

Use these commands for SSL Client Authentication.

To enable SSL Client Authentication for a cluster, type the command:

```
dx% set cluster <name> listen ssl clientauth enabled
```

To disable SSL Client Authentication for a cluster, type the command:

```
dx% set cluster <name> listen ssl clientauth disabled
```

To configure the Advertised CA certificate file for a cluster, type the command:

```
dx% set cluster <name> listen ssl clientauth cacertfile <filename>
```

To configure the CRL file for a cluster, type the command:

```
dx% set cluster <name> listen ssl clientauth cacrlfile <filename>
```

To configure the trusted CA certificate file for a cluster, type the command:

```
dx% set cluster <name> listen ssl clientauth catrustfile <filename>
```

To clear the Advertised CA certificate file for a cluster, type the command:

```
dx% clear cluster <name> listen ssl clientauth cacertfile
```

To clear the CA CRL file for a cluster, type the command:

```
dx% clear cluster <name> listen ssl clientauth cacrlfile
```

To clear the trusted CA certificate file for a cluster, type the command:

```
dx% clear cluster <name> listen ssl clientauth catrustfile
```

To display configurations for the client authentication, type the command:

```
dx% show cluster <name> listen ssl clientauth
```

To display the configuration value for the advertise CA certificate file, type the command:

```
dx% show cluster <name> listen ssl clientauth cacertfile
```

To display the configuration value for the CA CRL file, type the command:

```
dx% show cluster <name> listen ssl clientauth cacrlfile
```

To display the configuration value for the trusted CA certificate file, type the command:

```
dx% show cluster <name> listen ssl clientauth catrustfile
```

### ***Browsers that Poorly Support SSL Client Authentication***

Certain browsers do not have stable SSL client authentication implementations and thus their interoperability with this feature is unpredictable and not recommended. The browsers that exhibit this behavior are:

- Netscape 4.x
- Opera



## Chapter 11

# Configuring Health Checking

This chapter describes how to configure health checking for the DX Application Acceleration Platform. It includes the following topics:

- “Configuring Layer 4 Health Check Settings” on page 221
- “Configuring Layer 7 Health Check Settings” on page 222
- “Viewing Health Check Configuration Settings” on page 223
- “Layer 7 Health Logging System Log Messages” on page 225
- “Notes on Layer 7 Health Checking” on page 226
- “Customizing Layer 7 Health Checking for SLB Services” on page 227
- “Scriptable Health Checking” on page 229

## Configuring Layer 4 Health Check Settings

---

Layer 4 health checking of clusters (specifically the connections between the DX appliance and target hosts) and forwarders is performed automatically, and cannot be disabled. You can configure some of the settings for the health checking.

To configure health checking parameters:

1. Specify the number of seconds between Layer 4 connection checks (1 to 3600). The default is 1 second.

```
dx% set forwarder <name> health connect interval <interval>
dx% set cluster <name> health connect interval <interval>
```

2. Specify the maximum number of seconds (1 to 60) that the DX appliance waits to establish a connection during a Layer 4 connection check.

```
dx% set forwarder <name> health connect timeout <1-60>
dx% set cluster <name> health connect timeout <1-60>
```

- Specify the number of consecutive failed health checks required (1 to 20) before the target host is marked as down. The default is 4 attempted health checks.

```
dx% set forwarder <name> health retry <1-20>
dx% set cluster <name> health retry <1-20>
```

- Save your configuration changes.

```
dx% write
```

## Configuring Layer 7 Health Check Settings

---

Layer 7 health checking occurs at the request level and can be configured for cluster services.

To configure L7 health checking for clusters:

- Specify the number of consecutive failed health checks required (1 to 20) before the target host is marked as down. The default is 4 attempted health checks.

```
dx% set cluster <name> health retry <1-20>
```

- Specify the number of seconds between Layer 7 connection checks (1 to 3600). The default is 1 second.

```
dx% set cluster <name> health request interval <interval>
```



**NOTE:** If you have both Layer 4 and Layer 7 health checking intervals defined, review the DX appliance policies for marking a target host up or down in Table 6 on page 47.

---

- Specify the maximum number of seconds (1 to 60) that the DX appliance waits for the last byte of the HTTP response, measured from the time that the Get request was sent. The default is 15 seconds.

```
dx% set cluster <name> health request timeout <1-60>
```

- Specify the number of health checks with good responses (1 to 20) that the DX appliance must receive before declaring the target host as operational. The default is one.

```
dx% set cluster <name> health request resume <1-20>
```

- Specify the URL path that the DX appliance sends to target servers for health checks. The URL path must begin with a forward slash (/) and does not include the domain name.

```
dx% set cluster <name> health request urlpath <url path>
```

- Specify the expected return code. The default is 200.

```
dx% set cluster <name> health request returncode <return code>
```



- Optionally specify the expected size of the response. This is the number of bytes in the body of the response as would be reflected in an HTTP Content-Length header. The default is -1 indicating this value is disabled or ignored.

```
dx% set cluster <name> health request size <size of response>
```

For a Web page, the size does not include embedded objects such as GIF or JPEG graphics, style sheets, javascript files, and so forth.

- Optionally, specify searches for a string in the non-header portion of the HTTP response. This option only applies to the following MIME types: text/html, text/css, text/plain, text/xml, and application/x-javascript.

```
dx% set cluster <name> health request string <string>
```

The `string` is case-sensitive, and the maximum length of the string is 64 bytes. Additionally, the string must be enclosed in double quotes if there is white space in the string.

- Specify the user agent for health check requests.

```
dx% set cluster <name> health request useragent <default | n>
```

- Enable L7 health checking on the cluster:

```
dx% set cluster <name> health request enabled
```

- Save your configuration changes.

```
dx% write
```

## Viewing Health Check Configuration Settings

---

You can view all of the configuration settings for a cluster or forwarder, or each of the individual settings separately, using the `show cluster/forwarder health` commands.

### Viewing All Settings

To view all Layer 4 forwarder health check settings, enter:

```
dx% show forwarder <name> health connect
Health Check Retry: 2
Health Check Connection Interval: 3600
Health Check Connection Timeout: 2
```

To view all Layer 4 cluster health check settings, enter:

```
dx% show cluster <name> health connect
Health Check Retry: 2
Health Check Connection Interval: 1
Health Check Connection Timeout: 2
```

To view all Layer 4 and Layer 7 cluster health check settings, enter:

```
dx% show cluster <name> health
Health Check Retry: 2
Health Check Connection Interval: 1
Health Check Connection Timeout: 2
Health Check Request Status: disabled
Health Check Request Interval: 15
Health Check Request Timeout: 15
Health Check Request Resume: 1
Health Check Request Url Path: /
Health Check Request Return Code: 200
Health Check Request Size: 0
Health Check Request String:
Health Check Request User Agent: Mozilla/4.0 (compatible; MSIE 5.5; Windows
NT
```

### **Viewing a Particular Setting**

To view the individual forwarder settings, enter one of the following:

```
dx% show forwarder <name> health connect interval
dx% show forwarder <name> health connect timeout
dx% show forwarder <name> health retry
```

To view the individual cluster settings, enter one of the following:

```
dx% show cluster <name> health retry
dx% show cluster <name> health request status
dx% show cluster <name> health [connect | request] interval
dx% show cluster <name> health [connect | request] timeout
dx% show cluster <name> health request resume
dx% show cluster <name> health request urlpath
dx% show cluster <name> health request returncode
dx% show cluster <name> health request size
dx% show cluster <name> health request string
dx% show cluster <name> health request useragent
```

The following example shows sample output for the health check status of a cluster

```
dx% show cluster <name> health request status
Health Check Status: enabled
Target Hosts:
[1] 66.218.71.87:80 Up
    Total:003 In Use:000 Hot:003 Cold:000 Discards:000
[2] 66.218.71.88:80 Layer 7 Down; Pending Change to Up
    Total:003 In Use:001 Hot:002 Cold:000 Discards:000
[3] 66.218.71.89:80 Layer 7 Down; Return Code Mismatch
    Total:003 In Use:000 Hot:002 Cold:001 Discards:000
[4] 66.218.71.90:80 TCP Layer Down; Unknown Reason
    Total:003 In Use:000 Hot:003 Cold:000 Discards:000
```

For information about interpreting the output, refer to the *Command Line Reference* manual.



**NOTE:** You can have the DX E-mail you an ALERT when a target host goes down. For additional information, refer to the event notification example “Receive Notification of L7 Health Check Errors via E-Mail” in “Administering Your DX Platform” on page 113.

## Layer 7 Health Logging System Log Messages

After Layer 7 health checking is enabled on a cluster, the system log will record messages about the health of the system. These system log messages are generated by default as soon as a user enables L7 health checking on a target.

Each messages contains a variety of information, including:

- date and time in [yyyy-mm-dd hh:mm:ss (timezone adjustment)] format
- level of message importance; alert, health
- indicator of whether this message is an inband (TSSN) or out-of-band (HEALTH) health check
- textual message
- IP address where failure was detected
- one or more communication protocols that caused the message to be logged and the number of failures seen on that protocol
- indicator of whether this message was reported from an inband (Override = 0) or out-of-band (Override = 1) connection

For example:

```
[2006-10-04 19:25:00 (+0700)][ALERT][TSSN][Cannot contact Target Host
10.80.10.10:443 TCP/UDP(1), Override(0).]
```

```
[2006-10-04 19:25:00 (+0700)][ALERT][TSSN][Cannot contact Target Host
10.80.10.10:443 TCP/UDP(1), SSL(2), Override(0).]
```

```
[2006-10-04 19:25:00 (+0700)][ALERT][TSSN][Cannot contact Target Host
10.80.10.10:443 TCP/UDP(1), SSL(2), HTTP(4), Override(0).]
```

```
[2006-10-04 19:25:00 (+0700)][ALERT][TSSN][Cannot contact Target Host
10.80.10.10:443 TCP/UDP(1), HTTP(4), Override(0).]
```

These messages were generated on October 4th, 2006 in the Pacific timezone. They were generated by an in-band health check on port 443 of the target host with address 10.80.10.10. The difference between the four log messages indicate what type of health checking was enabled.

The first of the messages tells us that connect health checking is enabled, the target host could not be contacted, and there was one TCP/UDP connection attempt that failed. The second message tells us that connect and SSL health checking are enabled, and there was one TCP/UDP connection attempt that failed and two SSL connections attempts that failed. The third message tells us that connect, SSL and request health checking is enabled, and there was one TCP/UDP connection attempt that failed, two SSL connection attempts that failed, and four request (HTTP) connection attempts that failed.

Some examples of successful health check messages:

```
[2006-10-04 19:25:41 (+0700)][ALERT][HEALTH][Health : Target Server
10.80.184.48:443 SSL connect health check succeeded.]
```

```
[2006-10-04 19:25:41 (+0700)][ALERT][HEALTH][Health : Target Server
10.84.0.112:80 TCP connect health check succeeded.]
```

```
[2006-10-04 19:25:41 (+0700)][ALERT][HEALTH][Health : Target Server
10.80.184.48:443 SSL connect health check succeeded.]
```

```
[2006-10-04 19:25:41 (+0700)][ALERT][HEALTH][Health : Target Server
10.84.0.112:80 TCP connect health check succeeded.]
```

## Notes on Layer 7 Health Checking

---

The DX assumes all target hosts are down and only logs state transitions. This means that with two servers to be checked when we turn-on L7 health checking (one down and one up), the server that is up will be logged in the system log as “Server A passed L7 Health Check” but the server that is down will never be mentioned in the logs until such time as it comes up.

For example:

- Server 0.0.31.20 is normal. It responds to both a ping and an HTTP request (machine is up, the Web server is up).
- Server 10.0.31.10 is in a semi-bad state. It responds to a ping, but not an HTTP request (machine is up, the Web server is down)

In this state, when L7 health checking is first enabled, you will never see 10.0.31.10 marked as “bad” by L7 health checking. This is because it was never seen as “up” by the DX, and therefore, there was never a transition to record.

If you are running a Web server with NTLM enabled, such as IIS, you will need to change the default health check code from 200 to 401 (“Access Denied”). Otherwise, the cluster target hosts will be marked as L7 down, and the cluster will not work.

Note the following steps:

```
dx% set cluster 3 health request returncode 200
(*) dx% write
Writing configuration.
Done.
```

```

dx% show cluster 3 target status
Target Hosts:
  [1] 10.0.22.22:80 Layer 7 Down; Return Code Mismatch
      Total:006 In Use:000 Hot:006 Cold:000 Discards:000

dx% set cluster 3 health request returncode 401
(*) dx% write
Writing configuration.
Done.

dx% show cluster 3 health request returncode
Health Check Return Code: 401

dx% show cluster 3 target status
Target Hosts:
  [1] 10.0.22.22:80 Up
      Total:008 In Use:000 Hot:006 Cold:002 Discards:000

```

This first shows the user setting the health returncode to 200 (the default value) for a cluster with an NTLM-enabled server in it. Note that the status of the cluster's target status shows that the target server is down. When the health return code is set to 401 ("Access Denied"), the cluster's target status then shows that the target server is up.

## Customizing Layer 7 Health Checking for SLB Services

---

In most situations, it is more appropriate to use the built-in DX L7 health check, however, it is still possible to have your SLB perform L7 health checking. If you want to use your SLB to perform L7, content-based health check on your target Web servers, you must assign each target host to its own cluster on the DX.

### One-to-one Cluster to Server Mapping

Typically, you would create a single cluster that contains a group of target Web servers as:

```

Cluster VIP      Target Hosts
-----
1.2.3.4:80  ->  target A, target B, target C, target D

```

However, this configuration will break SLB L7 health checking because the four target Web servers appear as a single server to the SLB. If the SLB detects an error, it has no way of knowing which server is down and would mark the whole cluster as down.

To use SLB L7 health checking, create a cluster for each target host on the DX. Note that each cluster requires a distinct IP port combination.

```

Cluster VIP      Target Hosts
-----
1.2.3.4:80  ->  target A
1.2.3.5:80  ->  target B
1.2.3.6:80  ->  target C
1.2.3.7:80  ->  target D

```

### **Conserving IPs with One-to-One Mapping**

If your Web servers use public addresses or you need to conserve IPs for some other reason, you can still use the one-to-one mapping as previously described. Instead of using a unique IP address for each cluster's VIP, you can give each cluster the same IP with a unique port.

Cluster VIP		Target Hosts
-----		-----
1.2.3.4:80	->	target A
1.2.3.4:81	->	target B
1.2.3.4:82	->	target C
1.2.3.4:83	->	target D

## Scriptable Health Checking

---

Scriptable Health Checking allows you to write Expect/TCL scripts that can dynamically pause and un-pause target hosts. For example, a script can be written to do an “HTTP GET” on a particular target host. If the HTTP result code is unexpected, the target host can be taken out of rotation. You import the script into the DX, configure it for execution, and execute it.

Scriptable Health Checking requires a license from Juniper Networks before it can be used. Contact your Juniper Networks Sales Representative for information.

### Expect/TCL Scripts

#### Capturing and Configuring Expect/TCL Scripts

You import the Expect/TCL scripts using the Command Line Interface. Once the script is imported, the DX appliance validates it by checking for syntax errors. The maximum size of a script is 1 MByte, and there is no restriction on the total number of scripts.



**NOTE:** The DX checks the script for correct syntax only, not for proper operation. It is possible to write a script that is syntactically correct, but that will produce errors or unexpected results during operation. Use discretion when coding Expect/TCL scripts.

---

The scripts can be configured to run once or execute at an interval. The DX allows you to delete Expect scripts that are not configured. You cannot edit expect scripts on the DX. There must be a minimum of 1MByte of free disk space for the capture to commence.

#### Run-Time Environment

The DX Application Acceleration Platform will not allow the script to damage or delete software running on the DX. The script may, however, purge its sandbox environment. If this does happen, new scripts may not run as designed until the sandbox environment is repaired.

The DX reports runtime script errors back to the user in the logs. The scripts are able to hard-pause, soft-pause, and un-pause hosts. The available hosts are target hosts of the type:

- HTTP and forwarder clusters
- SLB

Pausing and unpausing changes that are script generated are written to memory only, and will be lost across reboots.

Running scripts are killed:

- Upon a related configuration change (new IP address, port change, health configuration updated, etc.).



**NOTE:** This means that any scripts that are running will be killed and restarted based on the new configuration. This is captured as “Forced Termination” in the script statistics.

---

- When a scheduled script is executed by the DX (non-test mode), any existing scripts of the same name will be terminated.
- When writing a script, the script must use the following path as the first line:

```
#!/usr/bin/expect -f
```

### Sandbox Environment

Network communication is through an IP address assigned to the sandbox. The DX appliance does not allow file writes that are script invoked.

The following resource limitations apply when a script is executing:

- Total script size memory limit is 5 MB.
- Health scripts run at a lower priority than server processes to ensure that script doesn't take up CPU time when server processes are running.
- Only 32 pseudo-terminals are available; ptys(32).

This affects the expect command “spawn” which uses pseudoterminals (ptys) to launch the corresponding process. Since these are limited, this effectively means health scripts cannot run the spawn command more than 32 times. However, the same ptys are also used by other portions of the system (such as the Command Line Interface), so in practice this number is much less. In general scripts should minimize the amount of time they hold on to the pty to avoid this scenario and avoid launching many processes using the “spawn” command.

These binaries are provided in the sandbox:

- ping < host >
- ssh
- telnet
- openssl
- nslookup
- traceroute

For the ping command, the only command line argument allowed is < host > . For the remainder of the commands, all standard command line options are allowed.



## Scriptable Health Checking TCL API

The API commands used with Scriptable Health Checking are shown in Table 2.

**Table 2: API Commands for Scriptable Health Checking**

Command	Description
<code>rln_send_event -i ip -p port -e event -c class -m msg</code>	<ul style="list-style-type: none"> <li>■ “ip” is the IP address of the target host (required field). The IP address can be specified in either traditional dotted format (192.168.0.80) or in hexadecimal format (0xC0A80050).</li> <li>■ “port” is the port number of the target host. Default value is 0. A value of 0 means all ports. <b>Note:</b> When the port number is available, the port should be explicitly specified. This will result in better performance.</li> <li>■ “event” is the event to be generated. Valid values are up or down (required fields).</li> <li>■ “class” is the class of the event. Valid values are layer7, layer5, layer4, layer3, or none. Default class is none.</li> <li>■ “message” is the log message accompanying the event. The default message is empty.</li> </ul>
<code>rln_send_action -i ip -p port -a action -m msg</code>	<ul style="list-style-type: none"> <li>■ “ip” is the IP address of the target host (required field). The IP address can be specified in either traditional dotted format (192.168.0.80) or in hexadecimal format (0xC0A80050).</li> <li>■ “port” is the port number of target host. Default value is 0. A value of 0 means all ports.</li> <li>■ “action” is the suggested action. Values are hard-pause, soft-pause, or un-pause (required fields).</li> <li>■ “message” is the message accompanying the event. The default message is empty.</li> </ul>
<code>rln_send_log -l location -m msg</code>	<ul style="list-style-type: none"> <li>■ “location” decides the log destination. Only “health check” is supported as of this release. The default value is <b>healthcheck</b>.</li> <li>■ “message” is the message accompanying the event. The default message is empty.</li> </ul>
<code>rln_radius_auth -i ip -p port -k serverkey -t timeout -r retries -u username -w password</code>	<ul style="list-style-type: none"> <li>■ “ip” is the ip address of the RADIUS server.</li> <li>■ “port” is the port of the RADIUS server.</li> <li>■ “serverkey” is the client's RADIUS secret. The RADIUS server has a secret for every client ip.</li> <li>■ “timeout” is the time in seconds after sending a RADIUS request that the client waits for a response. If the response is not received within &lt; timeout &gt; seconds, the RADIUS request is resent.</li> <li>■ “retries” is the number of times the client resends the RADIUS request for a response before determining the server as down.</li> <li>■ “username” is the name of the user to be authenticated.</li> <li>■ “password” is the password of the user to be authenticated.</li> </ul>

**Table 2: API Commands for Scriptable Health Checking (continued)**

Command	Description
<code>rln_ldap_auth -i ip -p port -d admin_user_dn -s admin_password -b base_dn -a user_attribute -v version -c ca_cert_file -u username -w password</code>	<ul style="list-style-type: none"> <li>■ “ip” is the ip address of the LDAP server.</li> <li>■ “port” is the port of the LDAP server.</li> <li>■ “admin_user_dn” is the DN (Distinguished Name) of the admin user. This field is optional. When present, admin authentication is done before user authentication. When absent, admin authentication is not done and user authentication is done directly.</li> <li>■ “admin_password” is the password for the admin user. This field is optional, but it must be present when “admin_user_dn” is present.</li> <li>■ “base_dn” is the DN of the root of the tree in the LDAP database under which the LDAP search has to be done for the users.</li> <li>■ “user_attribute” is the name of the attribute uniquely identifying the users in the LDAP database.</li> <li>■ “version” is the LDAP version to be used.</li> <li>■ “ca_cert_file” is the name of the file containing trusted root ca certificates. This field is optional. When present, LDAP connection with the server is upgraded to a TLS connection before doing the authentication. When absent, LDAP connection with the server is in clear text. This field has to be provided to use LDAP over TLS.</li> <li>■ “username” is the name of the user to be authenticated.</li> <li>■ “password” is the password of the user to be authenticated.</li> </ul>

### ***The Expect/TCL Command Set***

The Expect/TCL commands are used to update target host status and send log messages. The DX has a subset of the TCL command set. This subset has only commands that are deemed safe and are needed for script writing. Commands such as `fork`, `exec`, and `filewrit` have been removed.

### Available TCL/Expect Commands

Table 3 shows the currently available TCL commands.

**Table 3: TCL Commands**

Command	Command	Command	Command	Command
Safe Base	eval	interp	proc	tcl_startOfPreviousWord
Tcl	exit	join	puts	tcl_wordBreakAfter
after	expr	lappend	re_syntax	tcl_wordBreakBefore
append	fblocked	lindex	read	tcltest
array	fconfigure	linsert	regexp	tclvars
auto_qualify	fileevent	list	registry	time
bgerror	filename	llength	regsub	trace
binary	flush	lrange	resource	udp_conf
break	for	lreplace	return	udp_peer
catch	foreach	lsearch	scan	unknown
clock	format	lset	set	unset
close	gets	lsort	socket	uplevel
concat	global	memory	split	upvar
continue	history	msgcat	string	variable
dde	http	package	subst	vwait
encoding	if	parray	switch	while
eof	incr	pkg::create	tcl_endOfWord	
error	info	pkg_mkIndex	tcl_startOfNextWord	

Table 4 shows the expect commands that are supported.

**Table 4: Supported Expect Commands**

Command	Command	Command	Command	Command
close	expect_after	match_max	send_error	timestamp
exit	expect_background	overlay	send_tty	trap
exp_continue	expect_before	parity	send_user	wait
exp_open	expect_tty	prompt1	sleep	
exp_pid	expect_user	prompt2	spawn	
exp_version	getpid	remove_nulls	strace	
expect	log_user	send	stty	

Table 5 shows the expect commands that are NOT supported.

**Table 5: Expect Commands that are Not Supported**

Command	Command	Command	Command	Command
debug	exp_internal	inter_return	interpreter	send_log
disconnect	fork	interact	log_file	system

## Logging and Statistics

The script generates information-level logs that are logged in a new health script log. ALERT system logs are generated for various failures. Some sample scenarios are:

- Cannot kill old scripts
- Cannot initialize the configuration
- Cannot setup for script launching
- Memory error
- Script terminates abnormally
- Periodic script is killed due to the next run interval
- Script configuration error
- Script launch error

Statistics are provided to report the state of each script. The following data points are available:

- Is the script running?
- The number of times a script has been launched.
- The number of times the script failed to start.
- The number of times a script failed after it started.
- The number of times script killed due forced termination (configuration change or the next script interval due).
- The number of successful runs.
- The last run (in UTC).
- The next run (in UTC).

Statistics can be cleared using the DXSHELL.

If you see too many “Force Termination” failures for a periodic script, it could mean that the periodic interval is too short. The script is not finishing in time, and is killed and restarted for the next run. Configuration changes will also kill currently running scripts that would also increment this statistic.

The statistics are only updated/captured for scripts automatically run by a scriptable health system. They are not updated for scripts that are run manually from the DXSHELL using the command `set health script <script_name> testrun`. This command was created so that operators can perform a test run of the script before adding the script for automatic running via the health system.

When the script is run manually, the operator can visually see whether or not the script ran successfully, so the statistics are not updated. The status for scripts that run automatically cannot be seen directly by the operator. Instead, the operator must query the statistics to see the status information.

## TCL UDP Extension

The TCL User Datagram Protocol extension (known as `tcludp`) provides commands to create and use a UDP socket. To use the extension, the script has to load the UDP package by adding “`package require udp`” in the `tcl/expect` file. Some useful UDP commands that are supported are:

`udp_open [port]`

`udp_open` will open a UDP socket. If `port` is specified the UDP socket will be opened on that port. Otherwise the system will choose a port and the user can use the `udp_conf` command to obtain the port number if required.

`udp_conf sock host port`

`udp_conf` in this configuration is used to specify the remote destination for packets written to this sock. You must call this command before writing data to the UDP socket.

`udp_conf sock [-myport] [-remote] [-peer] [-broadcast bool] [-ttl count]`

In addition to being used to configure the remote host, the `udp_conf` command is used to obtain information about the UDP socket.

- “myport” returns the local port number of the socket.
- “remote” returns the remote hostname and port number as set using the `udp_conf sock host port`.
- “peer” returns the remote hostname and port number for the packet most recently received by this socket.
- “broadcast [boolean]” UDP packets can listen and send on the broadcast address. For some systems, a flag must be set on the socket to use broadcast. With no argument, this option will return the broadcast setting. With a boolean argument, the setting can be modified.

- “ttl [count]” The time-to-live is given as the number of router hops the packet may do. For multicast packets this is important in specifying the distribution of the packet. The system default for multicast is 1 which restricts the packet to the local subnet. To permit packets to pass routers, you must increase the ttl. A value of 31 should keep it within a site, while 255 is global.

```
udp_conf [-mcastadd groupaddr]
udp_conf [-mcastdrop groupaddr]
```

tcludp sockets can support IPv4 multicast operations. To receive multicast packets the application has to notify the operating system that it should join a particular multicast group. These are specified as addresses in the range 224.0.0.0 to 239.255.255.255.

```
udp_peek sock [buffersize]
```

Examines a packet without removing it from the buffer. This function is not available on windows.

Command line arguments enclosed in [square brackets] are optional.

## Scriptable Health Checking Commands

Use these commands to configure scriptable health checking:

### Configuration Commands

To capture a script, type the command:

```
dx% import health script <scp or tftp path>
```

The maximum script name length is 64 characters.

To add the script, type the command:

```
dx% add health script <script_name>
```

To enable or disable the script, type the command:

```
dx% set health script <script_name> <enabled | disabled>
```

To set the script vip, type the command:

```
dx% set health script <script_name> vip <vip>
```

For this command, the DX determines the most appropriate interface to alias the IP address.

To set the script execute interval, type the command:

```
dx% set health script <script_name> interval <value>
```

If a zero is set, the script will only run once. A value greater than zero specifies the run interval in seconds. The maximum value is 86400 seconds.

To perform a test run of the health script, type the command:

```
dx% set health script <script_name> testrun
```

This command allows you to test drive a health script and visually inspect the results to see if the script is behaving properly. You can put debug messages to trace your logic and check the health logs to see if the health check status is being communicated properly by the script. When the script finishes (it might not finish if it's a run once script), you can see the exit status to see if it ran successfully. Once you are comfortable with this, you can enable the script for automatic execution by the scriptable health system.

To delete a health script configuration node, type the command:

```
dx% delete health script <script_name>
```

To delete a script file, type the command:

```
dx% delete file <script_name>
```

## Show Commands

To show the configuration of Scriptable Health Checking, use the commands:

```
dx% show health script <script_name | all> interval
dx% show health script <script_name | all> vip
dx% show health script <script_name | all> name
dx% show health script <script_name | all> status
dx% show health script <script_name | all> stats
```

To clear the Scriptable Health Checking statistics, type the command:

```
dx% clear health script <script_name | all> stats
```

## Logging Commands

To show the health script log, type the command:

```
dx% show log health script
```

To clear the health script log, type the command:

```
dx% clear log health script
```

To export the health script log, type the command:

```
dx% export log health script <destination>
```

## Capture and Configuration Example

This is an example of how to capture and configure a script:

```
dx% import health script tftp://qa/scripts/foo.exp
done. 255 bytes transferred.
dx% add healthscript foo.exp
added healthscript foo.exp.
(*) dx% set health script foo.exp vip 192.168.14.75
(*) dx% set health script foo.exp interval 10
```

```
(*) dx% set health script foo.exp enabled
(*) dx% write
Writing configuration.
Done.
```

## Sample Scripts

This sample script sends a L7 down event:

```
#!/usr/bin/expect -f

package require RlnTclExt
set thost 192.168.14.221
set port 80
if { $argv == "debug" } {
    set dbgflag -d
}

#Host $thost is down.
rln_send_event -i $thost -p $port -e down -c layer7 -m "http GET failed"
This sample script sends a hardpause message:
#!/usr/bin/expect -f

package require RlnTclExt
set thost 192.168.14.221
set port 80
if { $argv == "debug" } {
    set dbgflag -d
}

#Host $thost is down.
rln_send_action -i $thost -p $port -a hardpause
```

## Health Check Settings

Turn-on SMTP health checking with the command:

```
dx% set slb group <name | all> healthcheck smtp enabled
```

Turn-off SMTP health checking with the command:

```
dx% set slb group <name | all> healthcheck smtp disabled
```



## Chapter 12

# Configuring ActiveN

This chapter describes how to configure ActiveN for the DX Application Acceleration Platform. It contains the following topics:

- “Overview” on page 239
- “Configuring the ActiveN Service” on page 239
- “Modifying Your ActiveN Configuration” on page 242
- “Sample ActiveN Configuration” on page 243
- “Configuring Advanced Settings for the ActiveN Service” on page 244

### Overview

---

An active-active topology is one in which two DX appliances are configured to actively process client traffic and load balance client requests. When one DX becomes unavailable, the other DX takes over its duties. ActiveN is an active-active topology with more than two DX appliances connected together. ActiveN provides truly linear scalability in terms of performance.

The ActiveN service is commonly configured for failover as well. Instructions for configuring ActiveN failover can be found in Chapter 20, “Configuring Failover” on page 357.

### Configuring the ActiveN Service

---

The ActiveN service must be configured on each DX appliance that you want to include in the multi-appliance topology. ActiveN provides load balancing across all of the appliances.

Each appliance in the ActiveN group is configured in the same manner. ActiveN automatically determines the master appliance, unless you specify otherwise using the *forcemaster* attribute. All DXs must have the same settings for forcemaster (all enabled or all disabled) for ActiveN to work, so use caution if you modify it. Many other details are also automatically negotiated by the DX appliances.

ActiveN requires the Direct Server Return (DSR) feature to be enabled on all DX appliances in the ActiveN group. You do not need to enable DSR on your Web servers. This setting is transparent to your target hosts. For additional information on DSR, refer to “Integrating the DX Appliance into a Direct Server Return (DSR) Environment” on page 154.

The following assumptions apply to the configuration instructions contained here:

- The DX appliances are on the same subnet and are meant to be used with one another in a single group.
- Ether 0 is the network interface on each DX.

To configure the ActiveN service using DXSHELL:

1. Create and populate your cluster.

```
dx-1% add cluster
Created cluster 1
(*) dx-1% set cluster 1 listen vip 10.10.10.25
(*) dx-1% set cluster 1 listen port 495
(*) dx-1% set cluster 1 target host 192.168.70.2:24
(*) dx-1% set cluster 1 target host 192.168.51.100:32
...
(*) dx-1% write
Writing configuration.
Done.
dx-1%
```



**NOTE:** A warning message appears indicating the need to configure DSR:

```
WARNING: The following clusters/forwarders are in ActiveN with DSR not
enabled!!
Cluster <vip address> conflicts with ActiveN group <n>.
```

If DSR is not enabled for these, it can result in IP conflicts on your network!!

Continue with these steps to remedy this issue.

---

2. Enable the Direct Server Return feature.

```
dx% set cluster 1 dsr enabled
```

3. Create an ActiveN group. This group contains all of the DXs that you will use together in the ActiveN configuration. Link your cluster and ActiveN group by using the IP and port settings you used for the cluster in Step 1.

```
(*) dx-1% add activen group
add activeN group [name] <ip:port>
(*) dx-1% add activen group ang1 10.10.10.25:495
group ang1 created
(*) dx-1% write
Writing configuration.
Done.
dx-1%
```

4. Add the DX appliances to the ActiveN group. ActiveN calls each machine in the ActiveN group a *blade*. First, you create these blades, and then you assign them to your ActiveN group.

- a. Create blades by referring to the real IP address of Ether 0 on the machine you want to add to the ActiveN group. Repeat this step for each machine you want to add:

```
dx% add activeN blade 192.168.0.0
(*) dx% add activeN blade 192.168.10.0
...
```

- b. Assign the blades to your ActiveN group. You can assign them individually or all at once to a specific ActiveN group, or to all ActiveN groups. This example assigns all blades to all ActiveN groups.

```
(*) dx% set activeN group all blade all
```

5. Repeat Step 3 and Step 4 to add additional groups. For example, you might add a group and configure an SSL cluster on each DX to serve HTTPS requests.
6. Start the ActiveN service.

```
(*) dx-1% set activeN enabled
Unified Failover will start the process.
(*) dx% write
Writing configuration.
Done.
```

7. Verify your configuration.

```
dx% show activeN group ang1
Group ang1
Vip: 10.10.10.25
Port: 495
Interface: ether0
Sticky: disabled
Pause: none
HealthCheck Parameters:
Up Timeout: 4sec
Down Timeout: 4sec
Syn Timeout: 2sec
Max Tries: 1
Session Timeouts:
Active: 100sec
Closewait: 25sec
Ackwait: 10sec
Advanced:
Switching Policy: default
SynFlood Protect: default
Reset To Client: default
Reset To Server: default
BurstMax: 7000
Total Blades: 0
Active Blades: 0

Blades:
Index Status Pause Local Real IP Mac
```

8. (Optional) Configure failover on each DX appliance. Refer to Chapter 20, “Configuring Failover” on page 357.
9. Repeat Step 1 through Step 7 for each DX appliance in your ActiveN configuration.



The cluster and ActiveN configuration parameters must be identical on all of the DX appliances, except for the node ID.

---

## Modifying Your ActiveN Configuration

---

Because ActiveN has both load balancing and failover components, if you would like to disable ActiveN while you are making configuration changes, you must first disable failover for the ActiveN service and then disable the service itself:

```
dx% set failover disabled
dx% set activeN disabled
```

You may also want to stop your server while you are making changes:

```
dx% set server down
```



You can make your changes on the DX appliances acting as slaves first. Then, force a failover, and make changes to the remaining DX appliance (the one that used to be the master). Use the `show activeN failover` command to verify which DX appliance is acting as master.

---

Finally, it is important to note that if you make changes to one DX’s ActiveN configuration, you must ensure that you make that change to ALL the DXs you are using for that ActiveN group.

After you are finished, re-enable ActiveN load balancing and failover:

```
dx% set failover enabled
dx% set activeN enabled
```

If you stopped your server, remember to bring it back up again:

```
dx% set server up
```



**NOTE:** A warning message appears when you re-enable failover if the VIP for the activeN group conflicts with the VIP for an SLB group.

```
WARNING: The following vip conflicts were detected!!
ActiveN group <name> vip <vip> conflicts with SLB group <name>
```

We recommend resolving these conflicts to avoid potential IP conflicts on your network.

If you see this message, verify your activeN group and SLB group VIPs and reconfigure accordingly.

---

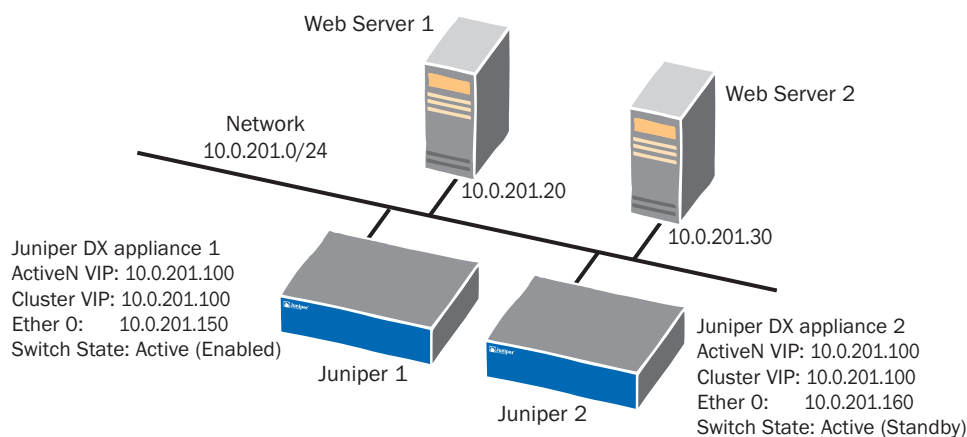
## Sample ActiveN Configuration

Figure 55 shows an example of an ActiveN configuration, along with the commands needed to set up the configuration. This example has two DXs. The network has the IP address map shown in Table 6.

**Table 6: Network IP Address Mapping**

DX Appliance	Port	IP Address
Web Server 1		10.0.201.20:80
Web Server 2		10.0.201.30:80
Juniper DX Appliance 1	Ether 0	10.0.201.150
	Ether 1	10.0.201.151
Juniper DX Appliance 2	Ether 0	10.0.201.160
	Ether 0	10.0.201.161
ActiveN VIP		10.0.201.100
Cluster VIP		10.0.201.100

**Figure 55: An Example ActiveN Configuration**



### DX Appliance 1 Configuration

The first DX appliance shown in Figure 55 is configured as follows:

#### Cluster Configuration

```
dx% add cluster 1
dx% set cluster listen vip 10.0.201.100
dx% set cluster 1 target host 10.0.201.20:80
dx% set cluster 1 target host 10.0.201.30:80
dx% set cluster 1 dsr enabled
dx% write
```

**ActiveN Configuration**

```

dx% add activen group 1 10.0.201.100:80
dx% add activen blade 10.0.201.150
dx% add activen blade 10.0.201.160
dx% set activen group 1 blade 10.0.201.150
dx% set activen group 1 blade 10.0.201.160
dx% set activen enabled
dx% write

```

**DX Appliance 2 Configuration**

The second DX appliance shown in Figure 55 is configured as follows:

**Cluster Configuration**

```

dx% add cluster 1
dx% set cluster listen vip 10.0.201.100
dx% set cluster 1 target host 10.0.201.20:80
dx% set cluster 1 target host 10.0.201.30:80
dx% set cluster 1 dsr enabled
dx% write

```

**ActiveN Configuration**

```

dx% add activen group 1 10.0.201.100:80
dx% add activen blade 10.0.201.150
dx% add activen blade 10.0.201.160
dx% set activen group 1 blade 10.0.201.150
dx% set activen group 1 blade 10.0.201.160
dx% set activen enabled
dx% write

```



Remember to configure failover for the ActiveN service on both DX appliances. See Chapter 20, “Configuring Failover” on page 357.

---

**Configuring Advanced Settings for the ActiveN Service**

A variety of commands are available to more finely tune your ActiveN configuration. This section describes the following commands:

- “Set Commands” on page 245
- “Delete Commands” on page 248
- “Clear Commands” on page 249
- “Show Commands” on page 249

## Set Commands

Set commands are used to set configuration parameters.

### Global Configuration Commands

This command is used to turn-on or turn-off the ActiveN feature by setting the switch state.

```
dx% set activeN <enabled|disabled>
```

The switch can be set in one of two states:

- Enabled: The switch is active.
- Disabled: The switch is stopped.

This command is used to set the cleaning interval (the interval at which two repeat cycles are spaced):

```
dx% set activeN cleaning_interval <secs>
```

Since the ActiveN switch works in DSR mode, it does not see the packets going from blade to the client. This makes it difficult for the ActiveN switch to track the connection state. Instead, it uses a timer to purge the sessions.

```
dx% set activeN session timeout active <secs>
dx% set activeN session timeout closewait <secs>
dx% set activeN session timeout ackwait <secs>
```

If a session has not been active for a period of time, it is purged in the timer. The three possible conditions are:

- Active: The session that is in active session.
- Closewait: The session the client has terminated from its side.
- Ackwait: The 3-way TCP handshake not completed.

This command is used to set the maximum allowable blades in the system.

```
dx% set activeN max_blades <number>
```

This command needs to be run before starting the ActiveN switch.

```
dx% set activeN advanced policy <roundrobin|leastconn>
```

This is used to set the switching policy to either round robin or least connection.

This command is used to enable protection against syn flood. Since the ActiveN operates in DSR mode, it cannot track if the 3-way TCP handshake completed successfully, but it needs to remember the session information for such sessions. In order to protect itself from an attack, the ActiveN purges a connection if the client does not send final acknowledge for the handshake.

```
dx% set activeN advanced synflood_protect <enabled|disabled>
```

Since all the sessions are purged in the timer routine, we can set the maximum number of timed out sessions to be purged in one timer interval.

Setting `burst_max` to zero will cause all the sessions that have timed out to be purged in timer cycle.

```
dx% set activeN advanced burst_max <number>
```

This command is used to enable or disable the sending of resets to the client. When active sessions are purged, a reset can be sent to the client and to the server to indicate the connection has been terminated.

```
dx% set activeN advanced reset client <enabled|disabled>
```

This command is used to enable or disable sending of resets to the server (“blade”).

```
dx% set activeN advanced reset server <enabled|disabled>
```

Many of the global configuration commands may also be set on an ActiveN group basis.

### Set Group Commands

This command is used to set a blade as a member of a group. Using the keyword “all” in the group argument results in the blade being added to all the groups, and using “all” in the blade argument results in adding all the blades into the group.

```
dx% set activeN group <name|all> blade <ip_addr|all>
```

This command is used to set the Client IP Sticky, which is where the load balancer chooses the same server for multiple TCP connections when the subsequent requests come from the same client IP address. Refer to “Client IP Sticky” on page 59 for additional information.

```
dx% set activeN group <name|all> blade sticky <enabled|disabled>
```

This command is used to set the timeout value for the Client IP Sticky feature. The default value is 120 minutes, the minimum is one minute, and the maximum is 30 days.

```
dx% set activeN group <name|all> blade sticky timeout <minutes>
```

### Health Check Commands

Periodic health checks of the blades are conducted for status of the blades. The following commands set the parameters associated with the health checking. Note that Health check is a default feature and it cannot be turned-off. Many of these health check configuration commands may also be set on an ActiveN group basis.

These commands are used to set the time duration between two health checks.

```
dx% set activeN healthcheck interval up <secs>
dx% set activeN healthcheck interval down <secs>
dx% set activeN healthcheck interval syn <secs>
```



Intervals between two health checks are defined for each different status of the blades.

- **up**: The blade has responded to the health check probe.
- **down**: The blade has not responded to the probe and has been taken out of rotation.
- **syn**: Time gap between sending two consecutive health probes, if no response is received.

This command is used to set the maximum number of health check tries before giving up.

```
dx% set activeN healthCheck maxtries <Number>
```

The default values for each of these parameters are:

- Up: 45 seconds
- Down: 20 seconds
- Syn Wait: 10 seconds
- Maxtries: 3

The default values for health checking are not optimum for all DXs. Using the ActiveN default health checking parameters can cause the ActiveN “active” unit to forward traffic to non-healthy blades. A non-healthy blade is a cluster with “all” target hosts down. As long as there is at least one target host up in a cluster; then ActiveN will consider that cluster/blade a healthy blade.

#### ***Worse Case Scenario for ActiveN Forwarding Traffic to a Non-Healthy Blade***

With the default health checking parameters (Up: 45, Down: 20, Syn wait: 10, and Maxtries: 3), ActiveN will distribute traffic to non-healthy blades for a period of up to 1 minute, 15 seconds (the “worse case” scenario).

#### ***Best Case Scenario for ActiveN Forwarding Traffic to a Non-Healthy Blade***

In the “best case” scenario in terms of time, traffic will still be forwarded to a non-healthy blade for a period of at least 20 seconds.

In some applications this is too long of a period of time to be forwarding traffic into a blackhole, and the default ActiveN health checking parameters are not aggressive enough. The health check parameters are configurable for applications that need quicker health checking results.

**Suggested Values**

A more aggressive ActiveN health checking parameters configuration might be:

- Up: 4 seconds
- Down: 4 seconds
- SYN wait: 2 seconds
- Maxtries: 1 seconds

Keep in mind that these configurable numbers can even be lower. However, in a healthy environment three packets (SYN, SYN ACK, and RST) are continuously exchanged between the ActiveN active unit and the blades that it is health checking. This packet exchange can become very chatty in ActiveN scenarios where there are multiple ActiveN blades and multiple ActiveN groups.

**Set Client IP Sticky Commands**

To enable Client IP Stickyness, type the command:

```
dx% set activeN group <name> sticky enabled
```

To disable stickyness, type the command:

```
dx% set activeN group <name> sticky disabled
```

The default value of sticky is disabled.

To set the timeout of sticky entries, type the command:

```
dx% set activeN sticky timeout < minutes>
```

The default timeout is 120 minutes, the minimum is 1 minute, and the maximum is 30 days.

These commands require a role of Administrator or Network Administrator before they can be executed. A `write` is required in order for the changes to take effect.

**Delete Commands**

This command is used to delete a group specified by name. Using `all` will delete all groups.

```
dx% delete activeN group <name|all>
```

This command is used to delete a blade specified by an index. Using `all` will delete all blades.

```
dx% delete activeN blade <ip_addr|all>
```

## Clear Commands

This command is used to disassociate a blade from a group. Using all will remove all the blades from the groups.

```
dx% clear activeN group <name|all> blade <index|all>
```

This command is used to clear the statistics for a group.

```
dx% clear activeN group <name|all> stats
```

This command is used to clear the statistics for a blade.

```
dx% clear activeN blade <name|all> stats
```

This command is used to clear overall statistics.

```
dx% clear activeN total stats
```

## Show Commands

This command is used to display the group characteristics. Using all will display all of the groups.

```
dx% show activeN group <name|all>
```

This command is used to display the blade characteristics. Using all will display all of the blades.

```
dx% show activeN blade <ip|all>
```

This command is used to display the overall statistics for the switch.

```
dx% show activeN stats
```

The ActiveN statistics are cumulative for all running ActiveN groups. The statistics displayed are shown in Table 7.

**Table 7: ActiveN Statistics**

Statistic	Description
<b>Total Statistics</b>	
Bytes	The total byte count received by all clients.
Packets	The total number of packets received by all clients.
Flushed	The total number of connections that have been flushed by ActiveN. Once the DX appliance receives a RST or a FIN from the client for an active connection, it then waits a number of seconds, and flushes the connection. The counter is then incremented.
syn	The total number of SYNs sent by all clients.
rst	The total number of RSTs sent by all clients.
fin	The total number of FINs sent by all clients.

**Table 7: ActiveN Statistics (continued)**

Statistic	Description
<b>Current Sessions</b>	
Active	The current number of established TCP sessions.
Fin	The current number of FINs sent by the client prior to ActiveN flushing.
Reset	The current number of RSTs sent by the client prior to ActiveN flushing.

Many of these statistics can also be displayed for ActiveN groups.

Troubleshooting ActiveN problems depends upon the nature of the problem that is occurring. For instance, if the “active” session count is high and increasing, but the “flushed” count is low and not increasing, this implies that there are either slow clients/target hosts, or a high latency on transactions with the DXs.

By knowing what these values mean, you can keep track of what is going on in your site (primarily from the client side to the DX). Dividing these numbers by time can give you the average occurrence count for each variable in the ActiveN stats.

This command is used to display the basic configuration parameters.

```
dx% show activeN
```

This command is used to display advanced configuration parameters.

```
dx% show activeN advanced
```

This command is used to display the switch state.

```
dx% show activeN status
```

This command is used to display the Client IP Sticky timeout entries.

```
dx% show activeN sticky timeout
```

For complete information on each of these commands, refer to the *Command Line Reference* manual.

## Chapter 13

# Setting up the DX Appliance for “Sticky” Traffic

This chapter describes setting up the DX for sticky traffic on the DX Application Acceleration Platform, discussing the following topics:

- Overview on page 251
- Configuration Instructions for Cookie-Based Client Stickiness on page 251
- Configuration Instructions for Client IP-Based Stickiness on page 252

## Overview

---

“Sticky” is the common term used to describe Web client requests being redirected to the same target host within a cluster. The client “sticks” with the server.

To configure the DX to create sticky connections between clients and target servers, simply specify on a cluster-by-cluster basis whether you want the cluster to use cookie or client-IP based stickiness, and then set a timeout value for the sticky connection.

To disable client stickiness, use the following commands:

```
dx% set cluster <name> sticky method none
```

The none option is the default setting on the DX.

## Configuration Instructions for Cookie-Based Client Stickiness

---

To choose cookie-based stickiness, use the following commands:

```
dx% set cluster <name> sticky method cookie  
dx% set cluster <name> sticky cookie expire [0-3000000]
```

The allowable range of cookie expire values is 1 minute to 3,000,000 minutes (5.71 years). Setting the cookie expire value to 0 means that the cookies never expire.

To display the sticky settings for this cluster:

```
dx% show cluster <name> sticky
```

To disable cookie-based client stickiness, set the sticky method to none.

```
dx% set cluster <name> sticky method none
```

Sticky cookies have been known to break some server applications. An option has been added to remove the sticky cookie from the request headers based upon the configuration. When there are multiple cookies in a “Single Cookie” header, it only strips the sticky cookie.

To control whether the cookie is stripped, type the command:

```
dx% set cluster <name> sticky passheader [disabled | enabled*]
```

When enabled, the cookie is passed through; when disabled, the cookie is stripped:

## Configuration Instructions for Client IP-Based Stickiness

---

To choose client IP-based stickiness, use the following commands:

```
dx% set cluster <name> sticky method clientip
dx% set cluster <name> sticky clientip timeout [1-43200]
```

The range of timeout values is 1 minute to 43200 minutes (30 days).

You can select the appropriate method for hashing depending upon whether the DX is deployed in front of a public Web site or in front of an intranet site. The command used to set this parameter is:

```
dx% set cluster <name> sticky clientip distribution <internet | intranet>
```

For optimum performance for a public Web site, set to “internet” using the command:

```
dx% set cluster <name> sticky clientip distribution internet
```

For optimum performance for an intranet Web site, set to “intranet” using the command:

```
dx% set cluster <name> sticky clientip distribution intranet
```

To display the sticky settings for this cluster:

```
dx% show cluster <name> sticky
```

To disable clientIP-based client stickiness, set the sticky method to none:

```
dx% set cluster <name> sticky method none
```

## Chapter 14

# Configuring HTTP(S) Authentication

This chapter provides instructions for configuring HTTP(S) Authentication for the DX Application Acceleration Platform. You must have a valid AFE or Advanced AFE license installed on your DX appliance to use this feature.

This chapter covers the following topics:

- “Authentication Commands” on page 253
- “Configuring Basic Authentication Parameters” on page 256
- “Configuring Authentication Using an LDAP Server” on page 257
- “Configuring Authentication Using Microsoft Active Directory as the LDAP Server” on page 260
- “Configuring Authentication Using RSA SecurID” on page 261
- “Logging Into Multiple Web Applications within a Domain” on page 263

## Authentication Commands

---

User authentication is configured and monitored through the DXSHELL using `set`, `show`, and `clear` commands. You must be logged in as a user with either `security_administrator` or `administrator` access permissions. Using the on-board cache to hold entries for authentication improves application performance for the client.

For more information about:

- User roles, see “Multi-Level Administrative Rights” on page 26
- Authentication commands, refer to the *Command Line Interface for DXOS*.

## Set Commands

set Command
-------------

<b>General</b>
----------------

<code>set cluster &lt;name&gt; aaa authentication [enabled disabled*]</code>
------------------------------------------------------------------------------

<code>set cluster &lt;name&gt; aaa authentication method www</code>
---------------------------------------------------------------------

<b>set Command</b>
set cluster <name> aaa authentication realm <string>
set cluster <name> aaa authentication response text <string>
set cluster <name> aaa authentication protocol [RADIUS LDAP]
set cluster <name> aaa authentication password empty_allowed [enabled disabled*]
set cluster <name> aaa authentication password maxage <integer>
set cluster <name> aaa authentication password maxlength <integer>
set cluster <name> aaa authentication redirect protocol [http* https]
set cluster <name> aaa authentication redirect host <ip>
set cluster <name> aaa authentication redirect url <url>
<b>RADIUS</b>
set cluster <name> aaa authentication radius server <name> ip <IP addr>
set cluster <name> aaa authentication radius server <name> port <port number>
set cluster <name> aaa authentication radius server key <shared-key>
set cluster <name> aaa authentication radius server timeout <integer>
set cluster <name> aaa authentication radius server retries <integer>
<b>LDAP</b>
set cluster <name> aaa authentication ldap version <integer> /* Default LDAPv3 */
set cluster <name> aaa authentication ldap server <name> ip <IP_addr>
set cluster <name> aaa authentication ldap server <name> port <Port>
set cluster <name> aaa authentication ldap server <name> type <NDS IPLANET ADS>
set cluster <name> aaa authentication ldap base-dn <string>
set cluster <name> aaa authentication ldap anonymous [enabled disabled]
set cluster <name> aaa authentication ldap bind user-dn <string>
set cluster <name> aaa authentication ldap bind password <string>
set cluster <name> aaa authentication ldap uid <string>
set cluster <name> aaa authentication ldap gid <string>
set cluster <name> aaa authentication ldap ssl [enabled disabled*]
set cluster <name> aaa authentication ldap ssl cacertfile <cert file>
set cluster <name> aaa authentication ldap ssl uri <LDAPS URI>
<b>Audit</b>
set cluster <name> aaa audit [enabled* disabled]
set cluster <name> aaa audit level [all failures]
<b>Cache</b>
set cluster <name> aaa authentication cache [enabled*   disabled]
set cluster <name> aaa authentication cache maxage [maxage]



## Show Commands

<b>show Commands</b>
<b>General</b>
show cluster <name> aaa authentication method
show cluster <name> aaa authentication response
show cluster <name> aaa authentication password empty_allowed
show cluster <name> aaa authentication password maxage
show cluster <name> aaa authentication password maxlength
show cluster <name> aaa authentication redirect protocol
show cluster <name> aaa authentication redirect host
show cluster <name> aaa authentication redirect url
show cluster <name> aaa authentication
<b>RADIUS</b>
show cluster <name> aaa authentication radius server
show cluster <name> aaa authentication radius server <name>
show cluster <name> aaa authentication radius timeout
show cluster <name> aaa authentication radius retries
show cluster <name> aaa authentication radius key
show cluster <name> aaa authentication radius
<b>LDAP</b>
show cluster <name> aaa authentication ldap
show cluster <name> aaa authentication ldap version
show cluster <name> aaa authentication ldap protocol
show cluster <name> aaa authentication ldap server
show cluster <name> aaa authentication ldap server <name>
show cluster <name> aaa authentication ldap server type
show cluster <name> aaa authentication ldap base-dn
show cluster <name> aaa authentication ldap bind
show cluster <name> aaa authentication ldap uid
show cluster <name> aaa authentication ldap gid
show cluster <name> aaa authentication ldap anonymous
show cluster <name> aaa authentication ldap ssl
show cluster <name> aaa authentication ldap ssl cacertfile
show cluster <name> aaa authentication ldap ssl uri
<b>Audit</b>
show cluster <name> aaa audit
show cluster <name> aaa
<b>Cache</b>
show cluster <name> aaa authentication cache
show cluster <name> aaa authentication cache status

**show Commands**

```
show cluster <name> aaa authentication cache maxage
show authentication cache stats
```

**Clear Commands****clear Commands****General**

```
clear cluster <name> aaa authentication response text
clear cluster <name> aaa authentication redirect protocol
clear cluster <name> aaa authentication redirect host
clear cluster <name> aaa authentication redirect url
```

**RADIUS**

```
clear cluster <name> aaa authentication radius server 1 ip
clear cluster <name> aaa authentication radius server 2 ip
clear cluster <name> aaa authentication radius realm
clear cluster <name> aaa authentication radius server key
```

**LDAP**

```
clear cluster <name> aaa authentication ldap server 1 ip
clear cluster <name> aaa authentication ldap server 2 ip
clear cluster <name> aaa authentication ldap base-dn
clear cluster <name> aaa authentication ldap uid
clear cluster <name> aaa authentication ldap gid
clear cluster <name> aaa authentication ldap ssl cacertfile
```

**Cache**

```
clear authentication cache
```

**Configuring Basic Authentication Parameters**

The following configuration steps apply to all authentication servers that communicate with the DX appliance. Configure these parameters first, then configure the LDAP or RADIUS parameters.

To configure basic authentication parameters on the DX appliance:

1. Specify the authentication method:

```
dx% set cluster secure_cluster_001 aaa authentication method www
```

2. Specify the realm name to be displayed in the login pop-up dialog box:

```
dx% set cluster secure_cluster_001 aaa authentication realm
    juniper
```

3. Specify the authentication HTML message used for the cluster.

```
dx% set cluster secure_cluster_001 aaa authentication response
text "You are not authorized to access this page."
```

4. Specify the authentication protocol to use for the cluster:

```
dx% set cluster secure_cluster_001 aaa authentication protocol
LDAP
```

5. Optionally, allow empty (null) passwords for cluster authentication (disabled by default). By default, AAA authentication fails if the password has a null value.

```
dx% set cluster secure_cluster_001 aaa authentication password
empty_allowed enabled
```

6. Optionally, specify the maximum number of days a password can be used, between 1 and 365 days:

```
dx% set cluster secure_cluster_001 aaa authentication password
maxage 180
```

7. Optionally, specify the maximum number of characters a password may have:

```
dx% set cluster secure_cluster_001 aaa authentication password
maxlength 20
```

8. Optionally, configure where to redirect a client when a password is to be changed:

- a. Specify the protocol to use when retrieving the password change custom page. The default is http.

```
dx% set cluster secure_cluster_001 aaa authentication
redirect protocol https
```

- b. Specify the URL where the LDAP server or Active Directory server sends a password change flag. The default is "/auth.shtml".

```
dx% set cluster secure_cluster_001 aaa authentication
redirect url /pswd_chng.company.com/pswdchng.html
```

- c. Specify the remote host from where this URL is retrieved. By default, the file is local, and the host is the IP address of the cluster.

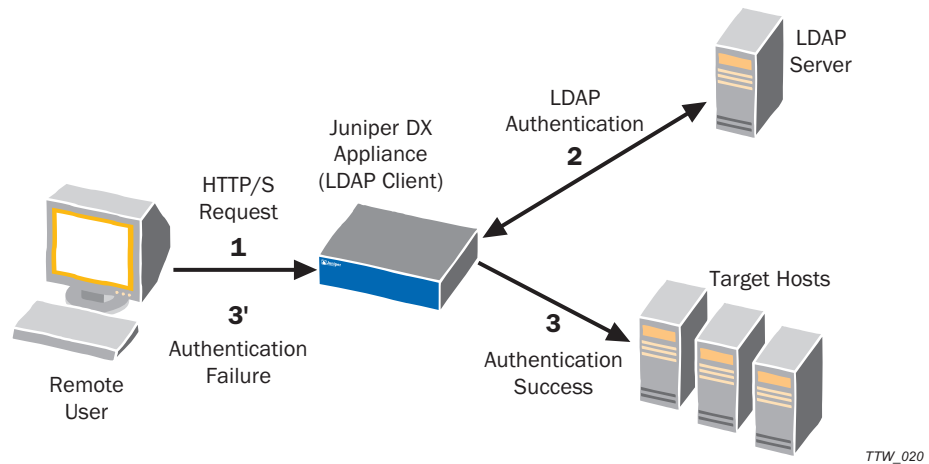
```
dx% set cluster secure_cluster_001 aaa authentication
redirect host 192.168.72.3
```

## Configuring Authentication Using an LDAP Server

---

Lightweight Directory Access Protocol (LDAP) is a client-server protocol for accessing directory services. LDAP servers can be used as a focal point for user authentication over the network. LDAP provides authentication using the user data stored in an LDAP server.

A typical authentication configuration for an Enterprise includes LDAP v.3 directory software, one DX, and a remote user (typically a browser connected to a network). See Figure 56. The authenticating DX supports LDAP authentication as a client.

**Figure 56: Using the DX with LDAP for Authentication**

The numbers shown in Figure 56 indicate the authentication process flow: (1) A client makes a request for information on a target host; (2) The DX receives the client request and makes an authentication request to the LDAP server; (3) Authentication is successful and the DX obtains the requested information and accelerates it back to the client, or (3') Authentication is not successful and the DX returns an authentication error to the client.

To configure LDAP authentication, follow these general steps:

1. Configure your LDAP server to receive HTTP/S requests from remote clients.

Refer to the *LDAP v.3 Configuration Guide* provided by your LDAP software vendor. An excellent open source implementation can be downloaded from <http://www.openldap.org/>. This site also provides a reference guide for configuring OpenLDAP.

2. Configure your DX for LDAP authentication. See “Configuring Your DX Appliance for LDAP Authentication” in the next section.
3. Verify your authentication configuration by sending a request to the secured Web or application server (target host) using the DX. The DX will prompt for authentication information through a pop-up window in the browser.

### **Configuring Your DX Appliance for LDAP Authentication**

Follow these steps to enable LDAP authentication on your DX:

1. Set the general cluster parameters for LDAP authentication described in “Configuring Basic Authentication Parameters” on page 256.
2. Set the LDAP specific cluster parameters:

```
dx% set cluster secure_cluster_001 aaa authentication ldap
version 3
dx% set cluster secure_cluster_001 aaa authentication ldap
server type ADS
```

```

dx% set cluster secure_cluster_001 aaa authentication ldap
server 1 ip 192.168.40.202
dx% set cluster secure_cluster_001 aaa authentication ldap
server 2 ip 192.168.40.203
dx% set cluster secure_cluster_001 aaa authentication ldap
base-dn dc=junipernetworks,dc=com
dx% set cluster secure_cluster_001 aaa authentication ldap bind
user-dn cn=Manager,dc=junipernetworks,dc=com
dx% set cluster secure_cluster_001 aaa authentication ldap bind
password juniper
dx% set cluster secure_cluster_001 aaa authentication ldap uid
cn
dx% set cluster secure_cluster_001 aaa authentication enabled
dx% write

```

The `user-dn` information is the administrative user (the Manager in this example) for the LDAP directory. The `uid` information refers to the column name in the LDAP database that stores the username. By default it is “uid” or “cn” on most LDAP servers.

### Configuring Your DX Appliance for LDAP Authentication over SSL

If you are using SSL for the communication protocol between your DX appliance and LDAP Authentication server, the LDAP server sends a certificate to the DX appliance. The DX appliance must evaluate whether the certificate authority (CA)—in this case, the LDAP server—who issued the certificate is trusted. The LDAP server may also request a certificate from the DX appliance so that it may also evaluate whether it is a trusted source. The exchange of certificates is called certificate-based client authentication or mutual authentication. Once the DX appliance is considered a trusted source, the LDAP server uses the subject name in the certificate to determine if the client has access rights to perform the requested operation. A certificate database is required to hold the CA’s certificate, and the client’s certificate (if certificate-based client authentication is used). SSL communication is disabled by default.

To configure the DX appliance to support communication over SSL for LDAP authentication:

1. Specify the URI to the domain name specified in the certificate authority (CA) certfile for SSL. The format is “http:// <domain\_name >”.

```
dx% set cluster <name> aaa authentication ldap ssl uri <string>
```

2. Specify the CA certificate for a cluster:

```
dx% set cluster <name> aaa authentication ldap ssl
cacertfile <string>
```

3. Enable the SSL communication:

```
dx% set cluster <name> aaa authentication ldap ssl enabled
```

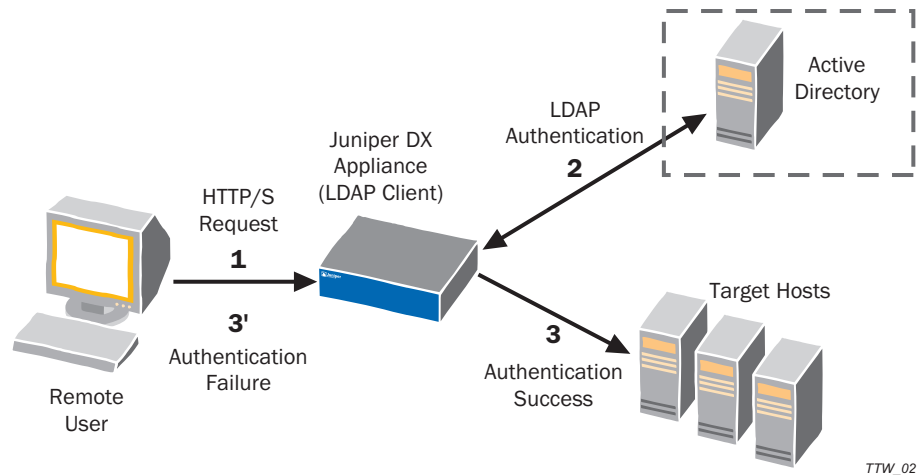
4. Save your configuration changes:

```
dx% write
```

## Configuring Authentication Using Microsoft Active Directory as the LDAP Server

To use the Microsoft Active Directory as the LDAP server for authentication with the DX, you must make a few changes to the Active Directory default configuration. By default, the Microsoft Active Directory does not permit anonymous LDAP queries. To create LDAP queries or browse the directory, an LDAP client (like the DX) must bind to the LDAP server using the Distinguished Name (DN) of an account that belongs to a Windows administrator group. Group membership search in Active Directory is performed by enumerating the `memberof` attribute of a given user entry, rather than browsing through the member list in each group. If you change this default behavior to browse each group, you can change the Group Member ID map field from `memberof:member` to `group:member`.

**Figure 57: LDAP Authentication with Microsoft Active Directory**



The numbers shown in Figure 57 indicate the authentication process flow: (1) A client makes a request for information on a target host; (2) The DX receives the client request and makes an authentication request to the Windows Active Directory on the LDAP server; (3) Authentication is successful and the DX obtains the requested information and accelerates it back to the client, or (3') Authentication is not successful and the DX returns an authentication error to the client.

To configure LDAP authentication using Active Directory, follow these general steps:

1. Configure the LDAP server running Active Directory to accept requests from remote clients.

Refer to the configuration guide for Microsoft Active Directory at <http://www.microsoft.com/windowsserver2003/technologies/directory/miis/default.aspx>

2. Configure your DX for LDAP authentication through Active Directory. See “Configuring Your DX Appliance for LDAP Authentication Using Active Directory” in the next section.
3. Verify your authentication configuration by sending a request to the secured Web or application server (target host) using the DX. The DX will prompt for authentication information through a pop-up window in the browser.

## Configuring Your DX Appliance for LDAP Authentication Using Active Directory

The DX binds by default to the LDAP server before doing any searches, and currently, does not perform group-based queries.

Follow these steps to set up Microsoft Active Directory as your LDAP server:

1. Enable LDAP authentication.

```
dx% set cluster secure_cluster_001 aaa authentication ldap version 3
```

2. Specify the IP address and type of the the server running Active Directory.

```
dx% set cluster ad_secure_cluster_001 aaa authentication ldap server 1
  ip <Active Directory IP>
dx% set cluster ad_secure_cluster_001 aaa authentication ldap server
  type ADS
```

3. Determine the full DN and password for an account in the administrators group.

For example, if the Active Directory administrator creates an account in the Users folder of the Active Directory Users and the DNS domain is `juniper.net`, the resulting DN has the following structure:

```
cn=<adminUsername>, cn=users, dc=junipernetworks, dc=com
```

4. Specify the Base DN Name, the Bind DN Name, and password based on this information.

```
dx% set cluster ad_secure_cluster_001 aaa authentication ldap base-dn
  dc=junipernetworks,dc=com
dx% set cluster secure_cluster_001 aaa authentication ldap bind user-dn
  cn=<adminUserName>,dc=junipernetworks,dc=com
dx% set cluster secure_cluster_001 aaa authentication ldap bind password
  juniper
```

5. Save the changes.

```
dx% write
```

6. Stop and restart the DX to run the new configuration.

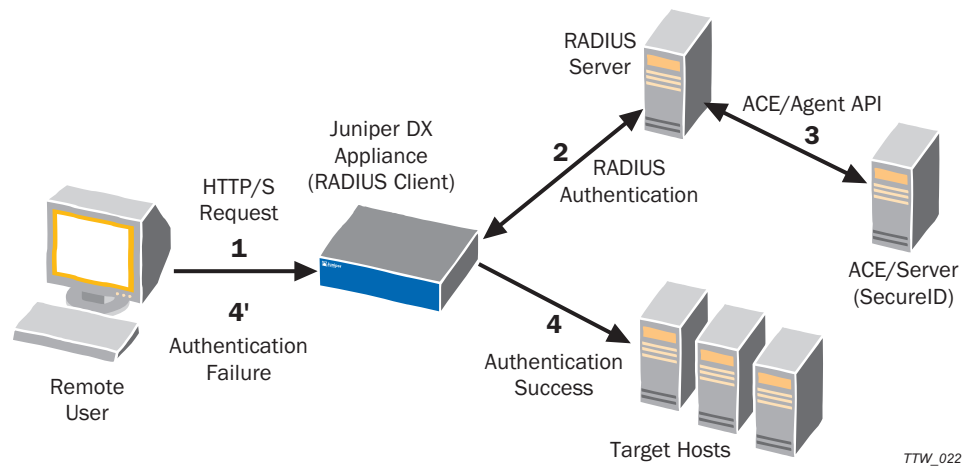
```
dx% set server down
dx% set server up
```

## Configuring Authentication Using RSA SecurID

RSA SecurID is a popular token authentication systems. RSA Security's ACE/Server and SecurID solutions provide centralized, two-factor authentication services for enterprise networks and operating systems. This allows only authorized users to gain access to network files, applications, and communications. The DX Application Acceleration Platform supports Enterprise topologies that have standardized on RSA's SecurID.

A typical authentication configuration for an Enterprise includes ACE/Server software, one DX, and a remote user (typically a browser connected to a network). The authenticating DX appliance supports token-based authentication through the RADIUS protocol. See Figure 58.

**Figure 58: Authentication Configuration Using RSA SecurID**



The numbers shown in Figure 58 indicate the authentication process flow: (1) A client makes a request for information on a target host; (2) The DX receives the client request and makes an authentication request to the RADIUS server; (3) The RADIUS server sends a request to the ACE/Server; (4) Authentication is successful and the DX obtains the requested information and accelerates it back to the client, or (4') Authentication is not successful and the DX returns an authentication error to the client.

To configure RADIUS authentication using SecurID, follow these general steps:

1. Configure the RSA ACE/Server.  
Refer to the ACE/Server configuration guide provided with your ACE/Server.
2. Configure the RADIUS server.  
Refer to RADIUS Server configuration guide provided with your RADIUS Server.
3. Configure the DX for RADIUS authentication. See “Configuring Your DX Appliance for Authentication Using SecurID” in the next section.
4. Verify your authentication configuration by sending a request to the secured Web or application server using the DX. The DX will prompt for authentication information through a pop-up window in the browser.



**NOTE:** The RADIUS protocol does not pass the syntax challenge from the authentication server. You will be challenged for a password and not a passcode.



## Configuring Your DX Appliance for Authentication Using SecurID

Follow these steps to enable RADIUS authentication for the DX:

1. Set the general cluster parameters for authentication described in “Configuring Basic Authentication Parameters” on page 256.
2. Set the RADIUS specific cluster parameters:

```
dx% set cluster radius_secure_cluster_001 aaa authentication radius
server 1 ip 120.120.120.10
dx% set cluster radius_secure_cluster_001 aaa authentication radius
server 1 port 1020
dx% set cluster radius_secure_cluster_001 aaa authentication radius
server key <shared-key>
dx% set cluster radius_secure_cluster_001 aaa authentication radius
server timeout 20
dx% set cluster radius_secure_cluster_001 aaa authentication radius
server retries 3
dx% set cluster radius_secure_cluster_001 aaa authentication enabled
```



Make sure that the secret key matches the one configured on the RADIUS Server.

3. Save the changes.

```
dx% write
```

4. Stop and restart the DX to run the new configuration:

```
dx% set server down
dx% set server up
```

## Logging Into Multiple Web Applications within a Domain

With the Single Sign-On feature, an administrator can configure Cluster services on the DX appliance to only authenticate an end-user when he or she first requests access to a Web application. The end-user is then *not* asked to re-authenticate to access either the same application at another time within this session, or to access a new Web application that is hosted in the same domain or sub-domain as the first Web application.

For example, after a user is authenticated and has accesses to app1.company.com, that user would not need to re-authenticate to access app2.company.com or app3.customer.company.com. This user has been authenticated for the \*.company.com domain.

A user who has authenticated and obtained accesses to app1.company1.com, would need to re-authenticate to access app2.company2.com because company1 and company 2 are different domains.

Domain information and user credentials are encoded (like the authentication header) and stored in a cookie. Other information contained in the cookie is not encrypted.

The cookie expires after a user specified amount of time, requiring the user to reauthenticate on subsequent access requests.



**NOTE:** Cookies on the DX appliances within a cluster must have the same timeout value for Single Sign-On to operate correctly.

---

The Single Sign-On feature is compatible with both LDAP and RADIUS authentication, as well as across DX platforms with ActiveN configurations. It is disabled by default.

### **Sample Usage of the Single Sign-On Feature**

Three examples are provided here to further clarify the usage of the Single Sign-On feature.

#### **Scenario 1—Overlapping Domains**

Clusters 1, 2, 3, and 4 are configured for Single Sign-On. Applications are accessible through the specified domains on each cluster.

- app1.company.com is accessed through Cluster 1
- app2.company.com is accessed through Cluster 2
- subapp1.app2.company.com is accessed through Cluster 3
- subapp2.app2.company.com is accessed through Cluster 4

The Single Sign-On domains for Cluster 1 and Cluster 2 are app1.company.com and app2.company.com, respectively. The cookie for Cluster 1 and Cluster 2 is \*.company.com. The Single Sign-On domain for Cluster 3 and Cluster 4 are subapp1.app2.company.com and subapp2.app2.company.com, respectively. The cookie for Cluster 3 and Cluster 4 is \*.app2.company.com.

In this scenario, Single Sign-On would apply across all clusters if the user was first authenticated on Cluster 1 or Cluster 2 because the Single Sign-On domains for all of the clusters fall into the \*.company.com cookie.

If the user was first authenticated on cluster 3 or 4, the user would be prompted for user credentials again to access applications app1 and app2 on Cluster 1 and 2. In this case, the cookie for Cluster 1 and 2 (\*.company.com) falls outside the Cluster 3 and 4 cookie (\*.app2.company.com).



**NOTE:** Specify Single Sign-On domains in all of your clusters such that the cookie is \*.company.com ensures users can access all of the applications on those clusters without providing credentials a second time (except when a cookie has expired).

---

**Scenario 2—Crossing Domains**

Clusters 1, 2, and 3 have been configured for Single Sign-On with the following domains:

- www.company1.com is accessed through Cluster 1
- www.company2.com is accessed through Cluster 2
- www.company1.org is accessed through Cluster 3

In this scenario, the user is required to enter credentials each time he or she wants to access applications on these three domains because the cookies are different for each cluster.

**Scenario 3—Valid Domains and Cookies**

Clusters 1, 2, and 3 have been configured for Single Sign-On with the following domains:

- www.company1.com is accessed through Cluster 1
- www.company2.com is accessed through Cluster 2
- www.office.company.2.com is accessed through Cluster 3

In this scenario, the user is required to enter credentials each time he or she wants to access applications on these three domains because the cookies are different for each cluster.

While at first glance it would seem that they have a common cookie, \*.com, this is not a valid cookie. The minimum cookie must have two periods because a valid domain is defined as one that has at least two periods, and typically three periods. Top level domains, including .com, .edu, .net, .org, .gov, .mil, and .int, require two periods. All other domains require at least three periods.

**Configuring the DX Appliance for Single Sign-On**

To configure a Cluster service on your DX appliance for Single Sign-On:

**Using the DXSHELL:**

1. Specify the domain for the cluster on the DX that is to allow a user to authenticate only once:

```
dx% set cluster <name> aaa authentication sso domain <domain-name>
```

2. Optionally, specify the amount of time before the Single Sign-On cookie expires. By default, the value is set to zero and the cookie expires with the browser. Alternatively, you may select a value between zero and 525600 minutes (24 hours) inclusive.

```
dx% set cluster <name> aaa authentication sso cookie timeout
<integer>
```

- Optionally specify a name for the cookie. The default name is DXAUTH.

```
dx% set cluster <name> aaa authentication sso cookie name <string>
```

- Enable Single Sign-On for the cluster.

```
dx% set cluster <name> aaa authentication sso enabled
```

### Example Configuration

The following example configures Single Sign-On for Cluster 1 and the domain company.com. The cookie is then \*.company.com. The cookie has been specified so that it expires in 10 hours, after which time the user must provide credentials to access the applications on Cluster 1.

```
set cluster 1 aaa authentication sso domain company.com
set cluster 1 aaa authentication sso cookie timeout 6000
set cluster 1 aaa authentication sso enabled
```

### Using the WebUI:

- Select the cluster to configure using either of the following methods:
  - From the Dashboard, click the Cluster that you want to configure with Single Sign-On listed in the Cluster Health Box. Click Edit.
  - Click the Services tab, and then click the Cluster that you want to configure with Single Sign-On from the Cluster list.
- Expand the AAA configuration area and scroll down to the Single Sign-On portion.

Protocol	http
<b>Single Sign On (SSO)</b>	
Enable SSO	<input checked="" type="checkbox"/>
Cookie Name	DXAUTH
Cookie Expiration	6000
SSO Domain	juniper.net
<b>Audit</b>	
Enable Audit	<input checked="" type="checkbox"/>

- Click the check box next to Enable SSO.
- Optionally specify a name for the cookie. The default name is DXAUTH.
- Optionally, specify the amount of time before the Single Sign-On cookie expires. By default, the value is set to zero and the cookie expires with the browser. Alternatively, you may select a value between zero and 525600 minutes (24 hours) inclusive.

6. Specify the domain for the cluster.
7. Click Save Changes at the bottom of the window.

### **Modifying Single Sign-On Configuration for a Cluster**

You can change the domain, cookie name, and cookie timeout for a cluster or clear the configuration altogether.

To modify the domain, cookie name, and cookie timeout using the DXSHELL, execute the appropriate `set cluster <name> sso authentication` command. To modify the configuration using the WebUI, follow the procedure you used to originally configure the Single Sign-On feature.

To clear a Single Sign-On configuration for a cluster using the DXSHELL, enter the `clear cluster <name> aaa authentication sso domain` command. To clear the configuration through the WebUI, expand the AAA portion of the Cluster configuration and uncheck the Enable SSO box, and optionally clear the domain, set the cookie to the default (DXAUTH) and the cookie timeout to zero.

### **Viewing Single Sign-On Configuration for a Cluster**

You can display the status, domain, and cookie information for each Cluster configured with Single Sign-On using the DXSHELL or the WebUI.

An example using the DXSHELL:

```
dx% show cluster 1 aaa authentication sso
Authentication SSO Status: enabled
Authentication SSO Domain: juniper.net
Authentication SSO Cookie Name: DXAUTH
Authentication SSO Cookie Expire: 6000
```

See “Show Commands” on page 255 for a listing of all Single Sign-On show commands available.

To view the configuration using the WebUI, expand the AAA portion of the Cluster configuration.



## Chapter 15

# Configuring HTTP(S) Logging

This chapter describes how to set up logging with various Web servers and to send the resulting logs to a designated machine. It also describes how to record a client IP address for each web request.

This chapter includes the following topics:

- Overview on page 269
- Configuring Logging with Apache on page 274
- Configuring Logging with IIS on page 275
- Configuring Logging with Resin on page 280
- Configuring Logging with iPlanet on page 280
- Configuring Logging with NetCache on page 281

### Overview

---

The DX acts as a proxy to all target hosts (Web servers, caches, etc.) so the IP sent to target hosts is the DX's IP (refer to Figure 59). If logging is client cookie-based, no changes are required. All client cookie information will be sent from the DX to the target hosts with each request. For logging configurations that record the origin client IP, the DX offers two options.

- For site administrators who need a common log or combined log format, the DX can compile the log information and send it to a Master logging machine either in the form of SYSLOG with the appropriate security enhancements or in the form of batches using SCP.
- Alternatively, the DX can be configured to record the client IP address for each Web request in a custom HTTP header before forwarding the request to the Web servers. For details on configuring your server to recognize this custom HTTP header, refer to the specific section in this chapter that matches your host type or server platform.

## Compiling Log Information on a Master Logging Machine

You can configure the DX to compile log information for a Cluster service and send it to a Master logging machine one log at a time using system logging or multiple logs at a time using a batch process.

### Using System Logging

You can configure the DX to send log files using the DXSHELL or the WebUI.

- From the DXSHELL:

1. Provide the Cluster with a log host:

```
dx% set cluster <name> weblog syslog host <host ip>
```

2. Optionally, modify the default syslog port from 514:

```
dx% set cluster <name> weblog syslog port <port num>
```

3. Enable logging for the Cluster:

```
dx% set cluster <name> weblog enabled
```

- From the WebUI:

1. Click the Services tab. The Clusters page is displayed by default.
2. Click the name of the Cluster from the Cluster list to use for logging.
3. Expand the Logging section to view the current configuration for logging.

The screenshot shows a configuration panel for a cluster's logging settings. The 'Logging' section is expanded, revealing several fields:

- Enable Logging:** A checkbox that is currently unchecked.
- Log Destination:** A dropdown menu set to 'syslog'.
- Log Format:** A dropdown menu set to 'common'.
- Log Delimiter:** A dropdown menu set to 'space'.
- Log Host:** An empty text input field.
- Log Port:** A text input field containing the value '514'.

Below the logging section, there is a partially visible section for 'QOS Marking - Listen Side'.

4. Click the check box next to Enable Logging.
5. Optionally, modify the Log Destination (default is the syslog machine), Format, and Delimiter.
6. Enter the IP address of the destination machine into the Log Host field.
7. If you changed the default log destination from syslog, verify the Log Port default value of 514 still applies.
8. Click Save Settings at the bottom of the page to save and apply changes.



## Using a Batch Process

You can configure the DX to send log files in a batch process using the DXSHELL or the WebUI.

- From the DXSHELL:

1. Specify the destination machine for the batch:

```
dx% set cluster <name> weblog batch host <name of the
      destination machine>
```

2. Specify the directory on the SCP server where the logs are to be sent.

```
dx% set cluster <name> weblog batch scp directory <destination
      directory>
```

3. Specify at least one user who can run the batch process:

```
dx% set cluster <name> weblog batch scp username <username>
```

4. Specify a keyfile for login by the user:

```
dx% set cluster <name> weblog batch keyfile <scp keyfile for that
      user to login>
```

5. Enable logging for the Cluster.

```
dx% set cluster <name> weblog enabled
```



**NOTE:** You may also modify the default amount of memory allocated to log batches (10 MB) and set a schedule for when log batches are copied. Refer the the *CLI Reference Guide* for details.

- From the WebUI:

1. Click the Services tab. The Clusters page is displayed by default.
2. Click the name of the Cluster from the Cluster list to use for logging.
3. Expand the Logging section to view the current configuration for logging.

The screenshot shows a configuration panel with the following settings:

- Enable Logging:**
- Log Destination:** syslog
- Log Format:** common
- Log Delimiter:** space
- Log Host:** (empty text box)
- Log Port:** 514

4. Click the check box next to Enable Logging.

- Select batch from the Log Destination drop-down list.

The screenshot shows the 'Logging' configuration section. It includes the following fields and values:

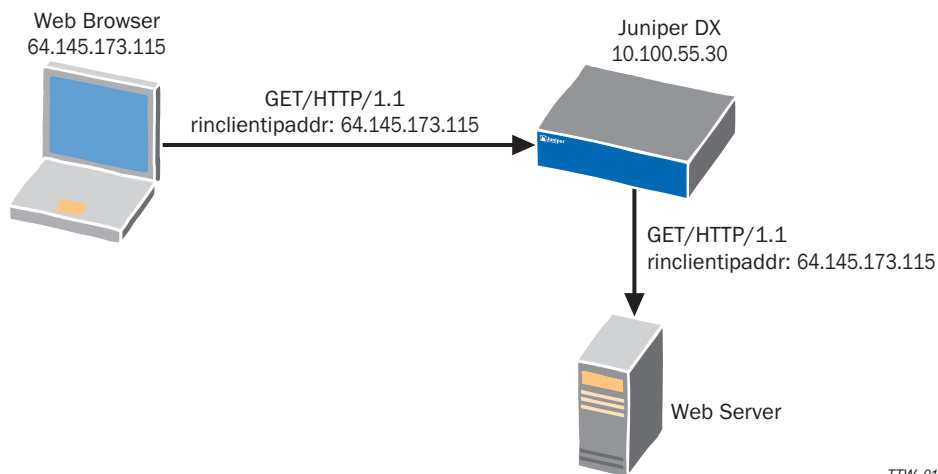
- Enable Logging:
- Log Destination: batch (dropdown)
- Log Format: common (dropdown)
- Log Delimiter: space (dropdown)
- Size: 10 (input) Total free: 50.0 MB
- Copy Time 1: (input)
- Copy Time 2: (input)
- Copy Time 3: (input)
- Copy Interval: 0 minutes (input)
- Destination Directory: (input)
- Destination Host: (input)
- SCP User: (input)
- Retry Interval: 60 seconds (input)
- SCP Keyfile: (dropdown)
- Enable Compression:

Additional fields are displayed.

- Specify the directory on the destination host where the batch logs are to be copied in the Destination Directory field.
- Specify the name of the destination host in the Destination Host field.
- Specify the username of at least one user who can run the batch process in the SCP User field.
- Specify the name of the keyfile to be used by the user to log in to the SCP server in the SCP Keyfile field. You will need to upload this file onto the DX.
- Optionally, modify any defaults for the remaining fields.
- Click Save Settings at the bottom of the page to save and apply changes.

### Recording the Client IP Address

To ensure that the client IP address is not obscured from the backend servers when using this logging method, the DX records the client IP address for each Web request in a custom HTTP header before forwarding the request to the Web servers.

**Figure 59: Flow of IP Address Information Between the Client, DX Appliance, and Server**

### Logging Client IP on the Webserver with a Custom Header

To pass client IP information on to Web servers in the HTTP header, you must set the name of the header attribute that will contain the origin client's IP address.

In the DXSHELL:

1. Specify the custom header:

```
dx% set server customiplogheader <header>
```

where the <header> is a unique string similar to standard header attributes such as "Accept" or "User-Agent."

For example, using *rinclientipaddr* as a customiplogheader value adds a line to the HTTP headers the DX sends to the Web servers it is accelerating similar to the following:

```
rinclientipaddr: 64.145.173.115
```



**NOTE:** You may also specify the custom header at the cluster level (instead of at a global level). In this case, use the `set cluster <name> customiplogheader <header>` command.

2. Configure the logging utility running on your Web servers to look for the custom HTTP header attribute and to record its value along with the other logging data.

## Configuring Logging with Apache

To configure logging with Apache from the DXSHELL:

1. Set the custom header field in which the DX will insert the origin client's IP address:

```
dx% set server customiplogheader rInclientipaddr <header>
```

2. On the Apache server, verify that `mod_log_config.so` is enabled. It is typically enabled by default on Apache 1.3.x, but it is best to check.
3. Edit `http.conf`, if necessary, to ensure that `CustomLog /var/log/httpd/access_log combined` is set (it is set by default).
4. Modify the `LogFormat` from:

```
LogFormat "%h %l %u %t \"%r\" %>s %b \"%{Referer}i\"  
          \"%{User-Agent}i\" combined
```

to

```
LogFormat "%{rInclientipaddr}i %l %u %t \"%r\" %>s %b  
          \"%{Referer}i\" \"%{User-Agent}i\" combined
```

where `{rInclientipaddr}i` matches the string that you set on the DX appliance.

5. Restart your Apache server to verify the logs reflect these changes.

As a result of this configuration, the origin client's IP address is recorded instead of the DX IP address. For example, prior to the configuration, the server's access log displayed the DX IP address (10.100.1.66):

```
10.100.1.66 - - [13/Aug/2001:12:11:25 -0700] "GET / HTTP/1.1" 304 - "-"  
"Mozilla/4.77 [en] (Win95; U)"
```

After these configuration changes, the server's access log displays the origin client's IP address, 192.168.3.87:

```
192.168.3.87 - - [13/Aug/2001:12:19:08 -0700] "GET / HTTP/1.1" 304 - "-"  
"Mozilla/4.77 [en] (Win95; U)"
```

## Configuring Logging with IIS

---

The file `rlllog.dll` is an Internet Server API (ISAPI) filter that can be installed on an IIS server in order to log the real client IP address instead of the DX IP address. Juniper Networks distributes two versions of `rlllog.dll`:

- `rlllog.dll` is compiled with the “Default” execution priority.
- `rlllog.dll_HIGH_PRIORITY` is compiled with a HIGH execution priority.

Both are included in the bundle available at the Juniper Networks Technical Support site.

In most cases, you should use the “Default” priority version of `rlllog.dll`. The “High” priority version forces IIS to execute the `rlllog.dll` filter before other ISAPI filters. This is useful when other filters need access to the real client IP address that the `rlllog.dll` inserts into the log structure `PHTTP_FILTER_LOG` in place of the DX IP address.

1. The DX must be configured to send the client IP address to IIS in a special HTTP header. From the DXSHELL command-line interface on the DX, use the command:

```
dx% set server customiplogheader rllnclientipaddr
```

to set the custom header field in which the DX will insert the origin client's IP address.

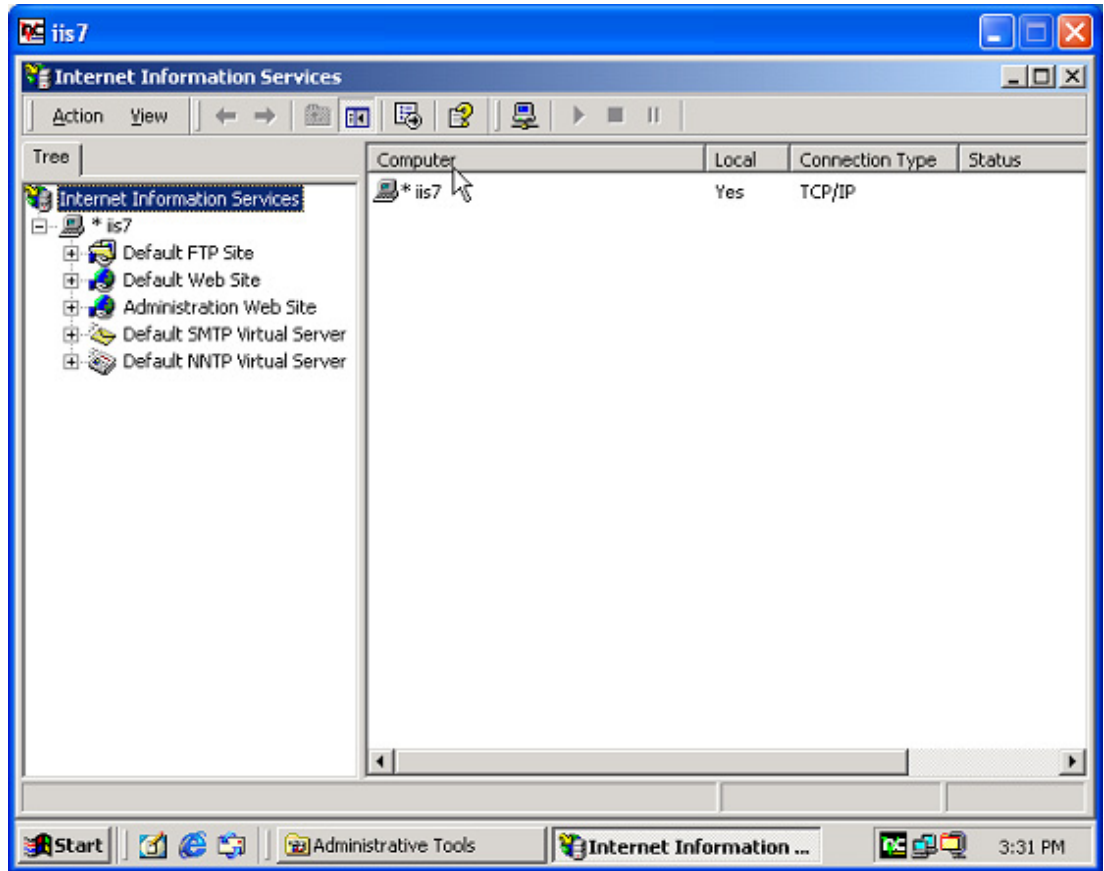
2. Select which version of the `rlllog.dll` version you are using and copy it to `%SYSTEMROOT%\system\` on the IIS server.
3. `%SYSTEMROOT%` is the directory that contains Windows system files, commonly `C:\WINNT`, so in most cases you should copy `rlllog.dll` to:

```
C:\WINNT\system\
```

**NOTE:** The DEFAULT priority version of the filter, `rlllog.dll`, is suitable for most uses.

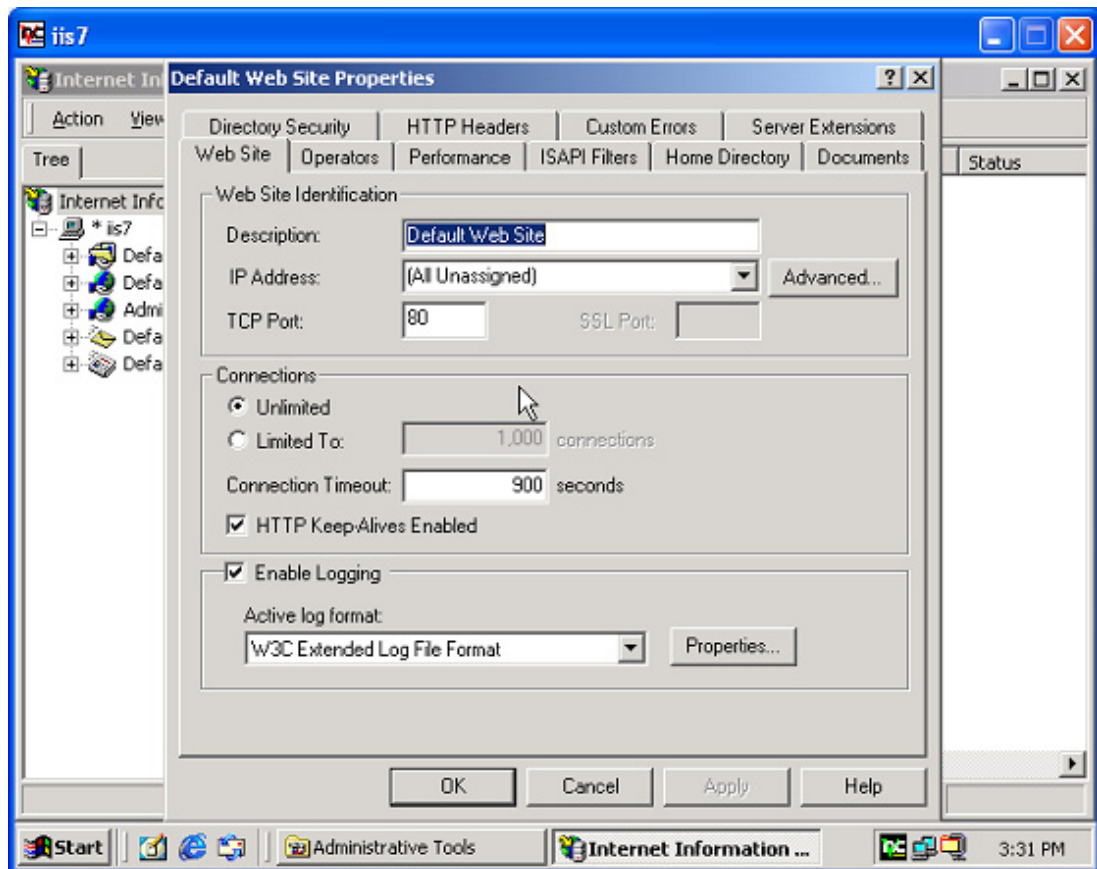
4. If you have decided to use `rlllog.dll_HIGH_PRIORITY`, rename it to `rlllog.dll`.
5. Configure the IIS server to use the `rlllog.dll` ISAPI filter.
  - Open the Internet Information Services Administrator window (refer to Figure 60).

Figure 60: The IIS Administrator Window



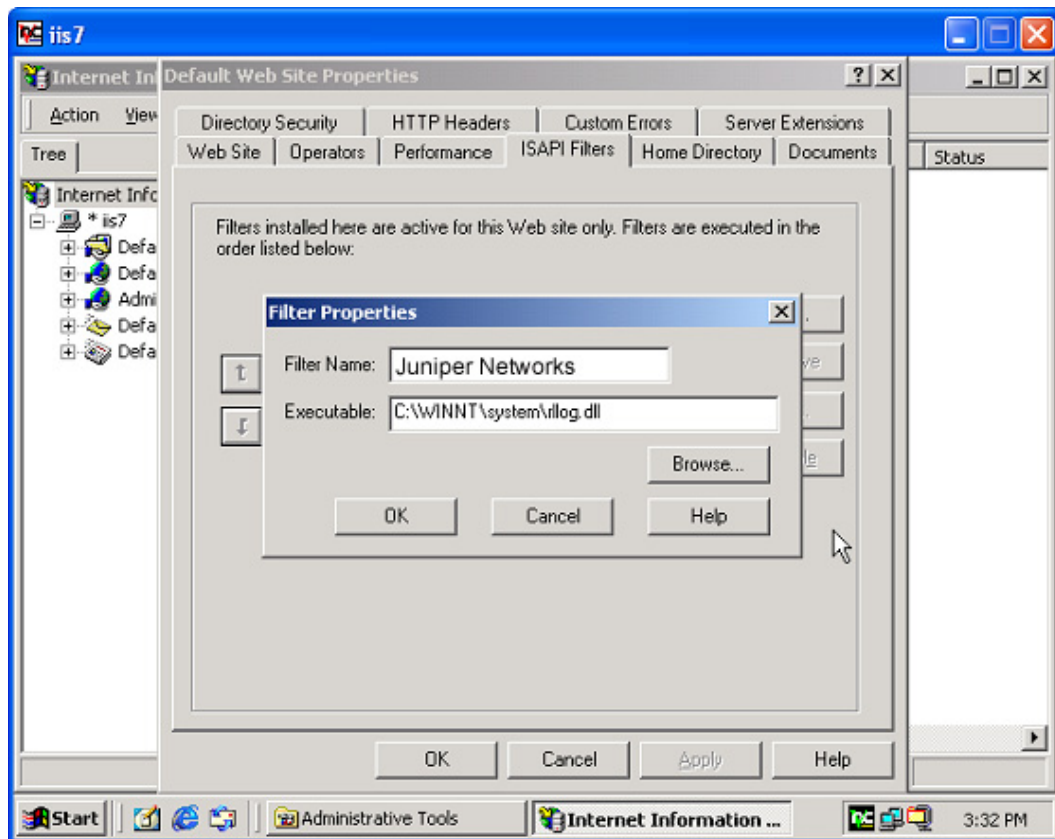
- In the left hand column of the IIS Administrator window, find the name of the Web site for which you wish to install the filter. Right click on the Web site's name to bring up a contextual menu and select PROPERTIES from the contextual menu (refer to Figure 61).

**Figure 61: The Web Site's Properties Dialog Box**



- In the PROPERTIES dialog, select the “ISAPI Filters” tab and click the ADD button to add a new filter.
  - For “Filter Name” enter:  
  
Juniper Networks
  - For “Executable” enter:  
  
C:\WINNT\system\r11log.dll  
or press the “BROWSE. ” button to locate r11log.dll on the computer (refer to Figure 62).

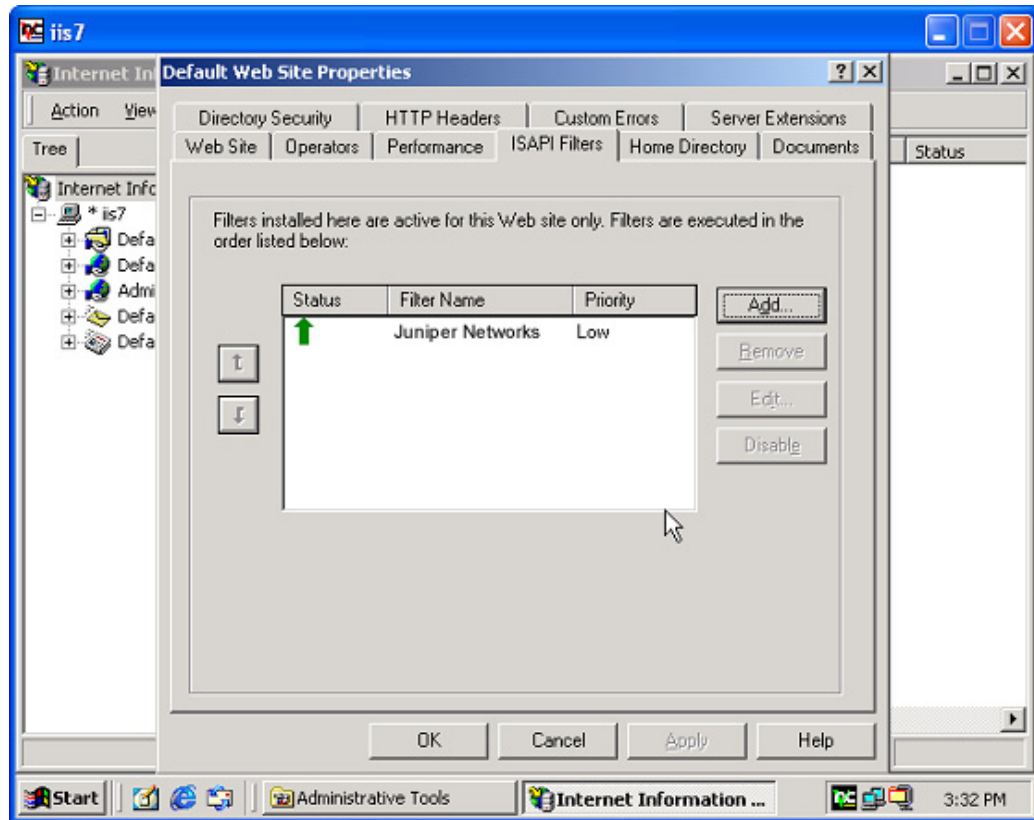
**Figure 62: Adding the r11log.dll Filter**





- Click the OK button to save your changes and close the PROPERTIES dialog box.

**Figure 63: After Adding the Juniper Networks r11log.d11 Filter**



- Use the stop and start buttons at the top of the IIS Administrator window to stop and then restart the Web service.

## Configuring Logging with Resin

---

In the file `resin.conf`, replace the line:

```
<access-log id='log/access.log' format='%h %l %u %t "%r" %s %b "%{Referer}i" "%{User-Agent}i"'/>
```

with the line:

```
<access-log id='log/access.log' format='%{rInClientipaddr}i %l %u %t "%r" %s %b "%{Referer}i" "%{User-Agent}i"'/>
```

From the DXSHELL command line interface on the DX, use the command:

```
dx% set server customipheader rInClientipaddr
```

to set the custom header field in which the DX will insert the origin client's IP address.

Now, instead of the DX IP address (10.100.1.66) in the example below appearing in the server's access log:

```
10.100.1.66 - - [19/Sep/2001:18:06:52 -0700] "GET / HTTP/1.1" 200 182 "-"
"Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)"
```

the origin client's IP address, 192.168.40.247, will be recorded:

```
192.168.40.247 - - [19/Sep/2001:18:06:52 -0700] "GET / HTTP/1.1" 200 182 "-"
"Mozilla/4.0 (compatible; MSIE 5.01; Windows NT 5.0)"
```

## Configuring Logging with iPlanet

---

Configure your DX to send the client IP address in a separate HTTP header.

From the DXSHELL, use the command:

```
dx% set server customiplogheader rInClientipaddr
```

This parameter can also be set using the WebUI from the Network Settings page.

Ensure that the shared object library from your iPlanet/Netscape server installation can be found by the Netscape binary at run time. You can do this by either copying the `rInLogTrans.so` library into your Netscape distribution's `lib` directory, or by setting the `LD_LIBRARY_PATH` environment variable to include a directory containing the `rInLogTrans.so` library.

Next, you will need to edit the `obj.conf` file for your Netscape server's installation. You cannot do this via the Netscape admin server. Find the `obj.conf` file (usually in the "configure" directory for a particular server instance). Open it with your favorite editor and add three lines:

1. At the beginning of the file `obj.conf`, near all the other Init functions (order is not important):

```
Init fn="load-modules" shlib="rInLogTrans.so" funcs="rInLogTransInit,rInLogTrans"
```

```
Init fn="r1nLogTransInit" customiplogheader="r1nclientipaddr"
```

The value for `customiplogheader` can be set to whatever you want to name the custom HTTP header used to pass the client's IP information. If it is not set, it defaults to `"r1nclientipaddr"`.

2. Now locate the `AddLog` line for the flex log. It will look something like:

```
AddLog fn="flex-log" name="access"
```

3. Put in the following line BEFORE the flex-log line:

```
AddLog fn="r1nLogTrans"
```

This forces the `r1nLogTrans` module to run before the logging module.

If you have followed the instructions in the Juniper Administrator's Guide and have converted your `"flex-init"` line to look for `r1nclientipaddr`, you should switch it back to the stock version. For example, change:

```
Init fn="flex-init" access="/opt/netscape/server4/https-servername/logs/access"
format.access="%Req->headers.r1nclientipaddr% - %Req->vars.auth-u ser% [%SYSDATE%]
\"%Req->reqpb.clf-request%\" %Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%"
```

back to

```
Init fn="flex-init" access="/opt/netscape/server4/https-servername/logs/access"
format.access="%Ses->client.ip% - %Req->vars.auth-user% [%SYSDATE%]
\"%Req->reqpb.clf-request%\" %Req->srvhdrs.clf-status% %Req->srvhdrs.content-length%"
```

Save your `obj.conf` file and restart the server.

Now instead of the DX IP address (10.100.55.30) in the example appearing in the server's access log,

```
10.100.55.30 - - [05/Jan/2004:21:58:53 -0800] "GET / HTTP/1.1" 200 24582
```

you get the client's IP address:

```
192.168.0.9 - - [05/Jan/2004:21:59:40 -0800] "GET / HTTP/1.1" 200 24582
```

If the DX is removed from the network, you get a log of whatever is upstream, such as a load-balancer or a router.

```
10.100.55.1 - - [05/Jan/2004:22:03:24 -0800] "GET / HTTP/1.1" 200 24582
```

## Configuring Logging with NetCache

1. From the DXSHELL command-line interface on the DX, type the command:

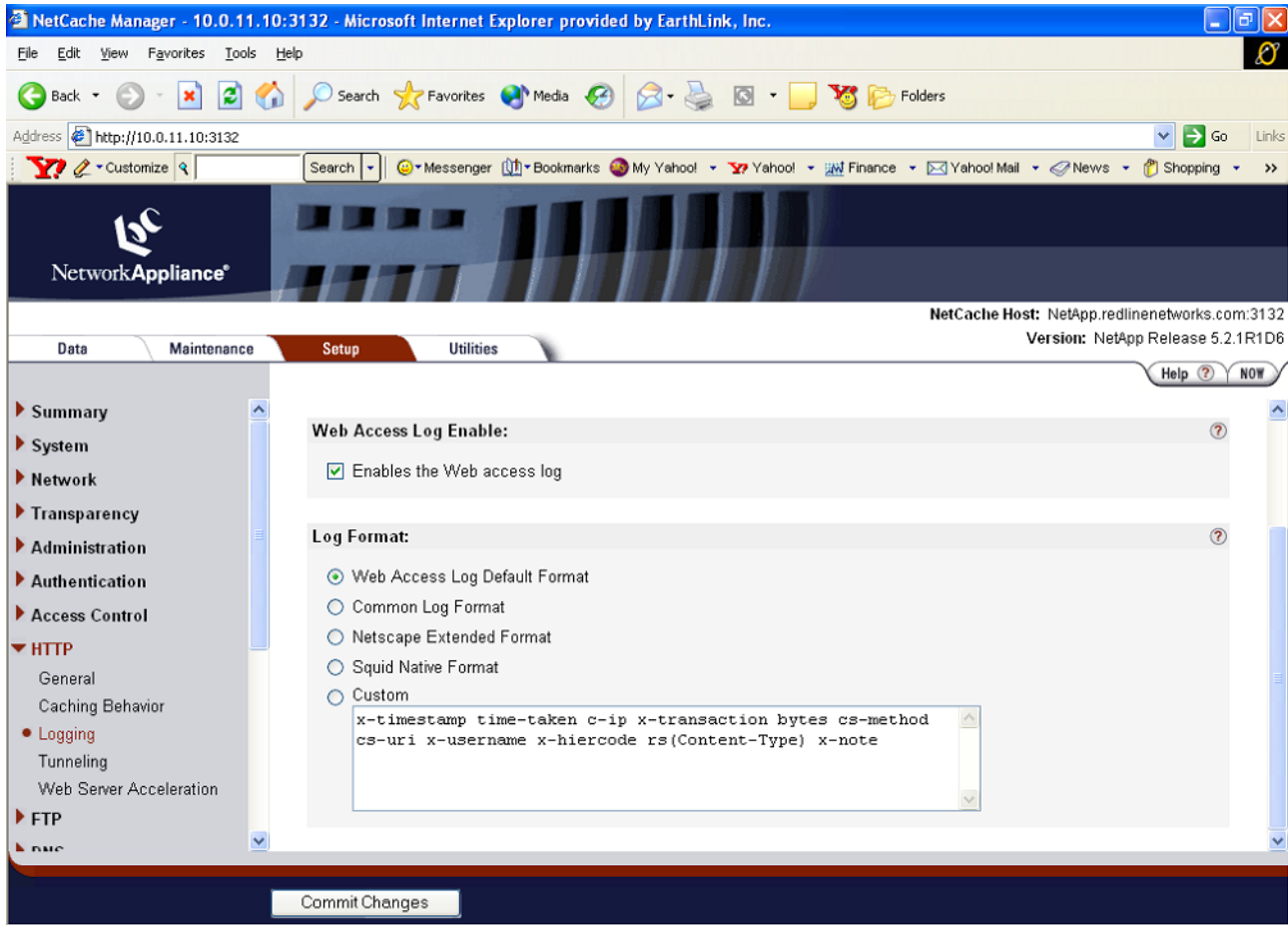
```
dx% set server customipheader r1nclientipaddr
```

to set the custom header field in which the DX will insert the origin client's IP address.

2. Configure the NetCache to retrieve the client IP from the custom HTTP header, `r1nclientipaddr`:

- Log into the NetCache Web Administrator
- Navigate to the Setup->HTTP->Logging setup screen (refer to Figure 64).

**Figure 64: The NetCache Logging Setup Screen**



- If the current log format setting is “custom”, then skip to the next step. Otherwise, copy the log format variables used by your current log format setting (in the screenshot above)

x-timestamp time-taken c-ip x-transaction bytes cs-method cs-uri  
x-username x-hiercode rs(Content-Type) x-note)

change the Log Format setting to custom and paste the variables into the Custom Log Format setting.

- In the log format variable string, replace:

c-ip

with

cs(r1nc1ientipaddr)

- Save your changes.
- Test your new configuration. Now, instead of the DX IP address appearing in the server's access log, the origin client's IP address will be recorded.



## Chapter 16

# Configuring the Forward Proxy Accelerator

This chapter describes the Forward Proxy Accelerator for the DX Application Acceleration Platform, discussing the following topics:

- “Overview” on page 285
- “Command Line Interface Commands” on page 285
- “Forward Proxy Accelerator with the WebUI” on page 287

## Overview

---

The Forward Proxy Accelerator enables the DX Application Acceleration Platform to accelerate HTTP traffic served by a forward proxy. The DX itself is NOT the forward proxy. From a position in front of a forward proxy, the DX transforms normal HTTP requests (i.e., GET, POST, PUT, etc.) as usual using compression, OverDrive (AppRules), etc. The DX also detects HTTP CONNECT requests from clients, and forwards data on those connections between the client and the forward proxy without any transformation.

The Forward Proxy Accelerator is an optional feature that requires a license file to work. Contact your Juniper Networks Sales Representative to obtain a license.

## Command Line Interface Commands

---

### Target Tuning

Use the `Target Tuning` command to enable the Forward Proxy Accelerator feature. An example:

```
dx% set cluster 1 target tune
```

This will help optimize the communication with the Target Hosts within this cluster. It will help ensure that functionality is maintained while providing the most possible benefit.

Please answer the following questions. Enter Control-C at any time to exit without modification ('\*' denotes default selection).

- 1) Please select the Target Application

- 1) Other (\*)
- 2) PeopleSoft
- 3) Domino5
- 4) Domino6
- 5) JDE
- 6) OWA
- 7) Fwd Proxy

Enter Selection: 7

- 2) Please select the Target Web Server Type
- 1) Other (\*)
  - 2) Apache
  - 3) IIS4

Enter Selection: 1

You have selected:  
Target Application: Fwd Proxy  
Target Web Server: Other  
NTLM Authentication: Required

- Continue using these selections?
- N) No, Start Over (\*)
  - Y) Yes, Use these values

Enter Selection: Y

Tuning based on your selections ...

Done.  
(\* ) dx% **write**



The audit log will display an entry as:

```
[2005-03-18 17:07:49 (+0800)] local [juniper] [cli] cluster "1" target tune:
Application = Fwd Proxy, Server = Other, NTLM = Required
```

### Cluster Set Commands

Alternatively, the Forward Proxy Accelerator feature can be enabled with these commands:

Connection Binding: Connection binding is required because target sessions that handle SSL traffic via the CONNECT method cannot be reused. Enable connection binding by typing the command:

```
dx% set cluster <name> connbind enabled
```

HTTP CONNECT: Enable support for the CONNECT method by typing the command:

```
dx% set cluster <name> httpmethod connect enabled
```

## Forward Proxy Accelerator with the WebUI

---

Target tuning has not been added as yet to the WebUI for any target type. On the WebUI, the user can manually enable “Connection Binding” and “HTTPMETHOD CONNECT”.



## Chapter 17

# Configuring the 3G Cache

This chapter describes 3G Cache for the DX Application Acceleration Platform, discussing the following topics:

- “3G Cache Commands” on page 289
- “AppRules” on page 293
- “Usage” on page 294

## 3G Cache Commands

---

The following commands are used from DXSHELL to support in-memory caching:

### Add Cache Commands

To add a cache, use the command:

```
dx% add cache <name>
```

For example:

```
dx% add cache secureImages-01_01  
(* ) dx%
```

The name can be up to 32 characters long, can be any valid character string, and may be integer-only. The valid characters are:

```
@;${^&*()=!<>,[\/_.-+0123456789
```

```
ABCDEFGHIJKLMNOPQRSTUVWXYZabcdefghijklmnopqrstuvwxyz.
```

Reserved DXSHELL keywords such as “all”, “none”, and “question” are considered invalid. If a name is not specified, one is automatically assigned. In general, this command conforms to the same rules as “add cluster.” Refer to Chapter 1 in the *Command Line Reference* manual for more information.

### Set Cache Parameter Commands

To set the total number of objects that can be stored in the named cache, use the command:

```
dx% set cache <name> max_objects <number>
```

The minimum number is 1000 and the maximum is 32000. The default value is 8192.

For example:

```
dx% set cache secureImages max_objects 28000
(*) dx%
```

As a convenience, the number may be abbreviated with a “k” suffix to indicate 1000 objects:

```
dx% set cache secureImages max_objects 2k
(*) dx%
```

To set the base size (in bytes) of the named cache, use the command:

```
dx% set cache <name> size <number>
```

The minimum number is 1,048,576 (1 megabytes) and the maximum is 104,857,600 (100 megabytes). The default value is 10,485,760 (10 megabytes). The actual size of the cache can be somewhat larger than this. As a short cut, the command is:

```
dx% set cache secureImages size 104857601
(*) dx%
```

and can be abbreviated with an “m” suffix to indicate a megabyte (1,048,576 bytes):

```
dx% set cache secureImages max_objects 3m
(*) dx%
```

### Clearing Cache Statistics and Objects

To clear the hit count statistics for the named cache, use the command:

```
dx% clear cache <name> stats
```

This does not affect the object counts.

For example:

```
dx% clear cache secureImages stats
```

To clear all objects and statistics from the named cache, use the command:

```
dx% clear cache <name>
```

For example:

```
dx% clear cache secureImages
```

## Delete Cache Commands

To delete a named cache, use the command:

```
dx% delete cache <name>
```

For example:

```
dx% delete cache secureImages
deleted
(*) dx%
```

**NOTE:** A cache cannot be deleted while it is still associated with a cluster. You must disassociate the cache from the cluster before deleting the cache.

## Associate or Disassociate a Cluster with Cache

To associate a cluster with a named cache, use the command:

```
dx% set cluster <name> cache <name>
```

For example:

```
dx% set cluster fred cache secureImages
(*) dx%
```

The cache is disabled by default. To enable or disable caching for a cluster, use the command:

```
dx% set cluster <name> cache <name> [enabled | disabled*]
```

## Clear Commands

To clear the association of a cluster with a cache, use the command:

```
dx% clear cluster <name> cache <name>
```

For example:

```
dx% clear cluster fred cache secureImages
(*) dx%
```

## Show Cache Commands

To show the configuration for a cache, use the command:

```
dx% show cache [<name>]
```

If no name is specified, all caches are displayed. For example:

```
dx% show cache
Cache [m1]
Max Objects: 8192 (8.19K)
Size: 2097152 (2.00MB)
Used by cluster: m1
```

To display existing target server-like statistics, use the command:

```
dx% show cache <name> stats [<number> | LRU <number> | MRU <number> | content_type
detail | hit_count <number> | object_size | summary ]
```

This command shows detailed statistics on the object based on criteria selected. If no criteria is selected, the statistics for all criteria are shown. LRU is the “Least Recently Used” element, and MRU is the “Most Recently Used” element. Where the commands take an optional < number > argument, < number > limits the count of printed records. The valid range for < number > is 1-100, and the default is 10.

**NOTE:** The show cache < name > stats command can display the statistics for a maximum of 100 objects.

Some examples are:

```
dx% show cache secureImages stats object_size
```

```
Object Size Statistics:
Object Size
(bytes)      # Objects  # Hits
-----
```

1 - 256	0	0
256 - 512	1	12
512 - 1K	4	48
1K - 2K	6	72
2K - 4K	1	12
4K - 8K	3	36
8K - 16K	1	12
16K - 33K	0	0
33K - 66K	1	12
66K - 131K	0	0
131K - 262K	0	0
262K - 524K	0	0
1M+	0	0

```
dx% show cache secureImages stats content_type
```

```
Content-Type Statistics:
Content-Type # Objects # Hits
-----
```

image/jpeg	3	36
text/html	1	12
image/gif	13	156

```
dx% show cache secureImages stats hit_count 5
```

Size	# Hits	Cache Time	Order	URL
2K	12	321	1	/images/FossilLogo.gif
3K	12	321	2	/images/bb120x30.jpg
2K	12	321	3	/images/main_pg.gif
1K	12	321	4	/images/yahoo_120X30.gif
1K	12	321	5	/images/yahoo_10_61.gif

```
dx% show cache secureImages stats MRU 3
```

Size	# Hits	Cache Time	Order	URL
2K	12	321	1	/images/FossilLogo.gif
3K	12	321	2	/images/bb120x30.jpg
2K	12	321	3	/images/main_pg.gif

```
dx% show cache secureImages stats LRU 3
```

Size	# Hits	Cache Time	Order	URL
35K	12	323	1	/
2K	12	323	2	/images/sh41.gif
357	12	322	3	/images/sm.gif

**NOTE:** An expired object is not removed from the cache until it is explicitly requested (a “miss”), another object needs to get cached (causing the DX to scavenge for space), or the operator removes it using the `clear cache` command. This means that the `show cache stats` command will occasionally include some expired, but not yet removed objects.

### Show Cluster Cache Commands

To show Cluster Cache statistics, use the command:

```
dx% show cluster <name> cache stats [http | io ]
```

This command shows target host-like statistics relating to the traffic a cluster is routing to a cache. If `http` is specified, only the HTTP stats are shown. If `io` is specified, only the I/O stats are shown. If neither is specified, both sets are shown.

For example:

```
dx% show cluster m1 cache stats
IO Statistics - cluster m1 cache m1
Bytes In (Resp from Cache)0B
Bytes Out (Inserts to Cache)0B

HTTP Statistics - m1
Responses from Servers:
** Total 1XX Response Codes **0
Response Code 1000
```

## AppRules

AppRules are provided to enable or disable in-memory caching. To enable caching of objects using an AppRule, the syntax is:

```
cache "<seconds>"
```

For example:

```
PTH: http_reply_code eq "200" and url ends_with ".gif" then cache "<seconds>"
```



**NOTE:** It is important that you receive an `http_reply_code` of 200, indicating that the request was successful, as you do not want to cache errant request data.

## Usage

---

This section describes how the cache feature can be implemented and configured for normal usage, and how you can test the feature to ensure that it is working correctly.

### Case 1

1. Add a cache using the command:

```
dx% add cache <name>
```

2. Set the cache parameters using the command:

```
dx% set cache <name> max_object_size <bytes> ... commands
```

3. Create an AppRule that will cache objects when they match the AppRule for caching. An example of an AppRule that caches PDF files is:

```
PTH: http_reply_code eq "200" and url ends_with ".pdf" then  
cache "30" (time in seconds)
```

To test the configuration, make an HTTP request to retrieve the HTTP object under test. Observe that the first request is retrieved from the origin server. For example, make a request to retrieve `JuniperCacheTestFile.pdf` using your Web browser. The first request for `JuniperCacheTestFile.pdf` must be retrieved from an origin server with or without caching.

Clear your Web browser cache and make another request for the same file. Make sure that the DX does not retrieve this file from the origin server. One way to assure that outcome is to delete the file from the origin server.

If the request succeeds, the caching functionality is working. If you receive error 404 ("requested object does not exist on this server"), then the object was not cached.

### Case 2

Specify an "expires" time for the cache. This is the time period after which the in-memory cache is invalidated. Run through the Normal Scenario (as described in "Case 1").

If you make the second request BEFORE the "expires" time, the cache should successfully return the object. If you receive error 404 ("requested object does not exist on this server"), then the caching functionality is not working, and the DX is attempting to retrieve the object from the origin server.

### Case 3

Specify an "expires" time for the cache. This is the time period after which the in-memory cache is invalidated. Run through the Normal Scenario (as described in "Case 1").



Make the second request AFTER the “expires” time. The cache should not have the object and since you have removed the file from the origin server, you should receive the HTTP error 404 (“requested object does not exist on this server”). If you replace the object on the origin server, the request for the HTTP object should succeed.

**NOTE:** An exception can occur when an object has expired in the cache in the DX but is still available on the server. If a client sends a request with an “If-Modified-Since” header, the DX appliance will bypass the cache (since the object is expired there) and go directly to the server to fetch the object. The server will return a code 304 (“Not Modified”), which the DX will not cache. This means that all requests with an “If-Modified-Since” header will result in a server hit until the DX receives a client request without an “If-Modified-Since” header.



## Chapter 18

# Configuring OverDrive Application Rules

This chapter provides procedures for configuring the OverDrive Application Rules Translator (AppRules) on the DX Application Acceleration Platform. AppRules is enabled and controlled using the Command Line Interface.



---

**NOTE:** The OverDrive feature requires a license. See “Obtaining a License Key” on page 124) or contact your local Juniper Sales Representative for more information.

---

This chapter contains the following topics:

- “Writing Application Rules” on page 297
- “Combining Rules into Rule Sets” on page 308
- “Importing Rule Sets” on page 318
- “Binding Rule Sets to Clusters” on page 319
- “Enabling OverDrive” on page 319
- “Modifying Rules” on page 320
- “Viewing Application Rules on the DX Appliance” on page 320
- “Limitations When Applying Application Rules” on page 321
- “Logging” on page 325
- “Application Rule Scenarios” on page 325

## Writing Application Rules

---

There are only a few steps to configuring application rules on your DX appliance. First the individual rules are written. The rules are then grouped into a rule set and imported to the DX appliance.

Application rules are roughly comprised of two parts—test conditions and actions. A single rule can contain multiple test conditions and multiple actions (although some rules only allow a single action). Actions are executed only when all of the test conditions have been met.

A “rule” is created when test conditions and actions are placed together in an ASCII text file. Use only pure text editors when writing your rules to ensure proper operation on the DX appliance. The rule defines what pieces of data can be analyzed (test variables), how they can be tested (test conditions), and what to do when the tests are true (actions). For example, a rule might be: “If the URL equals /index.html, then redirect the request to server http://www.myserver.com using the same URL as the one supplied in the request as the redirect URL”.

The basic syntax for application rules is as follows:

```
<rule_type>: <test_condition> [and <test_condition>...] then <action> [and <action> ]
```

where:

- **<rule\_type>** is a mnemonic indicating the application rule type, RS, RTH, PTH, or PTC. It is followed directly by a colon.
- **<test\_condition>** specifies a particular test condition statement. Multiple test conditions may be applied. The keyword **and** separates multiple conditions. See “Test Conditions” on page 298.
- **<action>** designates the action that is performed when all test conditions for a certain rule have been met. Some application rules only allow one action, and some allow multiple actions. The keyword **and** separates multiple actions. See “Actions” on page 304.

It is customary to separate each logical component by some amount of arbitrary whitespace, although this is not required. Single line comments can be placed in the ruleset by placing a pound sign (#) at the beginning of the line.

For example:

```
# This is a comment.
```

The following sections provide additional detail about test conditions and actions.

## Test Conditions

Test conditions are the criteria used by the DX to determine if the subsequent actions should be performed on a given HTTP request or response.

Each test condition is formatted using the following syntax:

```
<variable_statement> <operator> [sub_operator] [argument]
```

where:

- **<variable\_statement>** is either the name of the variable itself, or the variable type and then a variable name. See “Variables and Variable Types” on page 299.
- **<operator>** indicates the type of test operator to use against the variable in conjunction with the argument. See “Operators” on page 302.
- **[sub\_operator]** is an optional value that may be used with certain operators to further qualify how the operator is used.

- [argument] is the value to test against the current variable value. Not all operators require an argument. See “Arguments” on page 303.

### Variables and Variable Types

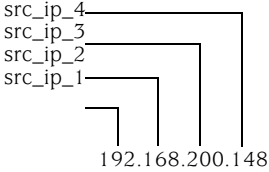
Data in either the HTTP request or response is compared to test condition variables to determine whether or not to perform an associated action. Variables are pre-determined values, such as URL or HTTP version. The DX compares the piece of request data against this pre-defined value to evaluate its validity. For example, a test condition might be “Is the URL for this request equal to /index.html?” If this condition is valid, then the test condition resolves as true.

The variables and variable types supported by the DX appliance are shown in Table 8. All variables must be contained inside double quotes.

**Table 8: Supported Variables and Variable Types**

Variable Type/ Variable	Description
url	Variable. URL of the HTTP request.
query_string	Variable. Portion of the URL that exists after the '?' If the '?' is not present in the URL, then the <query_string> is considered to be non-existent.
request_header “<header_name>”	Variable type. Examines the “request” HTTP header with the specified header name. A header name can only contain A-Z, a-z, 0-9, '-', and '_' characters. It should not refer to a “Cookie” header as they are treated separately.
any_request_header	Variable. Any HTTP header in the request (except Cookie headers).
request_cookie “<cookie_name>”	Variable type. Examines the “Cookie” HTTP headers for the specified cookie name. A cookie name can only contain A-Z, a-z, 0-9, '-', and '_' characters.
any_request_cookie	Variable. Any Cookie header and name pair in the request.
http_request_version	Variable. HTTP version of the request (1.0 or 1.1).
reply_header “<header_name>”	Variable type. Examines the “reply” HTTP header with the specified header name. A header name can only contain A-Z, a-z, 0-9, '-', and '_' characters. It should not refer to a “Set-Cookie” header as they are treated separately.
reply_cookie “<cookie_name>”	Variable type. Examines the “Set-Cookie” HTTP header with the specified name. A cookie name can only contain A-Z, a-z, 0-9, '-', and '_' characters.
http_reply_code	Variable. HTTP code that appears in the reply. For example 200, 404, or 502.
http_reply_version	Variable. HTTP version of the reply (1.0 or 1.1).
http_method	Variable. HTTP method used in the request. For example Post, Get, or Head.
src_ip, sip	Variable. Client's IP address. Also known as the source IP address. All comparisons are made against IPv4 dot notation addresses, so test arguments should be made accordingly.

**Table 8: Supported Variables and Variable Types (continued)**

Variable Type/ Variable	Description
src_ip_1, src_ip_2, src_ip_3, src_ip_4	<p>Test variables corresponding to the four octets of the IPv4 network address belonging to the requestor. The numeric designator indicates the octet of the address. The first octet is considered the "class A" octet, second octet is the "class B" one, and so forth. For example:</p>  <p style="text-align: center;">192.168.200.148</p> <p>These variables allow for fine-grained checking of source IP addresses on a per-octet level. You can specify ranges of IP addresses for which a rule applies, making it easier to "classify" many users much like subnetting does.</p>
content	Variable. Any ASCII-based data located in an HTTP response after the HTTP response headers. This variable only operates with the contains and ci_contains test condition operators.
ssl_cipher_bits	Variable. Size of the SSL cipher key. It can be up to four digits (less than 1024).
ssl_cipher_suite	Variable. List (suite) of ciphers.
ssl_version	Variable. Supported Secure Socket Layer (SSL) version. The value is case-sensitive, and must be entered in the form: SSLv2, SSLv3 or TLSv1.

The `header_name` variable used in the request or reply header variable types have predefined values, as listed in Table 9. These values are all case-sensitive.

**Table 9: Valid Header Variable Names**

Header Variable Name	Header Variable Name	Header Variable Name
~	~ ~	~ ~ ~
~ ~ ~ ~	~ ~ ~ ~ ~	~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~	~ ~ ~ ~ ~ ~ ~	~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~	~ ~ ~ ~ ~ ~ ~ ~ ~	~ ~ ~ ~ ~ ~ ~ ~ ~ ~
~ ~ ~ ~ ~ ~ ~ ~ ~ ~	~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~	~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~ ~
Accept	Accept-Charset	Accept-Encoding
Accept-Language	Accept-Ranges	Age
Allow	Allow-Rename	Apply-To-Redirect-Ref
Authorization	Brief	Cache-Control
Call-Back	Connection	Content-Base
Content-Encoding	Content-Language	Content-Length
Content-Location	Content-MD5	Content-Range
Content-Type	Cookie	Cookie2
DASL	Date	DAV

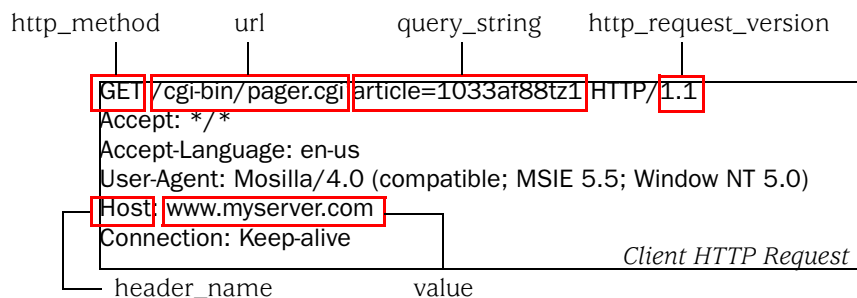
**Table 9: Valid Header Variable Names (continued)**

Header Variable Name	Header Variable Name	Header Variable Name
Depth	Destination	Edge-Control
E-Tag	Expect	Expires
From	FRONT-END-HTTPS	Host
If	If-Match	If-Modified-Since
If-None-Match	If-Range	If-Unmodified-Since
Keep-Alive	Label	Last-Modified
Lock-Token	Max-Forwards	Notification-Delay
Notification-Type	Ordered	Overwrite
Position	P3P	Pragma
Proxy-Authenticate	Proxy-Authorization	Proxy-Connection
Public	Range	Redirect-Ref
Referer	Retry-After	Server
Set-Cookie	Set-Cookie2	SSLClientCipher
Status-URI	Subscription-ID	Subscription-Lifetime
TE	Timeout	Trailer
Transaction	TransactionID	Transfer-Encoding
Upgrade	User-Agent	Vary
Via	Warning	WL-Proxy-SSL
WWW-Authenticate	X-iCMS-Cache-Control	X-Powered-By
X-rln-Effective-Len		



**NOTE:** Actions only operate on the content variable; all other variables are used only for reference within test conditions. The relationship between the actions is such that only one action can be performed per content rule (e.g., prepend, append, or replace). This differs from the header-oriented rules and the request sentry where multiple actions may be specified per rule.

The following example shows the relationship between an incoming client's HTTP request and the AppRule variables specified in a given rule.

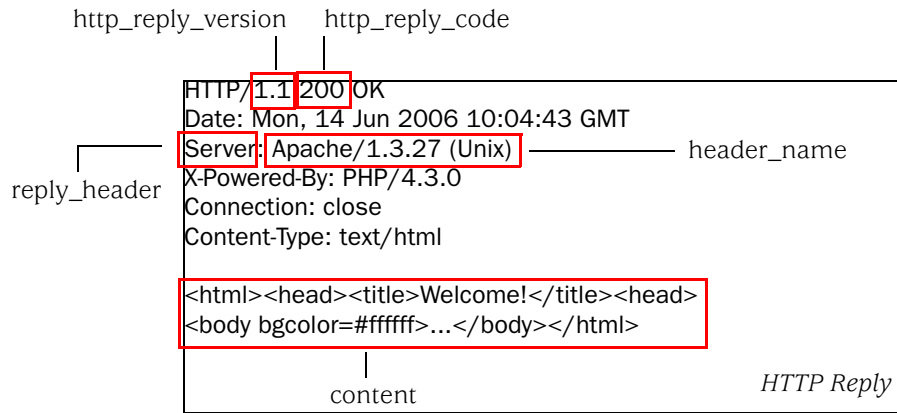


By way of example, you might have a language translation rule like this:

RTH: request\_header "Accept-Language" equals "fr-ca" then update\_header "Host" "french-canada.myserver.com"

In this rule, the host is changed based on the language in the HTTP request. All browsers that indicate a French Canadian user (language code fr-ca) are sent to the virtual host french-canada.myserver.com.

The following example shows the relationship between the HTTP reply and the AppRules variables specified in a given rule. The content variable includes any data returned back to the user.



The content variable is used only by the content-oriented AppRules (Request Translator Content and Page Translator Content).

### Operators

The operators shown in Table 10 are used when formulating test conditions. Each operator has a long syntax and a short syntax; both are equally valid and can be used interchangeably.

**Table 10: Operators Used When Formulating Test Conditions**

Operator	Description
ex, exists	True if the variable's value exists; no test argument is necessary. Applicable only to header variables.
nx, not_exists	True if the variable's value is not present in the request; no test argument is necessary. Applicable only to header variables <sup>1</sup> .
eq, equals	True if there is an exact (case-sensitive) match between a variable's value and the test argument.
ci_eq, ci_equals	True if there is a case-insensitive exact match between a variable's value and the test argument.
ne, not_equals	True if the variable's value is not equal to the test argument.
ci_ne, ci_not_equals	True if the variable's value is not equal to the case-insensitive test argument.
c, contains	True if variable's value contains the argument within it.



**Table 10: Operators Used When Formulating Test Conditions (continued)**

Operator	Description
ci_c, ci_contains	True if the variable's value contains the case-insensitive argument within it.
nc, not_contains	True if the variable's value does not contain the test argument.
ci_nc, ci_not_contains	True if the variable's value does not contain the case-insensitive test argument.
sw, starts_with	True if the variable's value starts with the specified argument.
ci_sw, ci_starts_with	True if the variable's value starts with the specified case-insensitive argument.
ns, not_starts_with	True if the variable's value does not start with the test argument.
ci_ns, ci_not_starts_with	True if the variable's value does not start with the case-insensitive test argument.
ew, ends_with	True if the variable's value ends with the specified argument.
ci_ew, ci_ends_with	True if the variable's value ends with the specified case-insensitive argument.
nw, not_ends_with	True if the variable's value does not end with the test argument.
ci_nw, ci_not_ends_with	True if the variable's value does not end with the case-insensitive test argument.
l_gt, length_greater_than	True if the variable's value is not longer than the test argument numeric value (as specified in bytes)
l_lt, length_less_than	True if the variable's value is less than the test argument numeric value (as specified in bytes).
l_eq, length_equals	True if the variable's value is exactly the same length as the test argument numeric value (as specified in bytes).
l_ne, length_not_equals	True if the variable's value is not the same length as the test argument numeric value (as specified in bytes).
greater_than	True if the variable's value is greater than the test argument's numeric value.
less_than	True if the variable's value is less than the test argument's numeric value.

1. The semantic of “any\_request\_header” does not mean “any one request header” when used with the “not\_exists” operator. This application rule is considered to be “all request headers as a group.” The implication is that the test condition “any\_request\_header” not\_exists, treats all the request headers as a group and determines their non-existence as a group.

For example, “GET / HTTP/1.0\r\n \r\n” results in the test condition returning TRUE, while the request “GET / HTTP/1.0\r\n Host: www.xyz.com\r\n \r\n” results in the test condition returning FALSE, because as a group the request headers do exist.

### Arguments

All test arguments must be enclosed within double quotes and cannot span more than a single line.



**NOTE:** Use a pure text editor, such as Wordpad, when creating the arguments. To ensure that the quotes are processed by the AppRules engine.

Character restrictions are placed on the test argument depending upon the test variable as shown in Table 11.

**Table 11: Restrictions on Test Variable Arguments**

Test Variable	Acceptable Characters
url test arguments	A-Z, a-z, 0-9, '%', ':', '/', '_', ',', ';', '~'
http_request_version and http_reply_version test arguments	0.9, 1.0, 1.1
http_request_header, any_request_header, and reply_header variable test arguments	A-Z, a-z, 0-9, space, '(', ')', '/', '\', ':', ';', ',', '_', '-', ':', '~', '%', '>', '+', '<', '!', '@', '#', '\$', '&', '\'', '='
header_name values	A-Z, a-z, 0-9, '_', '-'
request_cookie and reply_cookie variable test arguments	# A-Z, a-z, 0-9, space, '(', ')', '/', '\', ':', ';', ',', '_', '-', ':', '~', '%', '>', '+', '<', '!', '@', '#', '\$', '&', '\'', apostrophe('), '?', '[', ']', '^', '*', '{', '}', ' ', '='
cookie_name values	# A-Z, a-z, 0-9, '(', ')', '/', '\', ':', ';', ',', '_', '-', ':', '~', '%', '>', '+', '<', '!', '@', '#', '\$', '&', '\'', apostrophe('), '?', '[', ']', '^', '*', '{', '}', ' '
content variable test arguments	A-Z, a-z, 0-9, space, ':', ':', ':', ':', ':', '@', '#', '\$', '%', '^', '&', '(', ')', '=', '+', '[', ']', '{', '}', ' ', ':', ':', '<', '>', '?', '~', '\', (as escape character), '!', '/', '_', '-',  escaped characters: newline (\n), carriage return (\r), double quote (\"), asterisk (*), backslash (\),  wildcard character '*'
content-length header variable names	The length test argument is limited to a range from 1 to 99999.

The **content** test argument offers functionality not present in the other variables; namely, the ability to escape certain characters and the use of an asterisk (\*) as a wildcard character. The wildcard character represents zero to 26 arbitrary characters and can be present only within the content test argument (i.e., a content test argument cannot begin or end with an unescaped \*). The argument as a whole is considered a search term.

**Actions**

When test conditions are met, then actions occur. Actions involve changing something about the connection, the request, or the reply. Multiple action statements can exist for a single AppRule in some instances. When multiple action statements exist, the keyword **and** separates each action statement.

Each predefined action statement, which specifies the action to be taken, has its own unique syntax. Table 12 lists the available action statements with their syntax and describes what effect that action has on the connection, request, or reply.

**Table 12: Action Statements**

Action	Description
insert_request_header “ < header_name > ” “ < header_value > ”	The insert_request_header action is used to insert a new, previously non-existing HTTP header as defined by < header_name > with the value < header_value > into the request. If the header already exists in the request, then the old one is first deleted and the new one inserted. This action is only available in the Request Translator Header rule type.
update_request_header “ < header_name > ” “ < header_value_1 > ” [“ < header_value_2 > ” . . . “ < header_value_N > ”]	The update_request_header action alters the value of the existing request header as designated by < header_name > with one or more values, < header_value_1 > to < header_value_N > . If more than one value is specified, each value is used in a round-robin fashion. If the header < header_name > does not exist in the client request, then no action is performed (this is the primary difference between insert_request_header and update_request_header). This action is only available in the Request Translator Header rule type.
delete_request_header “ < header_name > ”	The delete_request_header action removes a request header from the client request matching < header_name > . This action is only available in the Request Translator Header rule type.
insert_request_cookie “ < name > ” “ < value > ”	This action injects an additional cookie with < name > and < value > into the client's request. This action is only available in the Request Translator Header rule type.
delete_request_cookie “ < name > ”	Removes the cookie corresponding to < name > if found in the client request. This action is only available in the Request Translator Header rule type.
insert_reply_header “ < header_name > ” “ < header_value > ”	The insert_reply_header action is used to insert a new, previously non-existing HTTP header as defined by < header_name > with the value < header_value > into the outgoing reply. If the header already exists in the reply, then the old one is first deleted and the new one inserted. This action is only available in the Page Translator Header rule type.
update_reply_header “ < header_name > ” “ < header_value_1 > ” [“ < header_value_2 > ” . . . “ < header_value_N > ”]	The update_reply_header action alters the value of the existing reply header as designated by < header_name > with one or more values, < header_value_1 > to < header_value_N > . If more than one value is specified, each value is used in a round-robin fashion. If the header < header_name > does not exist in the outgoing reply, then no action is performed (this is the primary difference between insert_reply_header and update_reply_header). This action is only available in the Page Translator Header rule type.
delete_reply_header “ < header_name > ”	The delete_reply_header action removes a reply header from the outgoing reply matching < header_name > . This action is only available in the Page Translator Header rule type.

**Table 12: Action Statements (continued)**

Action	Description
insert_reply_cookie “ < name > ” “ < value > ” “ < domain > ” “ < path > ” [“ < expires_date > ”] [secure]	<p>This action injects an additional cookie with &lt; name &gt; and &lt; value &gt; into the client's request that will be valid for the given &lt; domain &gt; and &lt; path &gt; .</p> <p>Optionally, the expiration date and/or secure flag may be indicated. The &lt; domain &gt; value must contain either two or three dots in it (juniper.net should be .juniper.net). The path must begin with "/". Note that the &lt; expires_date &gt; must be Wdy, DD-Mon-YYYY HH:MM:SS GMT. The secure flag indicates that the cookie should only be sent over an SSL connection. This action is only available in the Page Translator Header rule type. The &lt; expires_data &gt; can also be an integer specifying the number of seconds from the time the cookie is inserted that it will expire.</p>
append < variable > [term] “ < append_value > ”	<p>The append action inserts &lt; append_value &gt; either just after the &lt; variable &gt; 's value or just after the search string match point within &lt; variable &gt; 's value. If the term keyword is used, then the append operation occurs at the search string match point; otherwise, it occurs at the end of the &lt; variable &gt; 's value. An additional requirement may exist where the &lt; variable &gt; must be a valid &lt; variable &gt; that exists in one of the test conditions within the rule where the append action appears and that same test condition must employ certain test conditions to be valid. See “Using Prepend, Append, and Replace (PAR) Actions” on page 323 for more information. When the append action is used in conjunction with the content variable, the term keyword must be present (appending data to the end of content data makes no sense since the content data is being streamed in indeterminately-sized packets). This action is available in the Request Translator Header, Page Translator Header, and Page Translator Content rule types.</p>
prepend < variable > [term] “ < prepend_value > ”	<p>The prepend action inserts &lt; prepend_value &gt; either at the start of &lt; variable &gt; 's value or just before the search string match point within the &lt; variable &gt; 's value. If the term keyword is used, then the prepend operation occurs at the point in the test condition &lt; variable &gt; 's value where it is evaluated as being true (i.e., where the search string matched). An additional requirement may exist where the &lt; variable &gt; must be a valid &lt; variable &gt; that exists in one of the test conditions within the rule where the prepend action appears and that same test condition must employ certain test conditions to be valid. See “Using Prepend, Append, and Replace (PAR) Actions” on page 323 for more information. When the prepend action is used in conjunction with the content variable, the term keyword must be present (it makes no sense to prepend content data at the beginning of the content as the content being sent back to the client is not entirely buffered--the data is streamed out in packets). This action is available in the Request Translator Header, Page Translator Header, and Page Translator Content rule types.</p>

**Table 12: Action Statements (continued)**

Action	Description
replace <variable> [term] “ <replace_value> ”	<p>The replace action inserts &lt;replace_value&gt; either in place of the complete &lt;variable&gt;'s value or just the search string that matched within the &lt;variable&gt;'s value. If the term keyword is used, then the replace operation overwrites just the matched search string. An additional requirement may exist where the &lt;variable&gt; must be a valid &lt;variable&gt; that exists in one of the test conditions within the rule where the replace action appears and that same test condition must employ certain test conditions to be valid. See “Using Prepend, Append, and Replace (PAR) Actions” on page 323 for more information. When the replace action is used in conjunction with the content variable, the term keyword must be present (it makes no sense to replace content data since the content being sent back to the client is not entirely buffered--the data is streamed out in packets making for an indeterminate replacement based on the amount of data in a packet at any given moment). This action is available in the Request Translator Header, Page Translator Header, and Page Translator Content rule types.</p>
close_conn <RST FIN>	<p>Closes the client connection and sends either a FIN packet or an RST, depending on which is specified.</p>
redirect “http[s]://host[: <port > ]/[ <prepend_path > ]” [“ <URL > ”]	<p>Returns an HTTP 302 (redirect) reply to the client using the specified protocol/host and optional &lt;port &gt; , &lt;prepend_path &gt; , and &lt;URL &gt; (in actuality, it's the URI) as the Location header value. If the &lt;URL &gt; is not specified, then the URL from the client's request is used in its stead. If the optional &lt;prepend_path &gt; is specified, then whatever &lt;URL &gt; is used (either explicitly stated in the action or taken from the client's request) is prepended with that value.</p>
forward	<p>The forward action drops the client TCP connection into "forward" mode. In this mode the data is forwarded from the client TCP connection to the target TCP connection and visa-versa. The DX appliance does not perform any cluster-related processing (such as compression, any application rules that follow this one, caching, and so forth.) Subsequent requests on this TCP connection are not modified or enhanced. This action is available in the Request Sentry, Request Translator Header, and Page Translator Header rule types.</p>
reply 30x	<p>Redirection as follows:</p> <ul style="list-style-type: none"> <li>■ 301: Moved permanently</li> <li>■ 302: Found; returns “http[s]://[host[: &lt;port &gt; ]]/[ &lt;prepend_path &gt; ]” [“ &lt;URL &gt; ”] as the body of the message.</li> <li>■ 303: See other.</li> <li>■ 307: Temporary redirect.</li> </ul>
reply 404 [ <404_file > ]	<p>Returns an HTTP 404 (not found) message using the content from the &lt;404_file &gt; as the body of the 404 message. The &lt;404_file &gt; must be imported onto the DX appliance using the Capture File command, and the argument string can have the following allowed characters: a-z, A-Z, 0-9, '-', '_', '.' For example: RS: url sw “/” then reply 404 “my404.html”</p>

**Table 12: Action Statements (continued)**

Action	Description
cache	When specified, place the HTTP reply in on-board cache. This action is available with the Page Translator Header rule type.
log	When specified, the rule that is executed is logged. This action is available only with the Request Sentry rule type.
route_request target_host “ < ip:port > ” route_request target_host “ < ip1 :port1 > ” [“ < ip2:port2 > ”] ... [“ < ipN:portN > ”	The route_request AppRule for the Request Translator Header (RTH) allows users to route a request if an incoming request meets a test condition. You can specify the individual target host, a list of target hosts, or a group of target hosts. For more information, see “Route Request Application Rules” on page 325.
retry_request [same   nosame   all] “number” and log	The retry_request AppRule allows users to retry a request if the response code for a previous request meets a test condition. For more information, see “Request Retry, Alerting, and Log (Transaction Assurance) AppRules” on page 326.
continue	This is a special action that does not alter the request in any way. Rather, it is used to override the default behavior for how RTH and PTH rules are executed. When this action is present, the subsequent rule in the rule set will be executed. This allows for a logical AND behavior to exist across individual rules in a rule set. Note that this action cannot be used with any rule that contains an I/O-based action (for example, redirect, retry_request, or route_request). A continue action may not exist by itself since it does not add any additional value to the rule (it would only act as an AND operation and the DX appliance already supports multiple test conditions with the and keyword).

## Combining Rules into Rule Sets

An application rule set is a collection of application rules. The DX appliance executes application rule sets by category, beginning with the all of the RS rules, and continuing with all of the RTH rules, all of the PTH rules, and finally all of the PTC rules. Within each category of application rules, the DX appliance executes the rules in order, from 1 to n. When a rule in any given category hits (matches its test conditions and its action(s) are performed), or if none of the rules in that category hit, the DX appliance continues on to the first rule in the next category.

There are a couple of exceptions to this process:

- A **continue** action is included in a RTH or PTH rule. When an RTH or PTH rule hits and ends with the “**and continue**” action, the next RTH or PTH rule in the list is tested.
- PTC rules in a rule set are tested and processed in order, no **continue** action is needed.

## Rule Execution Modes

Two execution modes exist for application rules, as follows:

- **Exclusive**—rule that operates on all aspects of requests and replies except for content. It must be executed by itself. No other rule of the same type may be combined with it. Once this rule is executed, the DX moves on to execute the next category of rules.
- **Collective**—rule that operates on content. It can be combined with another rule of the same type. Once this rule is executed, the DX moves on to execute the next of the same type (if the `continue` action is present in the rule) or to the next category of rules.

Table 13 summarizes the execution modes for each rule type.

**Table 13: Application Rule Execution Modes by Rule Type**

Rule Type	Execution Mode
Request Sentry (RS)	Exclusive
Request Translator Header (RTH)	Exclusive (Default)
Page Translator Header (PTH)	Exclusive (Default)
Page Translator Content (PTC)	Collective

Because there are exceptions for every rule, the AppRule grammar provides mechanisms that allow rules running in Exclusive mode to run “semi” exclusively, and allow rules running in Collective mode to run “semi” collectively.

## Action Execution Modes

Actions come in two forms, IO actions and non-IO actions. IO actions contained in a single rule operate in one of three execution modes, as follows:

- **Exclusive**—An action that must be executed by itself. The action *cannot* be combined with any other actions within the same rule. Actions in this category include `close_conn`, `redirect`, `reply`, and `retry_request`.
- **Semi-exclusive**—An action that *cannot* be combined with any other exclusive or semi-exclusive actions within the same rule, but *can* be combined with cooperative actions. Actions in this category include `route_request`.
- **Cooperative**—An application rule that contains an action that *can* be combined with any other cooperative or semi-exclusive actions within the same rule, but *cannot* be combined with exclusive actions. Actions in this category include `forward`.

This is summarized in Table 14. E represents an exclusive action, SE represents a semi-exclusive action, and C represents a cooperative action.

**Table 14: Action Execution Modes**

Action	IO	Non-IO
insert_request_header		X
update_request_header		X
delete_request_header		X
insert_request_cookie		X
delete_request_cookie		X
insert_reply_header		X
update_reply_header		X
delete_reply_header		X
insert_reply_cookie		X
append		X
prepend		X
replace		X
close_conn	E	
redirect	E	
forward	C	
reply 301	E	
reply 302	E	
reply 303	E	
reply 307	E	
reply 404	E	
cache		X
route_request target_host	SE	
retry_request	E	
log		X
continue		X

### Application Rule Relationships

This section describes the relationships between the test variables, test operators, and actions for each type of application rule.

#### Request Sentry Application Rules

The request sentry rules operate at the connection level by allowing, denying, or possibly redirecting a request based on certain criteria. The applicable test variables, test conditions, and actions for the RS rules are presented in Table 15 and Table 16.



**Table 15: Operator Availability for Request Sentry Test Variables**

Test Variable	Test Operator							
	exists, not_exists	equals, not_equals, ci_equals, ci_not_equals	contains, not_contains, ci_contains, ci_not_contains	ends_with, not_ends_with, ci_ends_with, ci_not_ends_with	starts_with, not_starts_with, ci_starts_with, ci_not_starts_with	length_less_than, length_greater_than	less_than, greater_than	length_equals, length_not_equals
url		X	X	X	X	X		X
query_string	X	X	X	X	X	X		X
request_header “ < header_name > ”								
any_request_header	X	X	X	X	X	X		X
request_cookie “ < header_name > ”								
any_request_cookie								
http_request_version		X						
http_method		X						
src_ip, sip		X	X	X	X			
src_ip1, src_ip_2, src_ip_3, src_ip_4		X					X	
ssl_cipher_bits							X	
ssl_cipher_suite		X	X	X	X			
ssl_version		X	X	X	X			

There is no qualification regarding the actions and their relationship to the test conditions. Any legal RS test condition can be used in combination with any legal RS action. The relationship of actions between one another in a single rule is shown in Table 16.

**Table 16: Request Sentry Action Matrix**

Action	Action				
	close_conn < RST FIN >	redirect (reply 301, 302, 303, 307)	reply 404	forward	log
close_conn < RST FIN >	X				X
redirect (reply 301, 302, 303, 307)		X			X
reply 404			X		X
forward					X
log	X	X	X		X

As you can see, each of the connection-handling actions must stand alone, but may be used in conjunction with the logging action.

**Request Translator Application Rules**

Request Translator AppRules are designed to modify incoming requests at either the header level or the content level. The header and content Request Translator rules are shown separately.

**Request Translator Header Application Rules**

The Request Translator Header (RTH) rules operate on the HTTP header segment of the incoming request. This includes the URL and query string, along with the headers that may be part of the request. The test variables, test conditions, and actions for RTH rules are described as follows.

The relationships between the RTH test variables and the RTH test operations that are allowed are shown in Table 17. Note that all of the actions can interoperate with all of the test variables with the exception of append, replace, and prepend actions which cannot interoperate with the HTTP version, HTTP method, and/or the client IP address. Note also that a cookie header cannot be operated on directly; a rule operates on the individual cookies. The relationship between the actions is such that all actions can interoperate with one another, however, if a rule contains the redirect action, that action must be last in the list of actions.

**Table 17: Request Translator Header Test Variable and Operator Matrix**

Test Variable	Test Operator					
	exists, not_exists	equals, not_equals, ci_equals, ci_not_equals	contains, not_contains, ci_contains, ci_not_contains	ends_with, not_ends_with, ci_ends_with, ci_not_ends_with	starts_with, not_starts_with, ci_starts_with, ci_not_starts_with	less_than, greater_than
url		X	X	X	X	
query_string	X	X	X	X	X	
request_header “ < header_name > ”	X	X	X	X	X	
request_cookie “ < header_name > ”	X	X	X	X	X	
http_request_version		X				
http_method		X				
src_ip, sip		X	X	X	X	
src_ip1, src_ip_2, src_ip_3, src_ip_4		X				X

The actions and their relationship to the test variables that are allowed are shown in Table 18.

**Table 18: Request Translator Header Action and Test Variable Matrix**

Action	Test Variable						
	url	query_string	request_header "< header_name >"	request_cookie "< header_name >"	http_request_version	http_method	src_ip, sip
insert_request_header, insert_reply_header	X	X	X	X	X	X	X
insert_request_cookie, insert_reply_cookie	X	X	X	X	X	X	X
update_request_header, update_reply_header	X	X	X	X	X	X	X
update cookie	X	X	X	X	X	X	X
delete header	X	X	X	X	X	X	X
delete cookie	X	X	X	X	X	X	X
append	X	X	X	X			
replace	X	X	X	X			
prepend	X	X	X	X			
redirect	X	X	X	X	X	X	X
forward	X	X	X	X	X	X	X
route_request	X	X	X	X	X	X	X

**Page Translator Application Rules**

Page Translator rules are designed to modify outgoing replies at either the header level or the content level. We will examine the header and content Page Translator rules separately.

**Page Translator Header**

The purpose of the Page Translator Header rules is to modify the outgoing HTTP reply headers based on certain test conditions. The variables, operators, and actions required for this type of rule are described as follows.



**NOTE:** Unlike the Request Translator Header, Page Translator Header rules can use both the request information and the reply information in test conditions.

The relationships allowed between the RTH test variables and the RTH test operations are shown in Table 19.

**Table 19: Page Translator Header Test Variable and Operator Matrix**

Test Variable	Test Operator					
	exists, not_exists	equals, not_equals, ci_equals, ci_not_equals	contains, not_contains, ci_contains, ci_not_contains	ends_with, not_ends_with, ci_ends_with, ci_not_ends_with	starts_with, not_starts_with, ci_starts_with, ci_not_starts_with	less_than, greater_than
url		X	X	X	X	
query_string	X	X	X	X	X	
any_request_header	X	X	X	X	X	
reply_header “ < header_name > ”	X	X	X	X	X	
request_cookie “ < header_name > ”	X	X	X	X	X	
reply_cookie “ < header_name > ”	X	X	X	X	X	
http_reply_code		X			X	
http_request_version		X				
http_method		X				
src_ip, sip		X	X	X	X	
src_ip1, src_ip_2, src_ip_3, src_ip_4		X	X	X	X	X

The actions and their relationship to the test variables that are allowed are shown in Table 20.

**Table 20: Page Translator Header Action and Test Variable Matrix**

Action	Test Variable								
	url	query_string	any_request_header	reply_header "< header_name >"	reply_cookie "< header_name >"	http_reply_code	http_request_version	http_method	src_ip_sip
cache	X								
insert_reply_header				X	X				
insert_reply_cookie				X	X				
update_reply_header				X	X				
delete_reply_header				X	X				
delete_reply_cookie				X	X				
append				X	X				
replace				X	X				
prepend				X	X				
forward	X	X	X	X	X	X	X	X	X
retry_request	X	X	X	X	X	X	X	X	X

Notice that all of the actions are based upon a generic reply header. The other test variables are solely available for test conditions; they cannot be altered by Page Translator Header rules. The relationship between the actions is such that all actions can interoperate with one another.

**Page Translator Content**

Page Translator Header (PTH) rules use all of the same test variables as Page Translator Content (PTC) rules, and also have the *content* test variable. Special consideration must be made when dealing with this variable relative to the other test variables, especially with regard to actions. The actions in a Page Translator Content rule operate only on the content variable and no other. The other variables are merely used for test conditions to determine if the content should be changed in some way.



**CAUTION:** Care must be taken when writing PTC rules as they change *any* matching string in the code.

For Page Translator Content rules, the only variable that may be operated upon in the actions is the content variable. All other variables may be used for reference in determining whether a rule is true (i.e., they can be used in test conditions), but not in actions.

The allowed relationships between the PTC test variables and the PTC test operations are similar to the header rules and are shown in Table 21.

**Table 21: Page Translator Content Test Variable and Operator Matrix**

Test Variable	Test Operator					
	exists, not_exists	equals, not_equals, ci_equals, ci_not_equals	contains, not_contains, ci_contains, ci_not_contains	ends_with, not_ends_with, ci_ends_with, ci_not_ends_with	starts_with, not_starts_with, ci_starts_with, ci_not_starts_with	less_than, greater_than
url		X	X	X	X	
query_string	X	X	X	X	X	
any_request_header	X	X	X	X	X	
reply_header “ < header_name > ”	X	X	X	X	X	
request_cookie “ < header_name > ”	X	X	X	X	X	
reply_cookie “ < header_name > ”	X	X	X	X	X	
http_reply_code		X			X	
http_request_version		X				
http_method		X				
src_ip, sip		X	X	X	X	
src_ip1, src_ip_2, src_ip_3, src_ip_4		X	X	X	X	X
content			X			

The actions and their relationship to the test variables that are allowed are shown in Table 22.

**Table 22: Page Translator Content Action and Test Variable Matrix**

Action	Test Variable									
	url	query_string	any_request_header	reply_header "<header_name>"	request_cookie "<header_name>"	reply_cookie "<header_name>"	http_request_version	http_method	src_ip, sip	content
append										X
replace										X
prepend										X
retry_request <sup>1</sup>	X	X	X	X	X	X	X	X	X	X

1. For the `retry_request` action to work correctly with Page Translation Contents, the factory setting `fc1` must be explicitly enabled (it is disabled by default). Contact your Juniper Administrator.

## Importing Rule Sets

Once you have your rule set defined, you can import the set onto the DX appliance.

To import a rule set to a DX appliance, do one of the following:

- Use the `import` command:

```
dx% import ruleset <filename>
dx% write
dx% set server down
dx% set server up
dx% write
```

The `filename` is the complete path to the application rules file. For example, `tftp://IP_address/directory/filename`. With the import process, the application rules are parsed for syntax.



**CAUTION:** If you import a rule set with the same name as an existing rule set, the existing rule set is overwritten with the new rule set.



- Use the `capture` command, followed by the actual rules on a blank line (do not include quotes):

```
dx% capture file <rulesetname>
PTH: HTTP_request_version eq "1.1" and reply_header "Content-Type"
starts_with "text" then insert_reply_header "Vary" "User-Agent,
Accept-Encoding"
PTH: HTTP_request_version eq "1.1" and reply_header "Content-Type"
starts_with "application-x-javascript" then insert_reply_header "Vary"
"User-Agent, Accept-Encoding"
...
.
```

End the file with a period (.). This method is less desirable because a syntax check is not performed until the AppRule is activated.

## Binding Rule Sets to Clusters

---

Once your rule sets have been imported to your DX appliance, you must bind the rule sets to clusters as appropriate for their purpose.

To apply a cluster to a rule set, enter:

```
dx% set cluster <name> apprule ruleset <ruleset_filename>
dx% write
```

You may also wish to configure a “high-water mark” for the number of bytes that are stored for a POST request to be retried. If the POST data exceeds this value, then the data is released and the retry mechanism is disabled for this request. The original request will proceed.

To set this limit, enter:

```
dx% set cluster <name> apprule limit retrypost <number>
dx% write
```

## Enabling OverDrive

---

Now that you have written your application rules, combined them into rule sets, imported and assigned them to clusters on the DX appliance, you can enable the OverDrive feature.

To enable apprule operation for a cluster, enter:

```
dx% set cluster <name> apprule enabled
dx% write
```

If you wish to disable apprule operation for a cluster, enter:

```
dx% set cluster <name> apprule disabled
dx% write
```

## Modifying Rules

---

You can modify specific application rules within a rule set and change or clear the rule set applied to a particular cluster.

To change a rule:

1. Open the application rules file containing the rule.
2. Edit the specific rule accordingly.
3. Disable OverDrive.
4. Reimport the rule set (remember that the existing rule set with the same name is overwritten).
5. Re-enable OverDrive.

To clear the rule rule set bound to a cluster, enter:

```
dx% clear cluster <name> apprule ruleset <ruleset_filename>
```

To change the rule set bound to a cluster:

1. Clear the rule set bound to the specific cluster.
2. Assign the new rule set using the `set cluster aprrule ruleset` command.

## Viewing Application Rules on the DX Appliance

---

These commands are used to show how the OverDrive feature is configured:

```
show cluster <name> aprrule ruleset
show cluster <name> aprrule status
show cluster <name> aprrule
```

These commands are used to display the AppRule configuration settings. They are used to display AppRule statistics. In each of the statistical DXSHELL commands shown, “M” represents the rule number:

```
show cluster <name> aprrule stats [all]
show cluster <name> aprrule stats rs [M|all]
show cluster <name> aprrule stats rth [M|all]
show cluster <name> aprrule stats pth [M|all]
show cluster <name> aprrule stats ptc [M|all]
```

The AppRule statistics are cleared when a new ruleset is applied (import ruleset/set server down/set server up).

This command displays all of the limit values:

```
show cluster <name> aprrule limit
```

This command displays the retrypost limit value:

```
show cluster <name> apprule limit retrypost
```

This command is used to display AppRule logging information.

```
show log apprule
```

These commands are explained in detail in the *Command Line Reference* manual.

## Limitations When Applying Application Rules

---

This section describes the various implications resulting from limitations that exist when using Application Rules. It contains the following sections:

- “Application Rules and Latency” on page 321
- “Displaying Rules” on page 322
- “User Data Parsing” on page 322
- “Using Prepend, Append, and Replace (PAR) Actions” on page 323

### Application Rules and Latency

Increasing the number of Application Rules necessarily increases the latency of a request or reply. The amount of added latency varies widely based on the type of rule (its test conditions and actions), and the number of rules applied. For example, case-insensitive searches take longer than case-sensitive ones. Clusters operating on thousands of rules execute more slowly than clusters with just a few rules. In general, higher performance can be obtained using the following general principles:

- Rules are executed essentially in a linear fashion, so place the popular rules near the top of the rule set list.
- Use case-sensitive searches whenever possible.
- The operators `starts_with` and `ends_with` should be used in preference to the `contains` operator, but use the `contains` operator if it means fewer individual rules.
- When using the `contains` operator, use the longest possible string for a resulting match, but when using `starts_with` and `ends_with` operators, use the shortest search string.
- Avoid the use of the wildcard whenever possible.
- Reduce the rule count by using the “and” keyword to join test conditions and actions whenever possible.

## Displaying Rules

In some cases, the DXSHELL (CLI) displays the rule that you entered back to you. Because of the way that rules are stored internally, what the system returns to you may not be precisely the way you entered it, although the effect is the same. The variations are as follows:

- A redirect/reply 302 command has its host and URL information combined together as a single string.
- All redirect rules are displayed as a reply 302.
- All rule keywords are displayed in lower case, effectively ignoring the way that you entered it.
- All test operator shorthand notation is shown in expanded format. For example, “eq” would be displayed as “equals.”

## User Data Parsing

There is a limit to how comprehensively data entered by a user is parsed for accuracy. The limits are:

- The `src_ip` (client IP address) value is constrained only to numbers and periods, but there is no check as to the validity of the entry. This is due to the additional complexity of checking a valid entry based against operators that allow partial matching (such as the `starts_with` operator).
- No checking is performed to ensure that a valid HTTP header name corresponds accurately to either a request header or a reply header. For example, if a user enters “request\_header “Server” ci\_contains “IIS,” this will not be flagged as an error, though in practice it actually is.

## Using the Forward Action with PTH Rules

When writing PTH rules containing the `forward` action, remember the following:

- The DX appliance only sends one request at a time (as opposed to in a pipe-lined fashion) to ensure that subsequent requests are directed to the same target TCP connection unmodified.
- If cache is enabled, then the `forward` action does not execute. The PTH rule statistics are incremented, and a log alert is generated when this occurs.
- If Web logging is enabled and the connection drops to "forward" mode, a Web log is not generated as the DX cannot whether or not an HTTP reply was returned.

## Using Prepend, Append, and Replace (PAR) Actions

The `prepend`, `append`, and `replace` (PAR) actions have unique inter-operations with test conditions, especially when the `term` keyword is used within the action. Whenever the `term` keyword is employed in a PAR operation, the variable that is being updated must have a corresponding test condition with that same variable, and must use one of the test operators shown in Table 23, depending upon the rule type.

**Table 23: PAR Test Operators**

Rule Type	Variable	Valid Test Operators
Request Translator Header	<code>url</code> <code>query_string</code> <code>request_header</code> <code>request_cookie</code>	<code>(ci)_contains</code> <code>(ci)_ends_with</code> <code>(ci)_starts_with</code>
Page Translator Header	<code>reply_header</code> <code>reply_cookie</code>	<code>(ci)_contains</code> <code>(ci)_ends_with</code> <code>(ci)_starts_with</code>
Page Translator Content	<code>content</code>	<code>(ci)_contains</code>

If the `term` keyword is not used in a PAR action, then the test conditions within that rule do not need to reference the same variable used in the PAR action.

For example:

```
RTH: url ends_with ".jpg" then prepend request_header "Host" "image-"
```

This rule does not use the `term` keyword and the test condition does not need to reference the `request_header "Host"` variable. This is a valid rule.

```
RTH: request_header "User-Agent" exists then replace request_header "User-Agent" term "Mozilla/8.0"
```

This is not a valid rule. While it does reference the `request_header "User-Agent"` variable in the test condition, it does not use a valid test operator as shown in Table 23. As such, there is no `term` to do a replace against. The entire `request_header "User-Agent"` value is in effect the `term`; therefore the rule should be rewritten without the `term` keyword as:

```
RTH: request_header "User-Agent" exists then replace request_header "User-Agent" "Mozilla/8.0"
```

This is now valid because the entire User-Agent header value is overwritten with `"Mozilla/8.0"`.

Note also that when the same variable appears more than once in the test conditions, the linkage between the PAR operation and the test condition is always to the FIRST test condition that meets the criteria shown in Table 23.

For example:

```
RTH: url ends_with ".gif" and url starts_with "/images" then prepend url term "/gif_repository"
```

This rule is probably NOT what the user intended because if the test conditions succeed then the sub-string term “.gif” will be prepended with the “/gif\_repository”, and not the “/images” sub-string. This is because the url “term” that was linked to the PAR operator was the first test condition in the rule that matched your criteria. In this case, it was the one that tests for the URL ending with “.gif”. To correct this rule, you can simply reverse the ordering of the test conditions:

RTH: url starts\_with “/images” and url ends\_with “.gif” then prepend url term “/gif\_repository”

Now you can cause a linkage between the PAR operation and the test condition that is looking at the start of the url that would result in the URL being “/gif\_repository/images ”; the correct behavior.

The characters shown in Table 24 are allowed for various PAR strings.

**Table 24: Allowable PAR String Variables**

Variable	Characters
content	A-Z, a-z, 0-9, space, ',', ':', '.', '!', '@', '#', '\$', '%', '^', '&', '(', ')', '=', '+', '[', ']', '{', '}', ' ', ';', '"', '<', '>', '?', '~', '/', '_', '-', '*'.  Escaped characters are: tab '\t', newline '\n', carriage return '\r', backslash '\\', double quote '\"'.  Note that the wildcard character used in test conditions is not escaped as it has no special meaning in a PAR string.
url	a-z, A-Z, 0-9, '~', '%', '.', '/', '_', '-'
query_string variable	a-z, A-Z, 0-9, '=', '+', '&', '?', ',', '~', '%', '.', '/', '_', '-'
header variable	a-z, A-Z, 0-9, '~', '%', '.', '>', '+', '<', '(', ')', '/', ',', '_', '-', '*', '?', '&', '\$', '\', '='

### Source IP Filtering

When writing rules using the octet-based Source IP variables, it is sometimes necessary to use the opposite test condition to achieve the desired results. For example, if you want to include only the range:

192.168.0.1 to 192.168.10.50

The first two octets can be satisfied by simply using the “equals” operator to get a valid match. However, the third octet requires that we use the “less\_than” operator to get what we want:

... src\_ip\_3 less\_than “11” ...

This implies that any value less than 11 is okay, which satisfies our example criteria. The last octet simply requires:

... src\_ip\_4 greater\_than “0” and src\_ip\_4 less\_than “51” ...

## Logging

---

Request Sentry rules require that logging be maintained whenever a rule is flagged with the `log` action. The logging data is sent to a new log type; “apprule” (as opposed to the “system” log). The log level is alert (ALRT) with the following format:

```
[<timestamp>][S:<source_ip>][D:<vip>][<rule_id>][<request>]
```

where:

- `<timestamp>` is the time the log entry is made in HH:MM:SS-YYYYMMDD format.
- `<source_ip>` is the client's IP address in the format AAA.BBB.CCC.DDD
- `<vip>` is the virtual IP address and port of the cluster in the format AAA.BBB.CCC.DDD:PPPP
- `<rule_id>` is the applications rule ID in the format ARID: AAA-BBBB-CCC
- `<request>` is as much of the request as can be reproduced given the constraints of the maximum logging length for an individual entry. The format is:

```
<method> <url> <protocol_version> <headers>
```

For example:

```
[16:07:22-20030901][S:155.12.33.234][D:19.84.128.12:80][ARID:
023-4555-121][GET /index.html HTTP/1.1 Host:ww]
```

For additional information on the logs, refer to “Syntax of the Log Entries” on page 128.

## Application Rule Scenarios

---

This section describes how Application Rules can be used in various scenarios.

### Route Request Application Rules

The `route_request` AppRule add a new action for the Request Translator Header (RTH) to allow users to route a request if an incoming request meets a test condition. You can specify the individual target host, a list of target hosts, or a group of target hosts using these criteria:

- If specified, the RTH rules are evaluated for every incoming request.
- Routing decisions are based upon examining the client request headers to determine which server is appropriate to handle the request.
- If an individual target host is specified for routing requests, load balancing will not be performed. This affects request distribution and means that some hosts may get more traffic than others.

- If a list of target hosts is specified, then “Fewest Outstanding Requests” load balancing is applied across the target hosts within the list.
- Route Requests supersede Sticky load balancing in a cluster.

The suggested request routing syntax is:

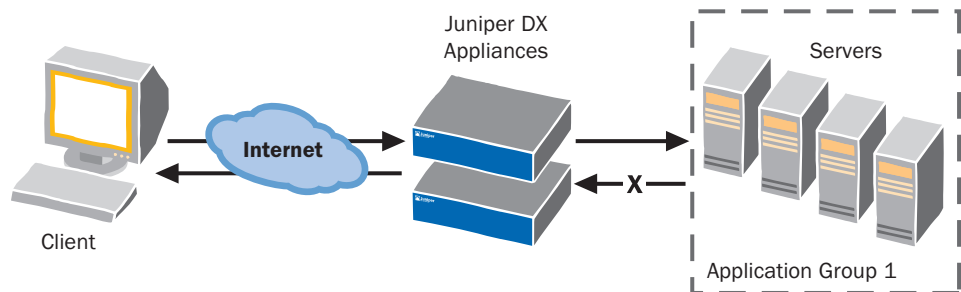
```
RTH: route_request target_host "<ip:port>"
RTH: route_request target_host "<ip1:port1>" ["<ip2:port2>"] ...
["<ipN:portN>"]
```

### Request Retry, Alerting, and Log (Transaction Assurance) AppRules

Request Retry AppRules allow you to specify retries for unsuccessful HTTP requests. The retries may be based upon the response code for the requests and are testable using DX OverDrive application rules. This feature is used by customers who require a mechanism for adding reliability to HTTP methods.

Lost HTTP responses have a negative impact on not only the end-user who experiences the request failure, but also the business itself as contributions from e-commerce fail to live up to expectations. Retry semantics using the DX OverDrive application rules to the target host help to fix the responses that are lost between the DX and the backend application. This application could be a Web, application, database, or an integration server. Refer to Figure 65.

**Figure 65: Request Retry Example**



The HTTP protocol is a synchronous protocol which requires that a client request<sup>TM 015</sup> completes with a reply (from the origin server, gateway, proxy, or an intermediary) that indicates the success or failure of the request. However, no reliability semantics are built into the HTTP protocol. Since the DX appliance is an intermediary in the HTTP request and response loop, it can initiate a “request retry” for failed requests.

Request Retry AppRules:

- Add a new action for Page Translator Header (PTH) and Page Translator Content (PTC) that allows you to retry a request if the response for a previous request meets a test condition.
- Add the ability to specify maximum number of retry attempts.
- If specified, the PTH or PTC rules are evaluated after every failed attempt.
- Add the ability to specify whether all retries are to one target host, or distributed through the list of target hosts up to the maximum retry limit.



- Add the ability to specify the logging of retry attempts. The logging, if specified, is done after every retry request.
- Makes the AppRule log entries available to an external alerting mechanism (like swatch) for administrator alerts.
- Adds the ability to log failures without retrying the request.

Request Retry syntax is:

```
PTH or PTC: retry_request [same | nosame | all] "number" and log
```

The default is the same target host "number" of times.

You can also set a value that acts as a "high-water mark" for the number of bytes that will be stored for a POST request to be retried by typing the command:

```
dx% set cluster N apprule limit retrypost <number>
```

If the POST data exceeds this value, then the data is released and the retry mechanism is disabled for this request. The original request will proceed.

If a value of zero is specified, then there is no limit imposed on the POST data amount. This is **very dangerous** since it allows a single user to issue a single request and use all of the resources on the box. The default value is 32768 kBytes. Most POST requests are typically less than 2 kBytes, so there should not be any problems with the default range limits. An upper limit of 100 MBytes is provided for installations that demand maximum flexibility.



**NOTE:** For the `retry_request` action to work correctly with Page Translation Content, the factory setting `fc1` must be explicitly enabled (it is disabled by default). Contact your Juniper Administrator.

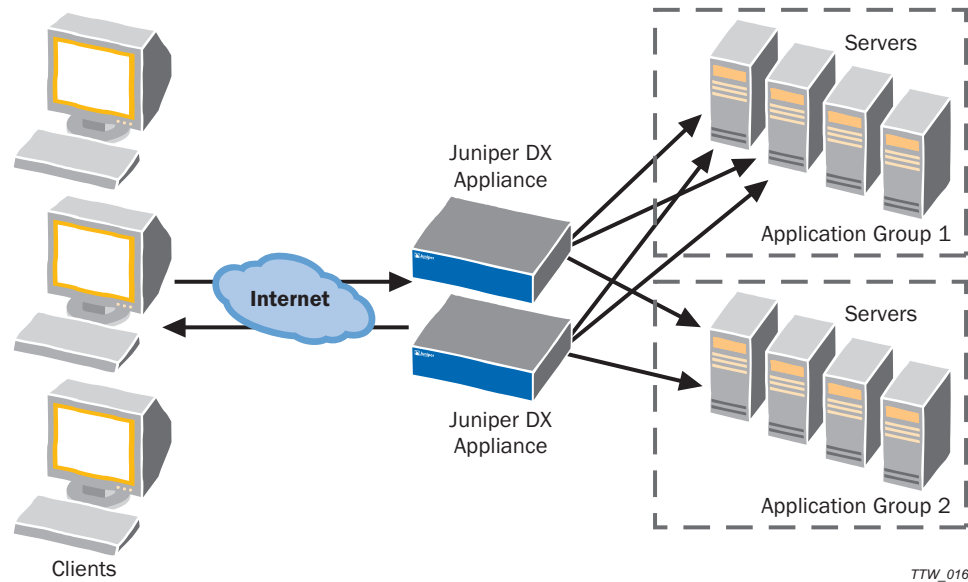
---

## Request Routing Application Rules

Data centers, Web servers, application servers, or integration servers are typically configured to provide distinct services such as payroll, billing, supply-chain integration, etc. A specific server, a list of servers, or even servers collectively referred to as a group, may provide these services. Request routing and dispatch at Layer 7 based upon user-defined information extends the OverDrive functionality to allow users to specify rules that control how requests are routed to user specified targets.

An example of this is a customer care application that routes a client's customer service telephone call to a particular call center (a bank of customer care agents) based upon the caller status (gold, platinum, or executive platinum.) The status can be provided as part of the initial HTTP response, and may be used on all subsequent responses for that session. Refer to Figure 66.

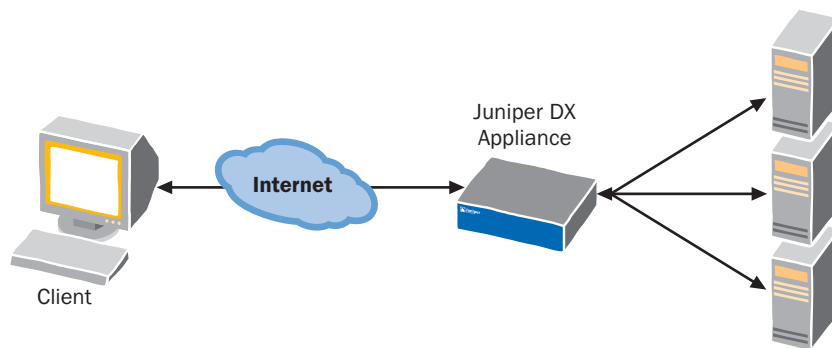
**Figure 66: Request Routing Example**



**Usage Scenario**

An example of routing to a particular target host is a site where target hosts respond with a session ID in the URL that includes an identifier (target host name, IP or custom information) when processing the initial HTTP request from a client. All subsequent requests from that particular client in that session would be routed to the initial target host (or group of hosts) that handled the initial call. This is a dynamic processing event and cookies are not used to make route determinations. Refer to Figure 67.

**Figure 67: Request Routing Usage Example**



## Request Sentry Examples

Table 25 shows some examples of Request Sentry AppRules that should help explain the AppRule grammar.

**Table 25: Request Sentry Examples**

Example	Description
RS: url length_greater_than "4096" then close_conn FIN and log	This example checks to see if the url exceeds 512 bytes and if so, close the connection by sending the client a FIN packet and then logging the result.
RS: any_request_header length_greater_than "2048" then close_conn RST	This example checks to see if any headers are longer than 2,048 bytes and if so, immediately closes the connection by sending a RST packet to the client.
RS: request_cookie "session_id" not_exists then redirect "https://www.myserver.com" "/login.cgi"	This example determines if the user has a session_id. If they do not, then they are redirected to the SSL /login.cgi URL on the server www.myserver.com.
RS: ssl_version eq "SSLv3" then redirect "https://www.newsite.com" "/login.cgi"	This example determines if the user is using SSL Level three, and if he is, redirects the request to the Web site: <a href="https://www.newsite.com/login.cgi">https://www.newsite.com/login.cgi</a> . The ssl_version test supports the test operators eq, not_eq, contains, not_contains, ends with, not_ends with, starts with, and not_starts with. The value for ssl_version is case-sensitive, and must be entered in the form: SSLv2, SSLv3, or TLSv1 instead of sslv2, sslv3, or tlsv1.
RS: ssl_ciphersuite eq "DES-CBC3-SHA" then redirect "https://www.newsite.com" "/login.cgi"	This example determines if the user is a specific suite of ciphers, and if he is, redirects the request to a different Web site. ssl_ciphersuite supports the test operators eq, not_eq, contains, not_contains, ends with, not_ends with, starts with, and not_starts with. The ciphersuites that are allowed are shown in Table 1 on page 209.
RS: ssl_cipher_bits eq "128" then redirect "https://www.newsite.com" "/login.cgi"	This example performs a specific action depending on the size of the key. ssl_cipher_bits supports the test operators less_than and greater_than only.
RS: src_ip_1 not_equals "192" and src_ip_2 not_equals "168" and src_ip_3 greater_than "254" and src_ip_4 greater_than "10" and src_ip_4 less_than "1" then close_conn rst	In this example, you only allow clients with IP addresses ranging from 192.168.1.1 to 192.168.254.10 to connect. All other clients are rejected abruptly with a TCP 'RST'

## Request Translator Examples

Table 26 shows some examples of Request Translator Header AppRules. The last two examples were taken from a sample mappings file.

**Table 26: Request Translator Examples**

Example	Description
RTH: url eq "/" then replace url "/pages/top_page.html"	This example will replace the url from being "/" to becoming "/pages/top_page.html" if the url (i.e., URI) exactly matches the value "/".
RTH: url eq "/autos" and request_header "Host" eq "www.myserver.com" then replace url "/" and update_request_header "Host" "autos.myserver.com"	This example will modify the "Host" header from "www.myserver.com" to "autos.myserver.com" if the "Host" header exactly matches the value "www.myserver.com" and the URI being requested is exactly "/autos". Note that in this example, we have placed the action on a separate line. This is perfectly legal since whitespace is ignored.
RTH: url ends_with ".jsp" then update_request_header "Host" "jspserver.myserver.com"	This example updates the "Host" header to have the value jspserver.myserver.com if the url ends in ".jsp".
RTH: request_header "Host" eq "motorway.dailybulletin.com" then replace url "/Stories/0,1413,250~25660~,00.html"	This is an example of a URL rewrite.
RTH: request_header "Host" eq "www.dailynews.com" and url eq "/motorway" then update_request_header "Host" "motorway.dailynews.com" and replace url "/Stories/0,1413,245~25661~,00.html"	This is an example of a URL rewrite and updating of the Host header.
RTH: url ends_with "gif" then route_request target_host "192.168.0.2:80"	This example looks for URLs that end with the file type "gif" and routes those requests to a specific host.
RTH: url starts_with "/images" then route_request target_host "192.168.0.2:80" "201.201.0.2:80" "198.168.6.2:80"	This example looks for urls that start with the file path of "/images" and reroutes those requests to one of three specified hosts.
RTH: src_ip_1 equals "10" and src_ip_2 equals "10" and src_ip_3 equals "0" then reply 302 "http://internal-apps.mycompany.com" "/login"	This example redirects all incoming requests from the 10.10.0/24 subnet to the login page of an internal application Web site. Note that you could have just as easily set up this rule using the traditional 'src_ip' variable like this:  RTH: src_ip starts_with "10.10.0" reply 302 "http://internal-apps.mycompany.com" "/login"

## Request Retry Examples

Table 27 shows some examples of Request Retry AppRules.

**Table 27: Request Retry Examples**

Example	Description
PTH: <code>http_reply_code starts_with "5" then retry_request "3" times same and log</code>	<p>In this Page Translator Header (PTH) example, if an HTTP request fails with a reply code of 5xx, then retry the request to the same target host (in the cluster where the earlier attempt failed) three more times and log. Other target hosts in the cluster will not be attempted at all.</p>
PTC: <code>content ci_contains "UNKNOWN" then retry_request "3" times nosame and log</code>	<p>In this Page Translator Content (PTC) example, the case-insensitive match of the reply content for the word "UNKNOWN" triggers a retry to the subsequent target host in the cluster where the earlier attempt failed. If that attempt fails, move to the next target host in the cluster.</p> <p>Specifying "nosame" means that the initial target host that failed the attempt is never retried. For example, if there are three target hosts (A, B, and C) in the cluster and target host A failed the initial request, target host B is tried first once, then target host C is tried once, then target host B is tried again for a retry count of 3. Note that the target host A was never retried.</p>
PTC: <code>content ci_contains "UNKNOWN" then retry_request "3" times all and log</code>	<p>In this Page Translator Content (PTC) example, the case-insensitive match of the reply content for the word "UNKNOWN" triggers a retry to the next target host in the cluster where the earlier attempt failed. If that attempt fails, the retry moves to the subsequent target host in the cluster.</p> <p>Specifying "all" means that the initial target host that failed the attempt is retried when the other target hosts in the cluster have been attempted. For example, if there are two target hosts (A and B) in the cluster and target host A fails the initial request, the target host B is tried first, then target host A, and then target host B again for a retry count of 3.</p>

## Request Routing Examples

Table 28 shows two Request Routing examples.

**Table 28: Request Routing Examples**

Example	Description
RTH: <code>url ends_with "gif" then route_request target_host "192.168.0.2:80"</code>	<p>In this Request Translator Header (RTH) example, if an HTTP request is to fetch a page and the URL ends with gif, then the request is served by the target host 192.168.0.2:80.</p>
RTH: <code>url starts_with "/images" then route_request target_host "192.168.0.2:80" "201.201.0.2:80" "198.168.6.2:80"</code>	<p>In this Request Translator Header (RTH) example, if the URL requested begins with /images, then the request service is load balanced across the three target hosts specified using Juniper's Fewest outstanding Requests algorithm.</p> <p>Note: Route Request overrides any sticky load balancing.</p>

## Page Translator Examples

Table 29 shows various Page Translator examples.

**Table 29: Page Translator Examples**

Example	Description
PTH: <code>http_reply_code</code> starts with "5" then <code>retry_request</code> same "3" and <code>log</code>	In this Page Translator Header (PTH) example, if an HTTP request fails with a reply code of 5xx, the DX appliance retries the request to the same target host in the cluster where the earlier attempt failed up to three more times and logs the results. The DX appliance will not try to route the request to other target hosts in the cluster.
PTH: <code>url</code> eq "/" then <code>update_reply_header</code> "Server" "Apache 2.0.47 (Amiga)" "Netscape-Enterprise/4.1" "GWS/2.1"	This Page Translator Header example will essentially update every outgoing request's "Server" header value with one of the three values shown above in a round-robin fashion. This effectively accomplishes the notion of "server cloaking" (or perhaps, server obfuscation from programs or people trying to determine your server type).
PTH: <code>url</code> eq "/" then <code>delete_reply_header</code> "Server"	This would remove the "Server" header from the outgoing reply making it more difficult to tell what kind of origin server is in operation.
PTH: <code>url</code> eq "/login.cgi" and <code>request_cookie</code> "login_challenge" "0" then <code>insert_reply_cookie</code> "login_challenge" "1" "login.myserver.com" "/" secure	This example would update the login_challenge cookie from the value 0 to the value 1 on the outgoing reply. The cookie would only be sent by the client whenever connecting to the server login.myserver.com with an SSL connection. The cookie does not have an expiration date, so it will be discarded by the client when the browser application closes. Note that the insert_reply_cookie action is used instead of update_reply_cookie as we are assuming that the origin server is not sending this cookie for this reply but did so at some time prior.
PTH: <code>url</code> <code>ci_contains</code> "/" then <code>insert_reply_cookie</code> "visit" "yes" ".myserver.com" "/" "3600" secure	This would set a cookie visit to a value yes. The cookie expires 3600 seconds from the time the response is sent to the client.
PTC: <code>content</code> contains "http://*.juniper.net" then <code>replace content term</code> "http://gateway.juniper.net/"	In this Page Translator Content example, the content term "http://*.juniper.net" will be prepended with "http://gateway.juniper.net/" wherever it is found in the response. Note that the search is a case-insensitive "contains" search.
PTH: <code>src_ip</code> starts_with "192.168" and <code>src_ip_3</code> greater_than "99" and <code>src_ip_3</code> less_than "105" and <code>src_ip_4</code> greater_than "0" and <code>src_ip_4</code> less_than "255" then <code>insert_reply_header</code> "X-Powered-By" "Juniper Web I/O Accelerator"	This trivial example shows how the traditional <code>src_ip</code> test variable can be used in conjunction with the octet-level test conditions to create a rule that tags all replies to clients 192.168.99.1 to 192.168.104.254 with an additional header, "X-Powered-By".
PTC: <code>content</code> <code>ci_contains</code> "</body>" then <code>prepend content term</code> " Powered by <b>Juniper Networks</b> "	This example will effectively place a text footer at the end of every HTML page. If you wanted to restrict this to only the home page, you might do something like this:

**Table 29: Page Translator Examples (continued)**

Example	Description
PTC: content ci_contains “</body>” and url eq “/” then prepend content term “ Powered by <b>Juniper Networks</b> ”	You now have two test conditions in operation; one looking for the </body> tag, and one looking for the URL being the home page (“/”). If both are true, then the prepend operation will occur.
PTC: content contains “<%AddBanner%>” then replace content term “<div align=center><a href=http://www.doubleclick.net/adsys.cgi?redir=http://www.dell.com&adsrc=www.mysite.com><imgsrc=http://adserv.doubleclick.net/default_leader.gif alt=\”Click here!\” border=0 width=728 height=90></a></div>”	This example shows how to use the Page Translator Content rule as a special tag replacement mechanism. Wherever the special tag < %AddBanner% > is found, it is replaced with an HTML snippet that displays a banner ad. Note that for speed, the “contains” search is case-sensitive.
PTC: content ci_contains “UNKNOWN” the retry_request nosame “3” and log	<p>This Page translator Content example shows that the case-insensitive match in the reply content of the word “unknown” triggers a retry to the subsequent target host in the cluster where the earlier attempt failed. If that attempt fails too, the DX appliance moves on to the next host in the cluster.</p> <p>Specifying “nosame” means that the initial target host that failed the attempt is never retried. For example, if there are three target hosts in the cluster (A, B, and C), and target host A fails the initial request, the DX appliance tries target host B once first, then tries target host C once, and then tries target host B again for a retry count of three. Note that host A was never retried.</p>
PTC: content ci_contains “UNKNOWN” the retry_request all “3” and log	<p>This Page translator Content example shows that the case-insensitive match in the reply content of the word “unknown” triggers a retry to the subsequent target host in the cluster where the earlier attempt failed. If that attempt fails too, the DX appliance moves on to the next subsequent target host in the cluster.</p> <p>Specifying “all” means that the initial target host that failed the attempt is retried when all of the other hosts in the cluster have been attempted. For example, if there are two target hosts in the cluster (A and B), and target host A fails the initial request, the DX appliance tries target host B once first, then tries target host A again, and then tries target host B again for a retry count of three.</p>





## Chapter 19

# Configuring Global Server Load Balancing

This chapter provides procedures for configuring Global Server Load Balancing on the DX Application Acceleration Platform. Currently, this feature can only be configured using the DXSHELL command line interface. Only users with an administrator or network administrator role have access to the GSLB configuration commands. All users can access the GSLB `show` commands.



The GSLB feature requires a license. Contact your local Juniper Sales Representative for information and to obtain a license.

---

This chapter contains the following topics:

- “GSLB Configuration Task Flow” on page 336
- “Configuring the GSLB Agent” on page 336
- “Defining the GSLB Remote Nodes” on page 337
- “Configuring a GSLB Resolver” on page 338
- “Synchronizing Your GSLB Configuration” on page 349
- “Removing Configuration Information” on page 349
- “Show Configuration Commands” on page 349
- “Statistics Commands” on page 350
- “Deployment Scenarios” on page 351
- “GSLB Failover” on page 355

For a discussion of GSLB and how it works with the DX Application Acceleration Platform, see “Global Server Load Balancing” on page 66.

## GSLB Configuration Task Flow

---

The following steps illustrate the basic steps needed to deploy GSLB with the DX platform:

1. Configure the GSLB agent on the DX appliance acting as the master.
2. Define the GSLB remote nodes on the master by specifying the IP address and port for each GSLB agent.
3. Configure a GSLB resolver on the master:
  - a. Specify the listen IP address and port of the GSLB resolver.
  - b. Specify the target IP address and port of the GSLB resolver.
  - c. Add GSLB groups to the resolver. Configure DNS and load-balancing parameters.
  - d. Add members to the groups. Configure IP addresses to be load balanced and the remote node association to each member.
  - e. Optionally, configure the Local (internal) DNS server.
4. Configure the agent, remote nodes, and resolver on each of the other DX appliances that are participating in the GSLB process.

## Configuring the GSLB Agent

---

When using metric-based load balancing, the GSLB agent must be configured on all DX appliances participating in GSLB. One of the roles of the GSLB master is to collect performance metrics from each agent.

To configure the GSLB agent on each remote node:

1. Specify the IP address of the GSLB agent. The IP address must be a unique VIP, and cannot conflict with a VIP used by any other subsystem.

```
dx-1% set gslb agent listen vip <IP>
```

2. Specify the port number (up to 65535) of the GSLB agent. The default port is 3587.

```
dx-1% set gslb agent listen port <N>
```

3. Optionally, enter a key used to encrypt the metrics and other messages sent via UDP. Repeat this command to specify multiple keys. For example, if there are two GSLB masters using this remote node, each master can use a different key.

```
dx-1% set gslb agent encryption key
```

```
New key: enter a key
```

```
Retype new key:
```

Enable message encryption (disabled by default).

```
dx-1% set gslb agent encryption enabled
```

4. Enable the GSLB agent to respond to requests from the GSLB master (disabled by default) and save your configuration.

```
dx-1% set gslb agent enabled
GSLB agent started.
(*) dx-1% write
Writing configuration.
Done.
dx-1%
```

5. Repeat these steps for the DX appliances at each site.

## Defining the GSLB Remote Nodes

---

When the GSLB agent has been configured on the DX appliance, define the GSLB remote nodes by specifying the IP address, port and, optionally, encryption.

To define a GSLB remote node:

1. Add a remote GSLB node. If you omit the name, a name is generated automatically. The keywords `all` and `internal` are reserved.

```
dx-1% add gslb remotenode <remotenodename>
```

2. Specify the listening IP address and port of a GSLB agent as well as encryption parameters if applicable.

```
dx-1% set gslb remotenode <remotenodename> agentip <IP>
dx-1% set gslb remotenode <remotenodename> port <N>
dx-1% set gslb remotenode <remotenodename> encryption key
New key: enter a key
Retype new key:
dx-1% set gslb remotenode <remotenodename> encryption <enabled|disabled>
```

3. Repeat step 2 for all other GSLB remote nodes acting as agents.
4. Specify the number of seconds that the GSLB master waits for a response from the GSLB agents.

```
dx-1% set gslb remotenode <remotenodename> timeout <seconds>
```

The `timeout` can range from one to 4294967295 seconds. If the `timeout` is exceeded, the node is assumed to be unavailable, and its metrics score is set to zero (refer to “Configuring Parameters for Metric-based Load Balancing” on page 342).

5. Save your configuration changes.

```
(*) dx-1% write
Writing configuration.
Done.
dx-1%
```

6. Repeat this procedure to add additional remote nodes.

## Configuring a GSLB Resolver

---

When all of the agents and remote nodes have been defined and configured, you can continue GSLB configuration by configuring the GSLB resolver. The GSLB resolver operates as a DNS proxy and can be configured to point to an external standalone DNS or to the internal DNS server. Multiple resolvers can be created, each listening on its own virtual IP. Each resolver contains one or more groups that contain multiple members.

This section contains the following procedures:

- “Configuring Resolver Basics” on page 338
- “Defining GSLB Groups” on page 339
- “Configuring the Local DNS Server” on page 346

### Configuring Resolver Basics

First configure the resolver on the DX appliance acting as the GSLB master and then configure the resolvers on the remote nodes (other DX appliances).

To configure a GSLB resolver:

1. Add a GSLB resolver. If you omit the name, a name is generated automatically. The keywords `all` and `localdns` are reserved.

```
dx-1% add gslb resolver <resolvername>
```

2. Specify the virtual IP of the GSLB resolver used for listening to public DNS requests.

```
dx-1% set gslb resolver <resolvername> listen vip <IP>
```

3. Specify the port number (up to 65535) of the GSLB resolver. The default port is 53, the standard DNS port.

```
dx-1% set gslb resolver <resolvername> listen port <N>
```

- Specify the target DNS server where DNS requests that are not load balanced are sent. Enter the IP address and port of a DNS server in the network, or enter `localdns` to use the internal DNS server.

```
dx-1% set gslb resolver <resolvername> target ip <IP:port|localdns>
```



See “Configuring the Local DNS Server (Optional)” on page 341 for further information about using the internal DNS server.

- Enable the GSLB resolver (disabled by default):

```
dx-1% set gslb resolver <resolvername> enabled
```

- Save your configuration changes.

```
(*) dx-1% write
Writing configuration.
Done.
dx-1%
```

- Repeat this procedure for the other DX appliances.

## Defining GSLB Groups

A GSLB group is a single DNS hostname that is mapped to a collection, or list, of IP addresses. When LDNS servers query the GSLB master for the hostname, the GSLB resolver responds to the query with the IP address from the list based on the specified load-balancing policy. To complete the GSLB configuration, you must define one or more GSLB groups that specify the IP addresses associated with the given DNS hostname, the load-balancing policy, and the metrics used for load balancing (if used).

To define a GSLB group:

- Add a GSLB group to a resolver. If you omit the group name, a name is generated automatically. The keywords `all` and `localdns` are reserved. The group name is used only for identification purposes and should not be confused with the DNS hostname.

```
dx% add gslb resolver <resolvername> group <groupname>
```

You can change a GSLB group's name using the `set gslb resolver <resolvername> group <groupname> name <name>` command.

- Specify a load balancing policy for the GSLB group. The default is “roundrobin”.

```
dx% set gslb resolver <resolvername> group <name> lba policy <policy>
```

The following table describes the available load balancing policies.

Policy	Description
roundrobin (default)	IP addresses in the group are returned in a sequential fashion, with each request getting the next IP in the group.
weightedroundrobin	Same as round robin, except that the weight assigned to each IP determines the number of times the IP is served for consecutive requests before the next IP is served.
random	IP addresses are returned in a random order.
fixed	IP addresses are returned in the order that they were added to the group.
forward	Requests are forwarded to the target DNS (no load balancing).
metric	Performance metrics collected from each GSLB node are used to determine which IP addresses to return. To specify the metrics used, refer to “Configuring Parameters for Metric-based Load Balancing” on page 342.

For all policies, pings are sent to each IP in the group, one per second. If the IP fails to respond to three pings in a row, it is removed from rotation until it responds to three consecutive pings.

- Consecutive requests from the same LDNS can be given the same GSLB node IP address if the requests occur within a specified number of seconds (one to 4,294,967,295 seconds). This affects all load balancing policies except “forward.” Disabling this policy may not be effective immediately if the LDNS has a local cache. Enable client to GSLB node persistence as follows:

```
dx% set gslb resolver <resolvername> group <name> lba sticky enabled
dx% set gslb resolver <resolvername> group <name> lba sticky timeout
<seconds>
```

Requests from different LDNS servers can be given the same IP address if they are all within a specified netmask. The default is 255.255.255.255 (each source IP address is treated individually).

```
dx% set gslb resolver <resolvername> group <name> lba sticky netmask
<netmask>
```

You can also specify the maximum number of entries (1 to 4294967295) in the sticky cache table. The default is 16384.

```
dx% set gslb resolver <resolvername> group <groupname> lba sticky max <N>
```

- Specify the fully-qualified hostname (FQHN) for the GSLB group.

```
dx% set gslb resolver <resolvername> group <name> dns hostname <FQHN>
```

- Specify the TTL for the DNS record returned in response to a hostname lookup request (1 to 4294967295 seconds). The default is 300.

```
dx% set gslb resolver <resolvername> group <name> dns TTL <seconds>
```

- Optionally, specify the following authentication parameters to assist LDNS servers in identifying authoritative name servers. The server and domain name of the authoritative server are used in the Authority section of the DNS response and a record is inserted into the Additional section of the DNS response providing the GSLB resolver's VIP and the IP address for the name of the authoritative server.

```
dx% set gslb resolver <resolvername> group <name>
dns authdomainname <FQDN>
dx% set gslb resolver <resolvername> group <name>
dns authservername <FQHN>
```

- Specify whether one or multiple IP addresses are returned with each DNS request.

```
dx% set gslb resolver <resolvername> group <name>
dns answermode <single | multiple>
```

If set to multiple, the order of the IP addresses depends on the load-balancing policy. When the `roundrobin` or `weightedroundrobin` policy is selected, the order is a snapshot of the current roundrobin ordering. When the `random` policy is selected, the order is random. When the `forward` policy is selected, this parameter is ignored and the response is determined by the target DNS. When the `metric` policy is selected, the order is determined by each GSLB group member's metric score.



If GSLB sticky is enabled (steps 3), you must specify `answermode` as `single`.

---

- Specify a failure IP address to send to the LDNS when all GSLB remote nodes are unavailable.

```
dx% set gslb resolver <resolvername> group <name> failip <IP>
```

- Save your configuration changes.

```
(*) dx% write
Writing configuration.
Done.
dx%
```

### Adding Members to a GSLB Group

A GSLB member is an IP address that is load balanced by a GSLB group.

To add a member to a GSLB group:

- Add a member. The maximum number of members per group is determined by the DX's License.

```
dx% add gslb resolver <resolvername> group <groupname> member <membername>
```

2. Specify an IP address for this group member. This is the address that is load balanced and returned in DNS queries. It can be arbitrary, but must be a valid IP address.

```
dx% set gslb resolver <resolvername> group <groupname> member <membername>
ip <IP>
```

3. Specify a weight for this group member. The weight is applied after load-balancing calculations are made, and is used to determine the member's final position in the load-balanced response. The weight can range from 1 to 100.

```
dx% set gslb resolver <resolvername> group <groupname> member <membername>
weight <N>
```

4. Specify the remote GSLB node to use to gather metric information for this member. `remotenodename` must be the name of a GSLB remote node already configured with the `add gslb remotenote` command.

```
dx% set gslb resolver <resolvername> group <groupname> member <membername>
remotenode <remotenodename>
```

5. Specify the DNS Time-to-Live parameter.

```
dx% set gslb resolver <resolvername> group <groupname> dns ttl <seconds>
```

6. Save your configuration changes.

```
(*) dx% write
Writing configuration.
Done.
dx%
```

7. Do one of the following:

- To configure metric-based load balancing, follow the procedure provided in the “Configuring Parameters for Metric-based Load Balancing” on page 342.
- To add another GSLB group to this resolver, repeat this procedure.
- To configure the internal DNS server, see “Configuring the Local DNS Server” on page 346.

### Configuring Parameters for Metric-based Load Balancing

You can configure one or more of the following metrics to fine tune load balancing across the DX appliances:

- Server connections
- SLB sessions
- Network interface usage
- Memory usage
- CPU usage



- Target host availability
- Round Trip Time (RTT) to the local DNS server (LDNS)

Each metric has a maximum (or minimum) value and an associated weight. If the weight is zero, the metric is not used for load balancing.

The GSLB master requests statistics from the GSLB remote nodes. The metric weights and collected values are used to calculate a total score for each GSLB remote node. If a remote node does not respond within the specified timeout (refer to “Defining the GSLB Remote Nodes” on page 337), the remote node receives a total score of zero for that request.

The GSLB remote node score and the weight specified for each DX appliance IP address is used to calculate a final score for each address. The final scores determine the IP address(es) returned to the LDNS.

To configure the metrics used for load balancing:

1. Specify how often (in seconds) the GSLB master requests statistics from the GSLB remote nodes. This does not apply to the RTT metric. RTT statistics are collected when the master receives a relevant request.

```
dx% set gslb resolver <resolvername> group <name> metric interval <seconds>
```

2. Specify the extent to which the collected statistics are smoothed out to alleviate the effects of sudden spikes in the data. Default is medium.

```
dx% set gslb resolver <resolvername> group <name> metric smoothing <low | medium | high>
```

3. Specify the appropriate weights and threshold values for each metric, as described in the following sections. Set the weight to zero on any metric that you do not want to use for load balancing.
4. Save your configuration changes.

```
(*) dx% write
Writing configuration.
Done.
dx%
```

5. Repeat the procedures for configuring a GSLB group for each group that is resolved by the GSLB master.

To restore the default metric settings, use the `set gslb resolver <resolvername> group <name> metric defaults` command.

**Server Connections**

This is the count of connections on a GSLB node. It takes into account all connections for clusters, forwarders, and redirectors on the node. It excludes health check connections.

1. Specify the maximum number of connections that the GLSB node may have to be considered available. If the number of connections exceeds this value, the IP addresses associated with the node are taken out of rotation. Set the maximum between zero and 4294967295. The default is zero.

```
dx% set gslb resolver <name> group <name> metric connections max <N>
```

2. Specify the weight (0 to 100) for this metric. Default is zero.

```
dx% set gslb resolver <resolvername> group <name> metric connections weight <N>
```

3. Return to step 4 in the “Configuring Parameters for Metric-based Load Balancing” section.

**SLB Sessions**

This is the number of Layer 4 sessions in the SLB service on a GSLB node.

1. Specify the maximum number of SLB sessions that on a GLSB node may have to be considered available. If the number of sessions exceeds this value, the IP addresses associated with the node are taken out of rotation. Set the maximum between zero and 4294967295. The default is zero.

```
dx% set gslb resolver <resolvername> group <name> metric sessions max <N>
```

2. Specify the weight (0 to 100) for this metric. Default is zero.

```
dx% set gslb resolver <resolvername> group <name> metric sessions weight <N>
```

3. Return to step 4 in the “Configuring Parameters for Metric-based Load Balancing” section.

**Byte Rate**

The byte rate is the kilobytes of network traffic handled by the GSLB node. The value is calculated by finding the busiest interface, and then comparing the input and output data rates. The higher of the two is then divided by the interface's maximum data rate to achieve a normalized percentage. The normalized value is independent of any artificial rate-limiting of interfaces based on the license key.

1. Specify the maximum allowable data rate (in bytes) on a GSLB node. If this value is exceeded, the IP addresses associated with the node are taken out of rotation. Append KB or MB to the maximum to indicate kilobytes or megabytes, respectively. Set the maximum between zero and 4294967295. The default is 125000000 bytes.

```
dx% set gslb resolver <resolvername> group <groupname> metric byterate max <N>
```

2. Specify the weight (0 to 100) for this metric. Default is zero.

```
dx% set gslb resolver <resolvername> group <name> metric byterate weight <N>
```

3. Return to step 4 in the “Configuring Parameters for Metric-based Load Balancing” section.

### **Memory Usage**

This is a percentage of the memory usage on a GSLB node.

1. Specify the maximum percentage of memory usage allowed before a GSLB node is considered to be unavailable (0 to 100). The default is 80 percent.

```
dx% set gslb resolver <resolvername> group <name> metric memusage max <N>
```

2. Specify the weight (0 to 100) for this metric. Default is zero.

```
dx% set gslb resolver <resolvername> group <name> metric memusage weight <N>
```

3. Return to step 4 in the “Configuring Parameters for Metric-based Load Balancing” section.

### **CPU Usage**

This is a percentage of the CPU usage on a GSLB Node.

1. Specify the maximum percentage of CPU usage allowed before a GSLB node is considered to be unavailable (0 to 100). The default is 80 percent.

```
dx% set gslb resolver <resolvername> group <name> metric cpuusage max <N>
```

2. Specify the weight (0 to 100) for this metric. Default is zero.

```
dx% set gslb resolver <resolvername> group <name> metric cpuusage weight <N>
```

3. Return to step 4 in the “Configuring Parameters for Metric-based Load Balancing” section.

### **Target Host Availability**

This is the percentage of the total unpaused target hosts that are available for all clusters and forwarders on the GSLB node.

1. Specify the minimum percentage of target hosts that must be available for the GSLB node to be considered available (0 to 100). The default is zero.

```
dx% set gslb resolver <resolvername> group <name> metric targethostavailability min <N>
```

2. Specify the weight (0 to 100) for this metric. Default is zero.

```
dx% set gslb resolver <resolvername> group <name> metric targethostavailability weight <N>
```

3. Return to step 4 in the “Configuring Parameters for Metric-based Load Balancing” section.

**Round Trip Time (RTT)**

When a GSLB master receives a DNS request from an LDNS, and the weight of the RTT metric is non-zero, the GSLB nodes are asked to calculate the round trip times to the LDNS (using ICMP pings).

To configure the RTT metric:

1. Specify the maximum acceptable RTT time (in milliseconds). If a node's RTT exceeds this value, the node is not a viable candidate. Set the maximum between one and 300,000. The default is 3 ms.

```
dx% set gslb resolver <resolvername> group <name> metric rtt
max <milliseconds>
```

2. Specify the number of seconds to wait for an RTT value from a GSLB node. Set the wait time between one and 300. The default is 15 seconds. Note that both the count and timeout settings affect the response time of the GSLB master.

```
dx% set gslb resolver <resolvername> group <name> metric rtt
timeout <milliseconds>
```

3. Specify the number of pings that each GSLB node sends to the LDNS perform. Pings are sent one second apart, and the average RTT is used. Set the number of pings between one and five. The default is 3 pings. Note that both the count and timeout settings affect the response time of the GSLB master.

```
dx% set gslb resolver <resolvername> group <name> metric rtt count <N>
```

4. You can use the same RTT value for all LDNS servers within a specified netmask. The default is 255.255.255.255 (the RTT is calculated for each LDNS).

```
dx% set gslb resolver <resolvername> group <name> metric rtt
netmask <netmask>
```

5. Specify the weight (0 to 100) for this metric. Default is zero.

```
dx% set gslb resolver <resolvername> group <name> metric rtt weight <N>
```

6. Return to step 4 in the “Configuring Parameters for Metric-based Load Balancing” section.

**Configuring the Local DNS Server**

The Local DNS Server function provided with the DX Application Acceleration Platform is a basic Domain Name System (DNS) server. The Local DNS server is not a fully-configurable server, but is intended instead to be used when you need rapid deployment of an easily available name server versus a complete set of DNS services. In GSLB applications, the DX platform acts as a proxy to DNS servers. configuring GSLB resolver to communicate with the LDNS provided on the DX platform is optional.

To configure the local DNS server:

1. Add a domain to the DNS Server. This adds a start of authority (SOA) record for the specified domain.

```
dx% add gslb localdns domain <domain>
```

2. Add one or more name server (NS) records for the domain.

```
dx% set gslb localdns domain <domain> ns <server name>
```

When `server name` does not end in a period "." (it is not fully qualified), the name server appends the domain name to it when responding to queries.

3. Add an address record for a host in the domain.

```
dx% set gslb localdns domain <domain> a <host> <ip>
```

When `host` does not end in a period "." (it is not fully qualified), the name server appends the domain name to it when responding to queries. There can only be one address record for a host in a domain; however, you can have multiple aliases.

4. Optionally, add one or more aliases (canonical name records) for a host in the domain.

```
dx% set gslb localdns domain <domain> cname <host> <alias>
```

The host must be one of the hosts for which an address record is already configured. If either `host` or `alias` does not end in a period "." (it is not fully qualified), the name server appends the domain name to it when responding to queries.

Some examples:

```
dx% set gslb localdns domain a.com cname www ftp
dx% set gslb localdns domain a.com cname www gopher
```

5. Add a pointer (PTR) record for an IP in the specified domain. This is used for reverse DNS lookups.

```
dx% set gslb localdns domain <domain> ptr <ip> <host>
```

If `host` does not end in a period "." (it is not fully qualified), the name server appends the domain name to it when responding to queries. There can be only one pointer record for an IP in a domain.

6. Add one or more mail exchange (MX) records to specify the name of the mail server(s) for the domain. Each server is assigned a different priority.

```
dx% set gslb localdns domain <domain> mx <mail server> <priority>
```

When `mail server` does not end in a period "." (it is not fully qualified), the name server appends the domain name to it when responding to queries. `priority` is a positive integer, with zero being the highest priority.

- Specify the Time to Live (TTL) for the specified domain. This determines how long a DNS record is cached before it is removed. It is applied to all Resource Records in a domain.

```
dx% set gslb localdns domain <domain> ttl <secs>
```

*secs* must be between 1 and 2147483647 seconds (inclusive). The default TTL value is 300 seconds.

- Specify the contact email for the domain. The contact email is not used by the name server, but is returned on request by DNS clients. The clients can then contact the administrator using this E-mail address, should a need arise.

```
dx% set gslb localdns domain <domain> contact <email>
```

*email* is specified in “name.domain” format. The default E-mail contact is “jnpr-dx.<hostname>”.

- Specify a sequence number for a domain (defaults to 1).

```
dx% set gslb localdns domain <domain> sequence number <N>
```

- Optionally, disable incrementation of the sequence number each time the domain is changed (enabled by default).

```
dx% set gslb localdns domain <domain> sequence autoincrement disabled
```

## Deleting Domains and Resource Records

To remove the domain and all its records:

```
dx% delete gslb localdns domain <domain>
```

To delete the specified name server record:

```
dx% clear gslb localdns domain <domain> ns <name server>
```

To delete the specified address record:

```
dx% clear gslb localdns domain <domain> a <host>
```

To delete the specified canonical name record:

```
dx% clear gslb localdns domain <domain> cname <host> <alias>
```

To delete the specified pointer record:

```
dx% clear gslb localdns domain <domain> ptr <ip>
```

To delete the specified mail exchange record:

```
dx% clear gslb localdns domain <domain> mx <mail server>
```

## Displaying the DNS Server Configuration

To display all the Resource Records for one or all the domains:

```
dx% show gslb localdns domain [domain | all]
```

## Synchronizing Your GSLB Configuration

---

The configuration of GSLB masters can be synchronized using the Configuration Synchronization feature. Use the `sync group <name> gslb` command to synchronize all GSLB parameters, including settings for the internal DNS server, GSLB resolvers, and GSLB groups.

## Removing Configuration Information

---

The following `clear` and `delete` commands can be executed by users with “Administrator” and “Network Administrator” roles.

To remove the IP address used when all GSLB nodes are unavailable:

```
dx% clear gslb resolver <resolvername> group <groupname> failip
```

To remove the Nth encryption key for a GSLB agent:

```
dx% clear gslb agent encryption key <N>
```

To remove the encryption key for a specific remote GSLB node:

```
dx% clear gslb remotenode <remotenodename> encryption key
```

To delete an entire group and its members:

```
dx% delete gslb resolver <resolvername> group <groupname>
```

To delete a remote node:

```
dx% delete gslb remotenode <remotenodename>
```

To delete a resolver:

```
dx% delete gslb resolver <resolvername>
```

## Show Configuration Commands

---

The following `show` commands can be executed by users with `administrator`, `network administrator`, or `network operator` roles. You can refine the output of each of these commands by adding additional optional parameters.

To show the GSLB agent configuration on the local DX:

```
show gslb agent
```

To show the internal DNS server configuration:

```
show gslb localdns
```

To show the configuration of one or all domains in the internal DNS server (default is all):

```
show gslb localdns domain [<name> | all>]
```

To show one or all remote node definitions on the GSLB master (default is all):

```
show gslb remotenode [<name> | all]
```

To show the configuration for one or all GSLB resolvers (default is all).

```
show gslb resolver [<name> | all]
```

## Statistics Commands

---

You can view GSLB statistics for an agent, remote node, resolver, or group.

### Agent Statistics

To view all agent statistics for a GSLB agent:

```
show gslb agent stats
```

To view a subset of the agent statistics that is updated every N seconds:

```
show gslb agent stats <seconds>
```

To reset the agent statistics to zero:

```
clear gslb agent stats
```

### Remote Node Statistics

To view all communication statistics with a remote node from a GSLB master:

```
show gslb remotenode <name> stats
```

To view a subset of a remote node's statistics that is updated every N seconds:

```
show gslb remotenode <name> stats <seconds>
```

To reset a remote node's statistics to zero:

```
clear gslb remotenode <name> stats
```



**Resolver Statistics**

To view all resolver statistics:

```
show gslb resolver <name> stats
```

To view a subset of the resolver statistics that is updated every N seconds:

```
show gslb resolver <name> stats <seconds>
```

To reset the resolver statistics to zero:

```
clear gslb resolver <name> stats
```

**GSLB Group Statistics**

To view group statistics:

```
show gslb resolver <name> group <name> stats
```

The output shows the number of times each member IP address has been served to an LDNS.

**Deployment Scenarios**

---

In the following scenarios, the domain “widgets.com” has three data centers. Data Center A, B, and C with address spaces 1.1.1/24, 2.2.2/24, and 3.3.3/24, respectively. The GSLB master is in Data Center A. The top-level “.com” domain registrar specifies 1.1.1.1 as the DNS server for “widgets.com”.

**Basic DNS, Resolver, and Group Configuration**

The following basic configuration on the GSLB master is used for each of the load-balancing examples that follow.

1. Configure the internal DNS server for “widgets.com”.

```
dx% add gslb localdns domain widgets.com
dx% set gslb localdns domain widgets.com ns widgets.com ns.widgets.com
dx% set gslb localdns domain widgets.com mx mx.widgets.com
dx% set gslb localdns domain widgets.com ttl 3600
dx% set gslb localdns domain widgets.com contact hostmaster.widgets.com
dx% set gslb localdns domain widgets.com sequence number 1
dx% set gslb localdns domain widgets.com sequence autoincrement enabled
```

2. Configure a GSLB resolver (the GSLB master listens on virtual IP 1.1.1.1):

```
dx% add gslb resolver 1
dx% set gslb resolver 1 listen vip 1.1.1.1
dx% set gslb resolver 1 target ip localdns
```

3. Create a GSLB group within the GSLB resolver.

```
dx% add gslb resolver 1 group 1
dx% set gslb resolver 1 group 1 dns hostname www.widgets.com
dx% set gslb resolver 1 group 1 dns ttl 3600
dx% set gslb resolver 1 group 1 dns authdomainname widgets.com
dx% set gslb resolver 1 group 1 dns authservername ns.widgets.com
```

### **Simple Round Robin**

In the following commands, entered on the GSLB master, DNS requests for “www.widgets.com” receive the DX addresses 1.1.1.100, 2.2.2.100, and 3.3.3.100 in sequence. Since metric-based load balancing is not used, each address is associated with the GSLB master (the “local” remote node). The master sends one ping per second to the remote addresses to verify availability. Addresses that fail to respond to three pings are removed from the rotation until they respond to three consecutive pings.

```
set gslb resolver 1 group 1 remotenode local 1.1.1.100 1
set gslb resolver 1 group 1 remotenode local 2.2.2.100 1
set gslb resolver 1 group 1 remotenode local 3.3.3.100 1
set gslb resolver 1 group 1 lba policy roundrobin
```

### **Weighted Round Robin**

The configuration is the same as simple round robin, except that each DX address is assigned a weight, and the group policy is set to “weightedroundrobin”. In this case, the DX address 2.2.2.100 is returned for two consecutive requests.

```
set gslb resolver 1 group 1 remotenode local 1.1.1.100 1
set gslb resolver 1 group 1 remotenode local 2.2.2.100 2
set gslb resolver 1 group 1 remotenode local 3.3.3.100 1
set gslb resolver 1 group 1 policy weightedroundrobin
```

### **Metric-based Load Balancing**

To use metric-based load balancing, the GSLB agent must be configured on each GSLB node, and each node must be defined on the GSLB master. In the following example, the default metric settings are used, and the DX nodes listen for statistics requests on addresses 1.1.1.2, 2.2.2.2, and 3.3.3.2.

1. Configure the GSLB agent on the DX in each data center.
  - a. Data Center A (the GSLB master):

```
dx-1% set gslb agent listen vip 1.1.1.2
dx-1% set gslb agent listen port 9843
dx-1% set gslb agent enabled
```

## b. Data Center B:

```
dx-2% set gslb agent listen vip 2.2.2.2
dx-2% set gslb agent listen port 9843
dx-2% set gslb agent enabled
```

## c. Data Center C:

```
dx-3% set gslb agent listen vip 3.3.3.2
dx-3% set gslb agent listen port 9843
dx-3% set gslb agent enabled
```

## 2. Define remote nodes for each DX appliance:

## a. Data Center A:

```
dx-1% add gslb remotenode sitea
dx-1% set gslb remotenode sitea agentip 1.1.1.2
dx-1% set gslb remotenode sitea port 9843
dx-1% set gslb remotenode sitea timeout 3
```

## b. Data Center B:

```
dx-2% add gslb remotenode siteb
dx-2% set gslb remotenode siteb agentip 2.2.2.2
dx-2% set gslb remotenode siteb port 9843
dx-2% set gslb remotenode siteb timeout 3
```

## c. Data Center C:

```
dx-3% add gslb remotenode sitec
dx-3% set gslb remotenode sitec agentip 3.3.3.2
dx-3% set gslb remotenode sitec port 9843
dx-3% set gslb remotenode sitec timeout 3
```

## 3. Add a resolver to each DX appliance:

## a. Data Center A:

```
dx-1% add gslb resolver 1 group 1 member 1
dx-1% set gslb resolver 1 group 1 member ip 1.1.1.2
dx-1% set gslb resolver 1 group 1 member remotenode sitea
dx-1% set gslb resolver 1 group 1 member weight 1
dx-1% set gslb resolver 1 group 1 lba policy metric
```

## b. Data Center B:

```
dx-2% add gslb resolver 1 group 1 member 1
dx-2% set gslb resolver 1 group 1 member ip 2.2.2.2
dx-2% set gslb resolver 1 group 1 member remotenode siteb
dx-2% set gslb resolver 1 group 1 member weight 1
dx-2% set gslb resolver 1 group 1 lba policy metric
```

## c. Data Center C:

```

dx-3% add gslb resolver 1 group 1 member 1
dx-3% set gslb resolver 1 group 1 member ip 3.3.3.2
dx-3% set gslb resolver 1 group 1 member remotenode sitec
dx-3% set gslb resolver 1 group 1 member weight 1
dx-3% set gslb resolver 1 group 1 lba policy metric

```

**Adjusted Metric Load Balancing**

In the following example, the metric weights are adjusted so that target host availability is primary, client connections are secondary, and CPU and memory usage have minor importance. Bit rates and RTT times are not considered. A high degree of smoothing is also enabled to minimize the effects for load spikes.

```

dx-1% set gslb resolver 1 group 1 metric targethostavailability min 3
dx-1% set gslb resolver 1 group 1 metric targethostavailability weight 100
dx-1% set gslb resolver 1 group 1 metric connections max 200
dx-1% set gslb resolver 1 group 1 metric connections weight 66
dx-1% set gslb resolver 1 group 1 metric memusage max 80
dx-1% set gslb resolver 1 group 1 metric memusage weight 33
dx-1% set gslb resolver 1 group 1 metric byterate weight 0
dx-1% set gslb resolver 1 group 1 metric rtt weight 0
dx-1% set gslb resolver 1 group 1 metric smoothing high

```

**RTT-only Load Balancing**

The following example uses only the round-trip time statistics for load balancing. The RTT settings are configured, and the weights of all other metrics are set to 0.

```

dx-1% set gslb resolver 1 group 1 metric rtt max 3000
dx-1% set gslb resolver 1 group 1 metric rtt timeout 3000
dx-1% set gslb resolver 1 group 1 metric rtt count 2
dx-1% set gslb resolver 1 group 1 metric rtt weight 1
dx-1% set gslb resolver 1 group 1 metric connections weight 0
dx-1% set gslb resolver 1 group 1 metric byterate weight 0
dx-1% set gslb resolver 1 group 1 metric cpuusage weight 0
dx-1% set gslb resolver 1 group 1 metric targethostavailability weight 0
dx-1% set gslb resolver 1 group 1 metric memusage max 0
dx-1% set gslb resolver 1 group 1 metric memusage weight 0
dx-1% set gslb resolver 1 group 1 metric sessions max 0
dx-1% set gslb resolver 1 group 1 metric sessions weight 0

```

## GSLB Failover

---

GSLB failover is managed by the failover feature. Metric-based load balancing is not compatible with older failover solutions, nor with ActiveN.

If a GSLB node fails over, metric calculations may be unstable for a short time (refer to Configuring Parameters for Metric-based Load Balancing on page 342). Any RTT calculations in progress will be lost, which results in a temporary availability score of 0 for RTT. If the GSLB master fails over, all state information is lost. Any pending DNS requests will not be answered, and the new master will have none of the cached statistics from the remote GSLB nodes.

For instructions on how to configure GSLB failover, see Chapter 20, “Configuring Failover” on page 357.



## Chapter 20

# Configuring Failover

This chapter describes the new failover method introduced in the DX Application Acceleration Platform Version 5.1. This method lets you specify a single failover configuration that applies to SLB, Forwarders, Clusters, ActiveN, and GSLB services. If you are upgrading from a prior version, we strongly recommend that you migrate to this new failover method.

This chapter includes the following topics:

- “Configuring Failover on Your DX Appliance” on page 357
- “Viewing Failover Configuration and Statistics” on page 359
- “Migrating to the New Failover Method” on page 365
- “Initiating a Manual Server Failover” on page 367
- “Gateway Failure Detection” on page 368

For a description of failover and how it works, see “Failover” on page 48.

## Configuring Failover on Your DX Appliance

---

After you configure your DX appliances with the services that are eligible for failover (SLB, Forwarder, Redirectors, Cluster, ActiveN, and/or GSLB), you can configure failover on two or more devices using the WebUI or the CLI. All CLI commands can be executed by users with `administrator` or `network_administrator` roles.

Use the `set failover` command to configure failover on a DX appliance using the CLI. Failover is disabled by default.

```
dx% set failover <enabled|disabled>
```

### Customizing the Failover Process

Optionally, you can also configure the following failover parameters:

- Enable or disable a DX appliance as the master (disabled by default):

```
dx% set failover forcemaster <enabled|disabled>
```

You can force a standby node to become the master at any time. If `forcemaster` is set on multiple peers, and the master becomes unavailable, the peer with the lowest node ID becomes the master.

- Specify a node ID for the DX appliance if you want to force the order in which the DX appliances take over during a failover:

```
dx% set failover nodeid <id|auto>
```

Enter `auto` to generate an ID from the IP address (default is `auto`). Unless `forcemaster` is set on one of the peers, the peer with the lowest node ID becomes the master when the master becomes unavailable.

- Specify an Ethernet interface for failover communication (ether0 is the default interface):

- Specify the interface used to discover the other peers (default is ether0).

```
dx% set failover discovery interface ether <n>
```

- Specify the port number used to discover peers enabled for failover (default is 9400). The ADFP Discovery packets are sent to this port. The port number should be the same for all nodes in a failover cluster.

```
dx% set failover discovery port <port>
```

- Specify the port number used to listen for ADFP Active and Standby packets (default is 9500).

```
dx% set failover listen port <port>
```

- Specify a virtual MAC address (VMAC) to be used on an Ethernet interface when a standby DX becomes the master:

- Enable the use of one or all VMAC address on an interface (default is enabled):

```
dx% set failover vmac ether <n|all> <enabled|disabled>
```

- Specify a specific VMAC ID (1 to 254) that determines the VMAC address (default is N + 1 for etherN).

```
dx% set failover vmac ether <n> id <id>
```

- Enable or disable failover for one or all links when a link fails (enabled for all links by default):

```
dx% set failover linkfail ether <n>|all> <enabled/disabled>
```

- Specify the polling interval (in seconds) to verify the availability of the other peers (default is one second). The polling intervals should be the same for all nodes in a failover cluster.

```
dx% set failover advanced pollinterval <seconds>
```



- Specify the number of missed packets that indicate a peer is unavailable (default is three):

```
dx% set failover advanced missedcount <n>
```

- To specify how often to monitor a failed service (default is eight):

```
dx% set failover advanced serviceinterval <seconds>
```

## Viewing Failover Configuration and Statistics

---

The following sections describe additional CLI commands used to monitor the failover feature and the services configured for failover.

- “Viewing the Current Failover Configuration” on page 359
- “Viewing Statistical Information” on page 360
- “Viewing Status Information about Services” on page 363

### Viewing the Current Failover Configuration

The following new `show` commands can be executed by users with `Administrator`, `Network Administrator`, or `Network Operator` roles.

To view the failover configuration:

```
show failover
```

To view only the failover status of the current device:

```
show failover status
```

To view specific failover configuration settings:

```
show failover forcemaster
show failover nodeid
show failover discovery
show failover discovery interface
show failover discovery port
show failover listen port
show failover listen ip
show failover linkfail
```

To view the VMAC settings and status for one or all interfaces:

```
show failover vmac
show failover vmac ether <n>
show failover vmac ether <n> id
show failover vmac ether <n> status
```

To view the failover settings for one or all peers enabled for failover:

```
show failover peer
show failover peer <ip>
show failover peer listen port
```

To view the advanced failover settings:

```
show failover advanced pollinterval
show failover advanced missedcount
show failover advanced serviceinterval
```

### Viewing Statistical Information

You can monitor failover activity using the `show failover stats` and `show failover stats advanced` commands. These commands display:

- the time the DX node has spent in each mode (Discover/Master/Standby)
- the number of transitions made between each mode
- the runtime status of all monitored links (up/down)
- the list of all services being monitored
- statistics about each peer

The following example shows the statistics displayed when the `show failover stats` command is entered:

```
dx% show failover stats
Unified Failover statistics (uptime = 102788 secs).
In Master mode since 102785 secs.
Discover: count=1 time=2 secs
Master: count=1 time=102785 secs
Standby: count=0 time=0 secs
Idle: count=0 time=0 secs
Link Status: eth0 = (1,1) eth1 = (1,1) eth2 = (1,0) eth3 = (1,0)
Link failure: Count = 0 Failover = 0
Sanity check failure: Count = 0 Failover = 0
Service failure: Count = 0 Failover = 0
Global Flags = 0xc3
Supported services: SLB GSLBResolver
10.80.48.40 Unreachable Count = 0 Flags = 0x11000c1
  Rcvd [DP,AP,SP] before [102787,102787,0] secs
  Pkts Sent (DP,AP,SP) = (0,98820,0)
  Pkts Rcvd (DP,AP,SP) = (0,1,98794)
  Pkt Send errs (AP,SP) = (0,0)
```

The output begins with the amount of time since the failover feature was started (uptime). It also indicates the current mode the DX appliance is in (discover, master, or standby) and how long it has been in this current mode. The output then lists the values for the following statistics:

- Discover/Master/Standby/Idle—The **count** indicates the number of times (since the start) this DX appliance has been in the discover, master, standby, or idle mode, and **time** indicates the total amount of time (in seconds) it has been so.
- Link Status—Indicates the link status of all interfaces (ether0 through ether3). For each interface, the first value tells whether the link-check is enabled (1) or disabled (0), and the second value shows the result of link-check (link is up = 1, link is down = 0).
- Link failure—The **Count** indicates the number of times a link went down, and **Failover** indicates the number of times the DX appliance failed-over when the link went down. Note that in a standalone configuration, the DX appliance does not failover on link failure, causing the **Count** and **Failover** values to be different.
- Sanity check failure—The **Count** indicates the number of times more than one master has been found on the network during any given polling interval. The **Failover** indicates the number of times that this DX appliance has failed over when the sanity check has failed.
- Service failure—Used in master mode only, the **Count** indicates the number of times a supported service has gone down. If a service does not come back up after waiting the amount of time specified by the **ServiceInterval**, **Failover** is incremented.
- Global Flags—Displays the internal value for failover.
- Supported services—Displays the list of supported services.
- <Peer IP Address> —Displays several statistics for each peer:
  - The **Unreachable Count** indicates the number of times this peer was unreachable.
  - **Flags** is an internal value used by failover.
  - **Rcvd [DP,AP,SP] before** indicates the amount of time that passed before this peer received a DiscoveryPacket (DP), an ActivePacket (AP), and a StandbyPacket (SP).
  - **Pkts Sent (DP,AP,SP)** indicates the number of packets sent (of each type) to this peer.
  - **Pkts Rcvd (DP,AP,SP)** indicates the number of packets received (of each type) from this peer.
  - **Pkt Send errs (AP,SP)** indicates the number of times active and standby packets could not be sent.

The following example shows the statistics displayed when the `show failover stats advanced` command is entered:

```
dx% show failover stats advanced
Loopback pkts rcvd = 3
Invalid (type) pkts rcvd = 0
Invalid (len) pkts rcvd = 0
Invalid (port) pkts rcvd = 0
Invalid (discovery port mismatch) pkts rcvd = 0
Invalid (discovery iface mismatch) pkts rcvd = 0
Master because no peers were found = 1
Gratuitous ARPs sent = 1
Dynamic config changes done = 2
Discovery listener: StartCnt = 1, FailedCnt= 0
Packet listener: StartCnt = 1, FailedCnt= 0
Force box to be Master:
  Cnt = 0, AckCnt = 0, NoAckCnt = 0, StandbyBeforeActionCnt = 0
```

The output displays values for the following statistics:

- Loopback pkts rcvd—Number of loopback packets received.
- Invalid (type) pkts rcvd—Number of packets received with a type mismatch.
- Invalid (len) pkts rcvd—Number of packets received with a length mismatch.
- Invalid (port) pkts rcvd—Number of packets received with a Listen port mismatch.
- Invalid (discovery port mismatch) pkts rcvd—Number of DiscoveryPackets received with a Discovery port mismatch.
- Invalid (discovery iface mismatch) pkts rcvd—Number of DiscoveryPackets received on a different interface.
- Master because no peers were found—Number of times this DX appliance became the Master because it was in a standalone configuration.
- Gratuitous ARPs sent—Number of times this DX appliance sent gratuitous address recognition protocol packets.
- Dynamic config changes done—Number of times a dynamic configuration change was made on this DX appliance.
- Discovery listener—The **StartCnt** indicates the number of times this DX appliance started the listener for DiscoveryPackets, and the **FailedCnt** indicates the number of times this failed.
- Packet listener—The **StartCnt** indicates the number of times this DX appliance started the packet listener, and the **FailedCnt** indicates the number of times this failed.

- Force box to be Master—Displays several statistics when `ForceMaster` is enabled, causing a Standby box to become the Master:
  - `Cnt` indicates the number of times this DX appliance was forced into being the Master.
  - `AckCnt` indicates the number of times when an acknowledgement was received from the previous Master (specifically a `StandbyPacket`).
  - `NoAckCnt` indicates the number of times and acknowledgement was not received.
  - `StandbyBeforeActionCnt` indicates the number of times this DX appliance became the Standby while it was being forced to act as the Master.

### Resetting Failover Statistics

Use the `clear failover stats` command to clear the failover statistics. This includes the following:

- All statistics displayed by the `show failover stats` command
- All statistics displayed by the `show failover stats advanced` command
- All failover statistics/logging information displayed by the `show system debug` command

The start time for failover is reset and continues in its current state (runtime behavior is not impacted).

### Viewing Status Information about Services

You can view status information about each of the configured services using various show commands. For example:

```
show server status
show activeN status
show slb status
show gslb agent
```

The status of the selected service is indicated with the terms shown in Table 6.

**Table 6: Status Displayed for Various Services**

Failover Status	Configuration Status	Process Status	Display Text
Disabled	Disabled	Disabled	down
Disabled	Disabled	Enabled	up (loaded config: down)
Disabled	Enabled	Disabled	down (loaded config: up)
Disabled	Enabled	Enabled	up
Enabled—Master	Disabled	Disabled	down
Enabled—Master	Disabled	Enabled	up (loaded config: down)
Enabled—Master	Enabled	Disabled	down (loaded config: up, Failover: Master)

**Table 6: Status Displayed for Various Services (continued)**

Failover Status	Configuration Status	Process Status	Display Text
Enabled—Master	Enabled	Enabled	up (Failover: Master)
Enabled—Standby	Disabled	Disabled	down
Enabled—Standby	Disabled	Enabled	up (loaded config: down)
Enabled—Standby	Enabled	Disabled	up (Failover: Standby)
Enabled—Standby	Enabled	Enabled	up (loaded config: up, Failover: Standby)

While the configuration status and process (service) status have been available, the failover status is new to the version 5.1 release. The failover status is displayed in one of four formats:

- As a single value—*value*. This format indicates whether both the service and configuration are enabled (up) or disabled (down).
- With two values—*value1 (loaded config: value2)*. This format indicates one value for the status of the service (up or down) and a second value for the status of the configuration (up or down).
- With two values—*value1 (failover: value2)*. This format indicates one value for the status of the configuration (up or down) and a second value for the failover mode (Master or Standby).
- With three values—*value1 (loaded config: value2, failover: value3)*. This format indicates one value for the status of the configuration (up or down), a second value for the status of the configuration (up or down), and a third value for the failover mode (Master or Standby).

For example:

```
dx% show slb status
SLB: up (Failover: Standby)
```

In this example, the SLB service is configured and enabled on the DX appliance, but it is not running as this device is in Standby Failover mode.

If both the Server and ActiveN services are supported, both the master and the standby DX appliances run the server service. This is reflected in the status:

```
dx% show server status
Server: up (loaded config: up) (failover: Standby)
```

When the ActiveN service is supported, the master and all standby DX appliances run all of the configured services (server, Cluster, Forwarder, and/or Redirector).

## Migrating to the New Failover Method

---

We recommend that you migrate your current failover configuration to the newly introduced failover configuration after completing the installation of the DX Application Acceleration Platform Version 5.1 software. The old methods will not be supported in future releases. We also recommend that migration is performed when there is *no* or *very little* traffic.

The configuration migration instructions contained here assume that you have knowledge of DX platform features such as Client IP Sticky and the existing failover mechanism.

This section contains the following topics:

- “Migrating Server Failover Configurations” on page 365
- “Migrating SLB Failover Configurations” on page 365
- “Migrating ActiveN and ActiveOne Failover Configurations” on page 366

### Migrating Server Failover Configurations

To migrate from an existing active/standby server configuration:

1. Set the standby DX appliance administratively down:
 

```
dx-standby% set server down
```
2. Disable server failover on the standby DX appliance.
 

```
dx-standby% set server failover disabled
```
3. Disable server failover on the active DX appliance.
 

```
dx-active% set server failover disabled
```
4. Enable failover on the active DX appliance.
 

```
dx-active% set failover enabled
```
5. Enable failover on the standby DX appliance.
 

```
dx-standby% set failover enabled
```
6. Set the standby DX appliance administratively up.
 

```
dx-standby% set server up
```

### Migrating SLB Failover Configurations

To migrate from an existing SLB failover configuration:

1. Disable SLB on the standby DX appliance.
 

```
dx-standby% set slb disabled
```

2. Disable SLB failover on the standby DX appliance.

```
dx-standby% set slb failover disabled
```

3. Disable SLB failover on the master DX appliance.

```
dx-master% set slb failover disabled
```

4. Enable failover on the master DX appliance.

```
dx-master% set failover enabled
```

5. Enable failover on standby DX appliance.

```
dx-standby% set failover enabled
```

6. Enable SLB on the standby DX appliance.

```
dx-standby% set slb enabled
```

### ***Migrating ActiveN and ActiveOne Failover Configurations***

To migrate from an existing ActiveN or ActiveOne configuration:

1. Set the standby DX appliance administratively down.

```
dx-standbyn% set server down
```

2. Disable the ActiveN service on each of the standby DX appliances.

```
dx-standbyn% set activen disabled  
dx-standbyn% set activen failover disabled
```

3. Disable ActiveN failover on the master DX appliances.

```
dx-master% set activen failover disabled
```

4. Enable failover on the master DX appliance.

```
dx-master% set failover enabled
```

5. Enable failover on each of the standby DX appliances.

```
dx-standbyn% set failover enabled
```

6. Enable the ActiveN service on each of the standby DX appliances.

```
dx-standbyn% set activen enabled
```

7. Set each of the standby DX appliances administratively up.

```
dx-standbyn% set server up
```



## Initiating a Manual Server Failover

---

There are times when you need to take a server off-line for maintenance or debugging purposes. You can initiate a manual failover in an active-standby configuration as follows:

1. Verify that the current master (or active server) does not have the forcemaster feature enabled:

```
dx-active% show failover forcemaster
ForceMaster: disabled
```

2. On the backup server (or standby server), enable the forcemaster feature:

```
dx-standby% set failover forcemaster enabled
```

When the backup server fails to detect the heartbeat messages coming from the active server, it takes over processing and becomes the active node.

Either the server that you took off-line or a replacement unit can be returned to activity as a backup unit by typing the command:

```
dx% set server up
```

For example, if you modify a configuration on a cluster on an active DX that requires a restart of the multiplexing engine, the process brings the active DX down, and the standby DX takes over and becomes the active DX. This may result in a Web site not processing requests while the standby DX appliance takes over.

If you want to make configuration changes to an active-standby configuration without affecting request processing, use the following sequence:

1. Ensure that DX 1 and DX 2 are in an active-standby configuration (DX 1 is active and DX 2 is the standby).
2. Change the cluster configuration on DX 2 (passive).
3. Move the traffic to DX 2 (set the server down on DX 1).
4. Check the failover status on DX 2 (now active).
5. Check the ActiveN status on DX 2 (now active).
6. Bring DX 1 up (set the server up on DX 1).
7. Check the failover status on DX 1 (now passive).
8. Check the ActiveN status on DX 2 (now standby).
9. Change the cluster configuration on DX 1.

## Gateway Failure Detection

---

Another method of achieving reliability is through the use of gateway failure detection. Gateway failure detection allows you to initiate failover in the active-standby topology by doing health checks on pre-configured hosts (ActiveN is supported).



**NOTE:** Gateway failure detection and unified failover are mutually exclusive features. You must disable failover (using the `set failover disabled` command) prior to configuring this feature. Failover configuration by service is compatible with gateway failure detection.

To configure gateway failure detection, configure a DX in an active-standby topology. Then configure a group of IP addresses to health check; these IP addresses will be checked for layer 3 connectivity. If a health check fails, the standby DX appliance will take over as the primary.



**WARNING:** Enabling this feature and having a failover event causes the DX appliance to reboot.

Be certain that the hosts you choose to health check are pingable when you add them into your health checking. If they are not, then the following occurs:

- Assume that DX A is the active unit and DX B is the standby unit. If you add the IP of a host that is down to remote host health checking, DX A will not be able to ping that host, and will eventually failover to DX B. This causes a reboot of DX A.
- Depending upon the configuration of DX B, it is possible that DX A will not be done rebooting by the time DX B reboots. This can cause significant problems.

The “Server Load Balancer” (SLB) can also be included in the failover (you must enable SLB failover). The configured IP addresses are checked for Layer 3 connectivity via a ping; no Layer 4 check is performed. If a configured number of health checks fail, the active DX is switched to Standby mode. This allows the other DX to become active.

To enable gateway failure detection, you must enable failover with `set server failover enabled`. The following items must be configured:

- The health check interval (default value is 10 seconds; range: 10-600 seconds).
- The timeout value waiting for a response (default value: 10 seconds; range: 1-60 seconds).
- The maximum health check attempts per host (default value is 5; range: 1-60).
- The minimum remote hosts failing before activating failover (default value is 1; range: 1-10).
- The hosts to health check (maximum: 10).
- Gateway failure detection enabled/disabled (default value is disabled).

The DX will add an entry to the system logs when failover occurs. If the “timeout value” is larger than the “health check interval,” and a remote host is not responding, a health check request will not be sent until the “timeout value” has expired. In other words, if a host is not responding, the health check interval becomes the maximum value of the interval and timeout.

If “minimum remote hosts failing” is larger than number of remote hosts, failover will never occur. The DXSHELL prints a warning when this condition is set.

The failover algorithm is:

When the number of consecutive health check failures equals the “maximum health check attempts,” the host is considered down.

After this, if the number of hosts down is equal or greater than the “minimum remote hosts failing,” failover will occur.

When failover is invoked, the active DX will be rebooted to allow the standby unit to takeover. This feature will not startup unless one of the following is true:

- Active-standby failover is enabled
- ActiveN failover is enabled
- SLB failover is enabled

One of these must be enabled AND the DX must be the active unit for gateway failure detection to work.

### Gateway Failure Detection Commands

Use these commands to configure gateway failure detection.

To add an IP address to health check, type the command:

```
dx% set health remotehost host [ip]
```

To enable gateway failure detection, type the command:

```
dx% set health remotehost [enabled | disabled]
```

To set the health check interval (how often to send the health checks), type the command:

```
dx% set health remotehost interval [seconds]
```

To set the health check timeout (how long to wait for a response), type the command:

```
dx% set health remotehost timeout [seconds]
```

To set the health check maximum number of attempts before considering the host down, type the command:

```
dx% set health remotehost retry [count]
```

To set the count for the minimum number of hosts failing, type the command:

```
dx% set health remotehost minhosts failing [count]
```

To remove an IP address from health check, type the command

```
dx% clear health remotehost host [ip]
```

## Show Commands

Use these show commands to see the status of gateway failure detection:

```
dx% show health remotehost host
dx% show health remotehost status
dx% show health remotehost interval
dx% show health remotehost timeout
dx% show health remotehost retry
dx% show health remotehost minhosts failing
```

These commands may be executed by the Administrator, Network Administrator, and Network Operator.



## Chapter 21

# Tuning the DX Appliance for Enterprise Applications

This chapter describes tuning the DX Application Acceleration Platform for Enterprise applications, discussing the following topics:

- Target Tuning Tool on page 373
- WebDAV on page 374

## Target Tuning Tool

---

The purpose of Target Tuning is to enable you to easily set up the interaction with target hosts and to properly set up the cluster/system behavior for a custom environment. Target Tuning is a single DXSHELL command that sets a number of configuration variables in an interactive format. The command is:

```
dx% set cluster n target tuning
```

An example of a typical tuning session is shown as follows. The default answer for each of the questions is marked with an asterisk (\*):

```
dx% set cluster N target tuning
```

```
This will help optimize the communication with the Target Hosts within  
this cluster. It will help ensure that functionality is maintained while  
providing the most possible benefit.
```

```
Please answer the following questions. Enter Control-C at any time to  
exit without modification.
```

- ```
1) Please select the Target Application  
  1) Other (*)  
  2) OWA (Outlook Web Access)  
  3) PeopleSoft  
  4) Domino 5  
  5) Domino 6  
  6) JDE OneWorld
```

```
Enter Selection: ____
```

- 2) Please select the Target Web Server Type
- 1) Other (\*)
  - 2) Apache
  - 3) IIS4

Enter Selection: \_\_\_\_

- 3) Is NTLM Authentication used ?
- N) No (\*)
  - Y) Yes

Enter Selection: \_\_\_\_

You have selected:

- Target Application: OWA
- Target Web Server: Other
- NTLM Authentication: Yes

- Continue on using these selections ?
- N) No (\*)
  - Y) Yes

Enter Selection: \_\_\_\_

Tuning based on your selections ...  
Done.

Question #3 (Is NTLM Authentication used?) will only be presented if the choice for the Target Application does not require connection binding. If connection binding is required by the Application, you cannot unknowingly disable it in question #3.

## WebDAV

---

Web-based Distributed Authoring and Versioning (WebDAV) is a set of extensions to the HTTP protocol that allows users to collaboratively manage and edit files on remote Web servers. The primary force driving development of WebDAV is a new generation of programs that “Webify” existing Enterprise applications such as Microsoft’s Outlook Web Access (OWA) program.

WebDAV is an in-progress effort of the Internet Engineering Task Force (IETF). The activities in this effort are centralized at the <http://webdav.org> Web site. References to all relevant documents, as well as links to applications that use WebDAV can be found here. The WebDAV RFCs and drafts specify a new set of HTTP request methods, response codes, and headers that add to the functionality of HTTP. Additionally, Microsoft has defined an additional set of request methods, response codes, and headers.

The WebDAV methods are only available when you have acquired the appropriate license. Additionally, you have the ability to enable and disable support for these new methods on a per-cluster basis.



## Methods

HTTP request methods are divided in two categories: “basic” methods and “enhanced” methods. This was done in order to deliver additional security, as well as to provide fine grained control.

The basic HTTP methods are those that are needed to run any Web site, intranet, or Enterprise application. The enhanced methods are the rest of the (non-WebDAV) HTTP request methods that are needed in much fewer instances. The basic HTTP methods are always enabled. The enhanced HTTP methods are disabled by default (for security), but can be enabled by any user.

The basic HTTP methods consist of:

- GET
- HEAD
- POST
- PUT

The extended HTTP methods consist of:

- DELETE
- TRACE
- OPTIONS
- CONNECT

## Compression of 401 Responses

OWA requires users to login using the standard HTTP WWW-Authenticate mechanism. When the WWW-Authentication header is not present, OWA returns a 401 response code with a relatively large HTML body.

Compression of this HTML content increases the effective compression ratio for the site. In order to support compression of this content, a per-cluster factory option has been added to control compression for 401 responses. It is disabled by default, but it is enabled as part of the recommended WebDAV configuration.

## Compression of “text/x-component” MIME Type

OWA delivers content at the beginning of a session with the MIME type, “text/x-component.” This content compresses well, and a factory option has been added to enable the compression of this MIME type. It is disabled by default, but it is enabled as part of the recommended WebDAV configuration.

## Integration with Application Rules

The new HTTP request methods, headers, and response codes have been added as options to the AppRules. This allows full control of the HTTP traffic by the end user.

## Optimization

In order for you to get the maximum value out of the DX Application Acceleration Platform in an OWA deployment, you must enable OWA for the desired cluster. This enables the extended and WebDAV methods, enables connection binding, and enables compression of unauthorized responses and XML and X-component MIME types.

## New WebDAV and HTTP Extensions

Table 1 shows the new WebDAV and HTTP extensions that have been added.

**Table 1: New WebDAV and HTTP Extensions**

| New WebDAV and HTTP Extensions |            |             |                 |
|--------------------------------|------------|-------------|-----------------|
| ACL                            | CHECKIN    | MKRESOURCE  | SEARCH          |
| BASELINE-CONTROL               | CHECKOUT   | MKWORKSPACE | SUBSCRIBE       |
| BCOPY                          | COPY       | MOVE        | UNCHECKOUT      |
| BDELETE                        | LABEL      | NOTIFY      | UNLOCK          |
| BIND                           | LOCK       | POLL        | UNSUBSCRIBE     |
| BMOVE                          | MERGE      | PROPFIND    | UPDATE          |
| BPROPFIND                      | MKACTIVITY | PROPPATCH   | VERSION-CONTROL |
| BPROPMATCH                     | MKCOL      | REPORT      | X-MS-EMUMATTTS  |

Table 2 shows the new WebDAV Response Codes that have been added.

**Table 2: New WebDAV Response Codes**

| New Response Codes                                        |
|-----------------------------------------------------------|
| 102 Processing                                            |
| 207 Multi-Status                                          |
| 422 Unprocessable Entity                                  |
| 423 Locked                                                |
| 424 Failed Dependency                                     |
| 425 Insufficient Space on Resource                        |
| 506 Loop Detected                                         |
| 507 Insufficient Storage / Cross-Server Binding Forbidden |

Table 3 shows the new headers that have been added by the various groups.

**Table 3: New Headers**

| New Headers           |                    |                   |                       |
|-----------------------|--------------------|-------------------|-----------------------|
| Allow-Rename          | Depth              | Notification-Type | Subscription-ID       |
| Apply-To-Redirect-Ref | Destination        | Ordered           | Subscription-Lifetime |
| Brief                 | If                 | Overwrite         | Timeout               |
| Call-Back             | Label              | Position          | Transaction           |
| DASL                  | Lock-Token         | Redirect-Ref      |                       |
| DAV                   | Notification-Delay | Status-URI        |                       |

“Outlook Web Access” (OWA) uses the WEBDAV extensions to the HTTP protocol to provide increased functionality. The DX supports WebDAV extensions to HTTP protocol and allows users to accelerate OWA.

## OWA Commands

The following commands support the OWA feature:

```
dx% set cluster <name> owa [enabled|disabled*]
```

This command enables or disables the WebDAV feature:

```
dx% show cluster <name> owa
```

This command shows whether the OWA feature is enabled or disabled, and also shows if the “child” commands have been modified by the OWA settings.

For complete information on OWA commands, refer to the *Command Line Reference* manual.



## Part 4

# Monitoring and Troubleshooting Information and Procedures

This part of the *Installation and Administration Guide for DXOS* provides procedures for monitoring the performance of your DX appliance. It also describes some of the tools available to troubleshoot problems you may have when configuring or monitoring your DX appliance.

These topics can be found in the following chapters:

- Chapter 22, “Performance Monitoring” on page 381
- Chapter 23, “Troubleshooting” on page 413



## Chapter 22

# Performance Monitoring

This chapter describes performance monitoring for the DX Application Acceleration Platform discussing the following topics:

- View Juniper Server Statistics on page 382
- Capacity Planning on page 383
- Historical Rates and Statistics on page 383
- DXSHELL Output Example on page 390
- CSV Export Statistics on page 391
- Advanced Statistics on page 393
- SSL Listen Statistics on page 401
- Web Log Configuration on page 405

## View Juniper Server Statistics

To see real-time statistics for the DX, connect to the DXSHELL command line and type the commands shown in the left column of Table 4.

**Table 4: Commands for Viewing Statistics from the DXSHELL Command Line**

| Command                          | Definition                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
|----------------------------------|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| <code>show dashboard</code>      | <p>This command provides a convenient summary of the performance and health of the DX server and the health of clusters and target hosts.</p> <ul style="list-style-type: none"> <li>■ Overall health of the DX's memory, CPU, and network</li> <li>■ VIP (Cluster) and Target server health status</li> <li>■ DX performance including               <ul style="list-style-type: none"> <li>■ Bytes the DX sent to clients</li> <li>■ Connections accepted or refused</li> <li>■ Requests handled</li> <li>■ Bytes saved</li> </ul> </li> </ul> |
| <code>show server stats 1</code> | <p>This refreshes every second and shows:</p> <ul style="list-style-type: none"> <li>■ Active TCP sessions, total TCP sessions</li> <li>■ Active HTTP requests, total HTTP requests</li> <li>■ Bytes into the DX from the origin Web server(s)</li> <li>■ Bytes the DX sent to clients</li> </ul>                                                                                                                                                                                                                                                |
| <code>netstat</code>             | Shows the IP addresses and ports of all TCP connections to the DX.                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| <code>netstat 1</code>           | Shows the number of packets and bytes being received by the DX from the origin Web servers (input) and shows the number of packets and bytes being sent to clients (output). These numbers are updated every second.                                                                                                                                                                                                                                                                                                                             |

The DX also provides a limited selection of real-time performance statistics on the DX Stats page of the WebUI. To view the DX stats page, log in to the WebUI and click on the DX Stats link in the left-hand navigation area. You will see the DX Stats page which provides information about uptime, connections, requests, and bytes in/out.



## Capacity Planning

---

A DXSHELL user has information available to make capacity decisions. The information presented by netstat is very generic and includes information that has spikes. This may unfavorably report system as loaded based on information that is transient. Similarly, the WebUI interface shows Uptime, CPU, Memory, and Network. The WebUI information may change for a few seconds to "red" if the peak value is over a "pre-defined" threshold.

The capacity planning feature allows a user to receive a more informative capacity planning data using the `show capacity` command to show the capacity of the system:

```
dx% show capacity [<seconds>]
```

where `< seconds >` is time intervals for printing the next row. The minimum value for `< seconds >` is 1 and the maximum value is 60. If this argument is missing, only one row of output will be printed and the command will exit.

The displayed values are average over the last minute and the sample interval is one second. This smooths the peaks in the values.

Reporting network usage (including data for `show dashboard`) is limited to Ether 0 and Ether 1. Other interfaces are ignored.

## Historical Rates and Statistics

---

Historical Rates and Statistics allow you to use historical statistics to gauge behavior of your network and data center as well as the ongoing benefit of employing a DX. For example, you can answer questions such as "How many bytes have we saved in the past one week?, one hour?" and so forth.

Statistics for Input/Output (I/O), HTTP, and SSL are available from the DXSHELL. Statistics can be displayed at the cluster, forwarder and redirector, target host, and physical target levels. Some statistical views can also be narrowed to just the listen or target side.

It is also important to note the difference between a target host and a physical target. A physical target is a Web server and a target host is that physical target assigned to a cluster or forwarder. A physical target can be assigned to multiple clusters.

The following are the different categories of statistics available:

- I/O—Listen, Target Host, Physical Target
- HTTP—Listen, Target Host
- SSL—Listen, Target Host

## **The Round Robin Database Mechanism**

The Round Robin Database (RRDB) is a mechanism to store time-series data such as network usage, method invocations, etc. It stores the data in a very compact way that does not expand over time, and it can be presented in useful graphs by processing the data to enforce a certain data density.

To limit the data acquisition to a finite memory size, the RRDB's store snapshots of information at a pre-set sampling interval or period. These periods are then averaged and rolled into the next higher "bucket." The round robin, as the name implies, limits the history to the last "n" sampling snapshots in a "bucket" or "category."

For example, you have a sampling interval of one second and are interested in keeping information about 100 different items for a year. Since there are 60 seconds in minute, 60 minutes in an hour, 24 hours in a day, and so forth, and assuming there are eight bytes of information per statistical field, and 100 fields for historical information, the DX appliance would allocate 149,600 bytes of information.

This implies a database size of:

$$(60 \text{ sec entries} + 60 \text{ min entries} + 24 \text{ hour entries} + 30 \text{ day entries} + 12 \text{ month entries} + 1 \text{ year entry}) * 100 \text{ items being measured} * 8 \text{ bytes per item} = 149,600 \text{ bytes.}$$

This calculation does not include item header information or other housekeeping storage.

At every sampling interval (one second in the example), one entry is made for each statistical item (here 100). When 60 entries are made in the "seconds" bucket, one entry is made into the minute bucket with a value that is computed for the past 60 seconds for that statistic. Then the "second" bucket rolls back to overwriting the oldest value, the first second. The same mechanism is employed for other buckets, i.e., there will be one entry in the hour bucket for every sixty entries in the minute bucket.

## **Memory Considerations**

The Historical Rates and Statistics feature uses system memory to store the data collected on Custers and target hosts. To prevent these statistics from effecting overall performance of the DX appliance, the maximum number of Clusters allowed is 10 and the maximum number of target hosts is 54. These limits allow up to 4 MB of data to be stored in Flash memory and approximately 9 MB of data to be stored in RAM.

## **Description**

The Historical Rates and Statistics feature gives you the ability to collect data samples for each statistic item supported by the DX appliance. It adds the ability to specify the number of data samples collected for a statistic item. The sampling interval is fixed at one second and can not be set.

The sampling interval also determines the number of possible entries for the “seconds” table. The default size in the table for seconds data is 60 entries, minutes is 60 entries, hours is 24 entries, days is 31 entries, months is 12 entries, and years is 1 entries.

The sampling interval is not user configurable; the setting has been made at Juniper for optimum results. The trade-offs considered for optimum results are sustained throughput, and performance for connections per second, requests per second, new SSL connections per second, Mbit per second, and simultaneous client and target host connections.

The ability to view statistics is available to a “normal” user. Historical statistics are enabled by default, but can be disabled for a specific cluster.

Historical statistics are written at a one-hour interval to flash. No configuration is allowed.

You can specify a predefined filename to store the historical snapshot data. This is in flash memory. You can then send the data to a remote location using SCP or TFTP.

The sample data is stored as a Comma Separated Value (CSV) file. The format of CSV file is shown in Table 5.

**Table 5: Historical Statistics File Format**

| <b>ClusterName<br/>(IP:PORT)</b> | <b>Hour Bucket<br/>(1)</b> | <b>Hour Bucket<br/>(2) (24)</b> | <b>Hour<br/>Bucket</b> | <b>Day<br/>Bucket (1)<br/>(31)</b> | <b>Month<br/>Bucket (1) (12)</b> | <b>Month<br/>Bucket<br/>(1)</b> | <b>Year<br/>Bucket<br/>(1)</b> |
|----------------------------------|----------------------------|---------------------------------|------------------------|------------------------------------|----------------------------------|---------------------------------|--------------------------------|
| 10.11.12.13:80                   | Item1 Item2 .....          | Item1 Item2 ....                | Item1 ....             | Item1 Item2 ...                    |                                  |                                 |                                |
| 10.10.0.1:80                     |                            |                                 |                        |                                    |                                  |                                 |                                |
| .....                            |                            |                                 |                        |                                    |                                  |                                 |                                |
| .....                            |                            |                                 |                        |                                    |                                  |                                 |                                |
| .....                            |                            |                                 |                        |                                    |                                  |                                 |                                |
| .....                            |                            |                                 |                        |                                    |                                  |                                 |                                |

Minute and second buckets are not written into the flash. The target host’s minute and second historical statistics are stored in memory. Consolidated target host statistics are written to flash at one hour intervals.

### Statistical Data Items

Historical information is provided for all of the HTTP, IO, and SSL statistics for Clusters and target hosts. Historical information is also provided for all of the IO and SSL statistics for forwarders. You specify these `<stats items>` via Tab-completion. For clarity, the titles of the statistics need to be appended to their `<stats item>` name.

### Enabling Historical Rates and Statistics

The historical statistics feature is enabled for forwarders by default on DX appliances with the Base DX license. Historical statistics are enabled for forwarders, Clusters, and target host by default on DX appliances with the HTTP Acceleration license. (See “DX Product Licensing Options” on page 17 for details of the features available under these licenses.)

To verify that the license installed on your DX supports historical rates and statistics, use the `show license` command. For example:

```
dx% show license
DX 3200 1408
Virtual IP Addresses:    128
Target Hosts/VIP:       32
Connections:            50000
OWA (WebDAV) licensed.
ActiveN licensed.
  Groups:                128
  Blades:               2048
SLB licensed.
  Groups:                512
  Targethosts:          32
...
Historical Stats licensed.
Aprrules licensed:
  Request Translator Header (RTH): Unlimited
  Request Translator Content (RTC): Unlimited
  Page Translator Header (PTH):  Unlimited
  Page Translator Content (PTC):  Unlimited
  Request Sentry (RS):           Unlimited
...
```

Historical statistics are collected once each second until the maximum number of Clusters and target hosts is reached.

### Disabling Historical Statistics

You can't disable the collection of *all* historical statistics on a DX appliance, or you can disable historical statistics for a particular cluster.

To disable all historical statistics collection:

```
dx% set admin stats history down
```

You can verify the results of this using a `show cluster` command, for example:

```
dx% show cluster 1 stats history http listen method GET minute
```

---

```
Last 60 minutes GET
```

---

|              | Absolute Value | Delta Value |
|--------------|----------------|-------------|
| [Jun02]16:02 | 100            | 0           |
| [Jun02]16:01 | 100            | 0           |
| [Jun02]16:00 | 100            | 0           |

```
[Jun02]15:59    100          0 <— DISABLED
[Jun02]15:58    100          1
[Jun02]15:57    99           10
[Jun02]15:56    89           0
```

In this example, on June 2nd at 15.58 the historical statistics were disabled. This is visible because the number of statistics collected (Absolute Value) remains unchanged from that time and the change in the absolute value (Delta Value) is zero from that time.

To disable historical statistics collection for a selected cluster:

```
dx% set cluster <name> stats history disabled
```

### Showing the Cluster Historical Statistics Items

To show the cluster historical statistics items, type the following commands:

```
dx% show cluster 1 stats history [TAB]
http io ssl
```

```
dx% show cluster 1 stats history io [TAB]
listen target
```

```
dx% show cluster 1 stats history ssl [TAB]
listen target
```

```
dx% show cluster 1 stats history http [TAB]
listen target
```

```
dx% show cluster 1 stats history http listen [TAB]
browser method reqerr request version
```

```
dx% show cluster 1 stats history http listen browser [TAB]
lists of all the browser types.
```

```
dx% show cluster 1 stats history http listen browser
<browser-type>[TAB]
second minute hour day month year
```



**NOTE:** A similar format is followed for I/O and SSL listen historical statistics.

Cluster listen side historical statistics have all the time buckets (second, minute, hour, day, month, and year).

Cluster target side historical statistics have only hour, day, month, and year.

Each target host maintains second and minute historical statistics, and they are accumulated for a cluster at every hour interval.

Target hosts do not maintain hour, day, month, and year historical statistics.

```
dx% show cluster 1 stats history http target [TAB]
bytesin bytesout content responsecode
```

```
dx% show cluster 1 stats history http target bytesin [TAB]
list of all the bytesin stats.
```

```
dx% show cluster 1 stats history http target bytesin
<bytesin-item> [TAB]
```

```
hour day month year
```



**NOTE:** A similar format is followed for I/O and SSL target historical statistics.

Cluster target side historical statistics have only hour, day, month, and year.

Each target host maintains second and minute historical statistics, and they are accumulated for a cluster at every hour interval.

Target hosts do not maintain hour, day, month, and year historical statistics.

### Showing Target Host Historical Statistics Items

To show the target historical statistics items, type the following commands:

```
dx% show cluster 1 target host ip:port stats history [TAB]
http io ssl
```

```
dx% show cluster 1 target host ip:port stats history http [TAB]
bytesin bytesout content responsecode
```

```
dx% show cluster 1 target host ip:port stats history http
bytesin [TAB]
list of all the bytesin stats
```

```
dx% show cluster 1 target host ip:port stats history http target
bytesin <bytesin-item> [TAB]
second minute
```



**NOTE:** A similar format is followed for I/O and SSL target historical statistics.

Each target host maintains second and minute historical statistics, and they are accumulated for a cluster at every hour interval.

Target hosts do not maintain hour, day, month, and year historical statistics for space limitations and performance.

### Showing Server Historical Statistics Items

To show the server historical statistics items, type the following commands:

```
dx% show server stats history [TAB]
http io ssl
```

```
dx% show server stats history io [TAB]
listen target
```

```
dx% show server stats history ssl [TAB]
listen target
```

```
dx% show server stats history http [TAB]
listen target
```

```
dx% show server stats history http listen [TAB]
browser method reqerr request version
```

```
dx% show server stats history http listen browser [TAB]
lists of all the browser types.
```

```
dx% show server stats history http listen browser <browser-type>
[TAB]
second minute hour day month year
```



**NOTE:** A similar format is followed for I/O and SSL target historical statistics.

Server listen side historical statistics have all the time buckets (second, minute, hour, day, month, and year).

```
dx% show server stats history http target [TAB]
bytesin bytesout content responsecode
```

```
dx% show server stats history http target bytesin [TAB]
list of all the bytesin stats
```

```
dx% show server stats history http target bytesin <bytesin-item>
[TAB]
hour day month year
```



**NOTE:** A similar format is followed for I/O and SSL target historical statistics.

Server target side historical statistics have only hour, day, month and year.

Server does not have target side historical statistics for second and minute bucket.

### Clearing Historical Statistics for All Clusters and Target Hosts

To clear the historical statistics items or all clusters and target hosts, type the following command:

```
dx% clear server stats
```

This command clears the historical statistics for all the clusters and target hosts by resetting the counter values to zero.

### Clearing Historical Statistics For a Cluster

To clear the historical statistics items for a cluster, type the following command:

```
dx% clear cluster <name> stats
```

This command clears the historical statistics for the cluster, and all the target hosts under that cluster.

## DXSHELL Output Example

An example of the `show server stats` command output when viewed from the DXSHELL system console is:

```
dx% show server stats history http listen browser IE6.0 day
-----
Last 31 days                                IE6.0
-----
                Absolute Value      |      Delta Value
-----
    Mar   02                23815162                2926001
    Mar   01                20889161                10189812
    Feb   29                10699349                8386042
    Feb   28                 2313307                2313307
    Feb   27                   0                          0
    Feb   26                   0                          0
    Feb   25                   0                          0
    Feb   24                   0                          0
    ...
    ...
    Feb   04                   0                          0
    Feb   03                   0                          0
```

In this example, the server was started on February 27th. By the end of the day on February 27, the site had received 2313307 hits from users using the Internet Explorer version 6.0 browser. This number became the Absolute Value for February 28th.

By the end of the day on February 28, the site had received 8386042 hits from users using the Internet Explorer version 6.0 browser. This Delta Value (8386042) was added to the previous Absolute Value (2313307) to become the Absolute Value for February 29th (10699349).

All of the `show status` commands use a similar format when executed from the DXSHELL system console.



## CSV Export Statistics

The CSV Export feature allows the historical statistics to be saved as a “Comma Separated Value” (CSV) file that can be exported outside of the DX. You can have a separate file for each cluster or a single file for all the clusters. The files are created using either a command from the DXSHELL or from the WebUI. Historical Statistics must be licensed for this feature to be available.

The format of the CSV file with statistics for one cluster is shown in Table 6.

**Table 6: Format of the CSV File with Statistics for One Cluster**

| <b>Item</b> | <b>Hour<br/>1</b> | <b>Hour<br/>2</b> | <b>...</b> | <b>Hour<br/>24</b> | <b>Day<br/>1</b> | <b>Day<br/>2</b> | <b>...</b> | <b>Day<br/>31</b> | <b>Month<br/>1</b> | <b>Month<br/>2</b> | <b>...</b> | <b>Month<br/>12</b> | <b>Year</b> |
|-------------|-------------------|-------------------|------------|--------------------|------------------|------------------|------------|-------------------|--------------------|--------------------|------------|---------------------|-------------|
| Item 1      |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| Item 2      |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| ...         |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| Item N      |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |

The format of the CSV file with stats for all the clusters is shown in Table 7.

**Table 7: Format of the CSV File with Statistics for All of the Clusters**

| <b>Cluster</b> | <b>Item</b> | <b>Hour<br/>1</b> | <b>Hour<br/>2</b> | <b>...</b> | <b>Hour<br/>24</b> | <b>Day<br/>1</b> | <b>Day<br/>2</b> | <b>...</b> | <b>Day<br/>31</b> | <b>Month<br/>1</b> | <b>Month<br/>2</b> | <b>...</b> | <b>Month<br/>12</b> | <b>Year</b> |
|----------------|-------------|-------------------|-------------------|------------|--------------------|------------------|------------------|------------|-------------------|--------------------|--------------------|------------|---------------------|-------------|
| Cluster 1      | Item 1      |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| Cluster 1      | Item 2      |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| Cluster 1      | ...         |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| Cluster 1      | Item N      |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| Cluster 2      | Item 1      |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| Cluster 2      | Item 2      |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| Cluster 2      | ...         |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| Cluster 2      | Item N      |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| Cluster ...    | Item 1      |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| Cluster ...    | Item 2      |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| Cluster ...    | ...         |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| Cluster ...    | Item N      |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| Cluster M      | Item 1      |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| Cluster M      | Item 2      |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| Cluster M      | ...         |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |
| Cluster M      | Item N      |                   |                   |            |                    |                  |                  |            |                   |                    |                    |            |                     |             |

Each of the values is separated by a comma. The items are each of the statistics for which historical statistics are currently available (refer to “Historical Rates and Statistics” on page 383).

## Export CSV Statistics Commands

The CSV file can be exported to either a TFTP server or a SCP server. To export the CSV Statistics file, type the command:

```
dx% export cluster <id | all> stats history <dst>
```

This creates the historical statistics file for the given cluster or all clusters and exports it to the specified URL. The file is deleted after the export is complete. This command requires Admin, Network Admin, or Network User administration rights.

The format of the destination <dst> is:

```
tftp://tftp_server/filename or  
scp://scp_server/filename
```

Double quotes must be used if the filename has spaces:

```
"tftp://tftp_server/dx config"
```

The <scp\_server> name is a host name or an IP address. The <filename> is an absolute path of the file where you would like to export the configuration. The directory specified for the filename must exist.

## Exporting CSV Statistics from the WebUI

The CSV file can be exported from the WebUI in one of two ways:

- A link is provided in the Server (DX) statistics page that downloads the Historical Statistics for all the pages. The downloaded data is saved as a file on the client machine.
- A link is provided in the Cluster Stats -> per cluster page that downloads historical statistics for a single cluster. The downloaded data is saved as a file on the client machine.

## Advanced Statistics

---

### Overview

Statistics for Input/Output (I/O), HTTP, and SSL are available from the DXSHELL. Statistics can be displayed at the cluster, forwarder and redirector, target host, and physical target levels. Some statistical views can also be narrowed to just the listen or target side.

It is also important to note the difference between a target host and a physical target. A physical target is a Web server and a target host is that physical target assigned to a cluster or forwarder. A physical target can be assigned to multiple clusters.

The following are the different categories of statistics available:

- I/O Listen
- I/O Target Host
- I/O Physical Target
- HTTP Listen
- HTTP Target Host
- SSL Listen
- SSL Target Host

### I/O Listen Statistics

I/O Listen statistics can be shown at the cluster, forwarder and redirector, and server levels (refer to Table 8).

To display the I/O statistics for a specific cluster, type the command:

```
dx% show cluster <name> stats io
```

To display the I/O statistics for all clusters, type the command:

```
dx% show cluster all stats io
```

To display the I/O statistics for the DX server, type the command:

```
dx% show server stats io
```

**Table 8: I/O Listen Statistics**

| Field                               | Description                                                                                                                                                           |
|-------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Bytes In<br>(requests from clients) | Cluster or Redirector: Number of bytes at the HTTP level (header or data) received or transmitted by the DX on the listen side.                                       |
| Bytes Out<br>(response to clients)  | Forwarder: Number of data bytes in TCP packets received or transmitted by the DX on the listen side.                                                                  |
| Current Client Connections          | Current number of established TCP connections from clients.                                                                                                           |
| Total Client Connections            | Total number of TCP connections that have ever been established (SYN, SYN-ACK, ACK) from the clients.                                                                 |
| Refused Client Connections          | Total number of TCP connections that the DX has accepted from clients and then immediately closed due to resource constraints. A busy message may or may not be sent. |

### ***I/O Target Host Statistics***

I/O Target Host statistics can be shown at the cluster target host, forwarder target host, cluster, forwarder, and server levels (refer to Table 9).

To display the I/O statistics for a specific target host within a cluster, type the command:

```
dx% show cluster <name> target host <name> stats io
```

To display the I/O statistics for all target hosts within a cluster, type the command:

```
dx% show cluster <name> target host all stats io
```

**Table 9: I/O Target Host Statistics**

| Field                                | Description                                                                                                       |
|--------------------------------------|-------------------------------------------------------------------------------------------------------------------|
| Bytes In<br>(responses from servers) | Cluster: Number of bytes at the HTTP level (header or data) received or transmitted by the DX on the target side. |
| Bytes Out<br>(requests to servers)   | Forwarder: Number of data bytes in TCP packets received or transmitted by the DX on the target side.              |

## I/O Physical Target Statistics

I/O Physical Target statistics can be shown in detail at the cluster target host and forwarder target host levels, or summarized at the cluster, forwarder, and server levels (refer to Table 10).

**Table 10: I/O Physical Target Statistics**

| Field                               | Description                                                                                                                                                                                                                                                                                  |
|-------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Current Active Server Connections   | Cluster Physical Target: The current number of established TCP connections on the target server side of the DX that are involved in fulfilling a current HTTP request.<br><br>Forwarder Physical Target: The current number of established TCP connections on the target side of the DX.     |
| Current Idle Server Connections     | Cluster Physical Target Only: The current number of established TCP connections on the target server side of the DX that are NOT involved in fulfilling a current HTTP request.                                                                                                              |
| Total Server Connections            | Cluster Physical Target: Total number of TCP connections that any cluster has ever established (SYN, SYN-ACK, ACK) to a physical target.<br><br>Forwarder Physical Target: Total number of TCP connections that any forwarder has ever established (SYN, SYN-ACK, ACK) to a physical target. |
| Target Status                       | Cluster Physical Target Level Only (no aggregation): Indicates the connection status to the backend Web server (e.g., Up, Layer 7 Down, Transport Protocol Failure, etc.).                                                                                                                   |
| Health Check Status                 | Cluster Physical Target Level Only (no aggregation): Indicates whether health checking is currently enabled or disabled for a physical target.                                                                                                                                               |
| Passed Health Checks (servers okay) | Cluster Physical Target only: Total number of health checks that have ever passed for a physical target.                                                                                                                                                                                     |
| Failed Health Checks (servers down) | Cluster Physical Target only: Total number of health checks that have ever failed for a physical target.                                                                                                                                                                                     |

## HTTP Listen Statistics: Requests from Clients

To display the HTTP statistics for a specific cluster, type the command:

```
dx% show cluster <name> stats http
```

To display the HTTP statistics for all clusters, type the command:

```
dx% show cluster all stats http
```

To display the HTTP statistics for a specific target host within a cluster, type the command:

```
dx% show cluster <name> target host <name> stats http
```

To display the HTTP statistics for all target hosts within a cluster, type the command:

```
dx% show cluster <name> target host all stats http
```

To display the HTTP statistics for the DX server, type the command:

```
dx% show server stats http
```

HTTP Listen statistics can be shown at the cluster and server levels.

In Table 11, a “legal” HTTP request is defined as one in which the request line and request headers conform to HTTP standards.

**Table 11: HTTP Listen Statistics: Requests from Clients**

| Field                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  | Description                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|--------------------------------------------------------------------------------------------------------|
| Requests Active (no reply yet)                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Current number of HTTP requests for which the HTTP headers and data are being processed.               |
| Requests Total                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         | Total number of legal AND illegal HTTP requests that have been received by the DX.                     |
| Method GET<br>Method HEAD<br>Method POST<br>Method PUT<br>Method DELETE<br>Method TRACE<br>Method OPTIONS<br>Method CONNECT                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            | Total number of legal HTTP requests that have been received with the given HTTP method.                |
| Method PROPFIND<br>Method PROPPATCH<br>Method MKCOL<br>Method COPY<br>Method MOVE<br>Method LOCK<br>Method UNLOCK<br>Method BCOPY<br>Method BDELETE<br>Method BMOVE<br>Method BPROPFIND<br>Method BPROPPATCH<br>Method NOTIFY<br>Method POLL<br>Method SEARCH<br>Method SUBSCRIBE<br>Method UNSUBSCRIBE<br>Method X_MS_ENUMATTS<br>Method<br>VERSION_CONTROL<br>Method REPORT<br>Method CHECKOUT<br>Method CHECKIN<br>Method UNCHECKOUT<br>Method MKWORKSPACE<br>Method UPDATE<br>Method LABEL<br>Method MERGE<br>Method<br>BASELINE_CONTROL<br>Method MKACTION<br>Method BIND<br>Method MKRESOURCE<br>Method ORDERPATCH<br>Method ACL<br>Method Other | Total number of legal HTTP requests that have been received with the given HTTP method.<br>(Continued) |

**Table 11: HTTP Listen Statistics: Requests from Clients**

| Field                         | Description                                                                              |
|-------------------------------|------------------------------------------------------------------------------------------|
| Version HTTP/1.1              | Total number of legal HTTP requests that have been received with the given HTTP version. |
| Version HTTP/1.0              |                                                                                          |
| Version Other                 |                                                                                          |
| Browser IE 6.0                | Total number of legal HTTP requests that have been received from the given HTTP browser. |
| Browser IE 5.5                |                                                                                          |
| Browser IE 5.1                |                                                                                          |
| Browser IE 5.0                |                                                                                          |
| Browser IE 4.x                |                                                                                          |
| Browser IE Other              |                                                                                          |
| Browser Netscape 4            |                                                                                          |
| Browser Netscape 6            |                                                                                          |
| Browser Mozilla               |                                                                                          |
| Browser Opera                 |                                                                                          |
| Browser Konquerer             |                                                                                          |
| Browser Safari                |                                                                                          |
| Browser None                  |                                                                                          |
| Browser Other                 |                                                                                          |
| Illegal request line too long | Total number of illegal HTTP requests that have been received in the given categories.   |
| Illegal method                |                                                                                          |
| Illegal 0.9 method            |                                                                                          |
| Illegal POST (no length)      |                                                                                          |
| Illegal POST (length < 0)     |                                                                                          |
| Illegal POST (length = 0)     |                                                                                          |
| Illegal header                |                                                                                          |
| Illegal header line too long  |                                                                                          |
| Illegal PUT (no length)       |                                                                                          |
| Illegal PUT (length < 0)      |                                                                                          |
| Illegal PUT (length = 0)      |                                                                                          |
| Disallowed HTTP Method        |                                                                                          |
| Disallowed WebDAV Method      |                                                                                          |

### HTTP Target Host Statistics

HTTP target host statistics can be shown at the cluster target host, cluster, and server levels (refer to Table 12).

**Table 12: HTTP Target Host Statistics**

| Field                          | Description                                                         |
|--------------------------------|---------------------------------------------------------------------|
| Responses from servers:        | Total number of HTTP responses with the given response code values. |
| ** Total 1XX Response Codes ** |                                                                     |
| Response Code 100              |                                                                     |
| Response Code 101              |                                                                     |
| Response Code 102              |                                                                     |
| ** Total 2XX Response Codes ** |                                                                     |
| Response Code 200              |                                                                     |
| Response Code 201              |                                                                     |
| Response Code 202              |                                                                     |
| Response Code 203              |                                                                     |
| Response Code 204              |                                                                     |
| Response Code 205              |                                                                     |
| Response Code 206              |                                                                     |
| Response Code 207              |                                                                     |
| ** Total 3XX Response Codes ** |                                                                     |
| Response Code 300              |                                                                     |
| Response Code 301              |                                                                     |
| Response Code 302              |                                                                     |
| Response Code 303              |                                                                     |
| Response Code 304              |                                                                     |
| Response Code 305              |                                                                     |
| Response Code 306              |                                                                     |
| Response Code 307              |                                                                     |
| ** Total 4XX Response Codes ** |                                                                     |
| Response Code 400              |                                                                     |
| Response Code 401              |                                                                     |
| Response Code 402              |                                                                     |
| Response Code 403              |                                                                     |
| Response Code 404              |                                                                     |
| Response Code 405              |                                                                     |
| Response Code 406              |                                                                     |
| Response Code 407              |                                                                     |
| Response Code 408              |                                                                     |
| Response Code 409              |                                                                     |
| Response Code 410              |                                                                     |
| Response Code 411              |                                                                     |
| Response Code 412              |                                                                     |
| Response Code 413              |                                                                     |
| Response Code 414              |                                                                     |
| Response Code 415              |                                                                     |
| Response Code 416              |                                                                     |
| Response Code 417              |                                                                     |



**Table 12: HTTP Target Host Statistics**

| Field                          | Description                                                                                                                                                                             |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Response Code 422              | Total number of HTTP responses with the given response code values.<br>(Continued)                                                                                                      |
| Response Code 423              |                                                                                                                                                                                         |
| Response Code 424              |                                                                                                                                                                                         |
| Response Code 425              |                                                                                                                                                                                         |
| ** Total 5XX Response Codes ** |                                                                                                                                                                                         |
| Response Code 500              | Total number of HTTP responses that contain body content with the given content type.                                                                                                   |
| Response Code 501              |                                                                                                                                                                                         |
| Response Code 502              |                                                                                                                                                                                         |
| Response Code 503              |                                                                                                                                                                                         |
| Response Code 504              |                                                                                                                                                                                         |
| Response Code 505              |                                                                                                                                                                                         |
| Response Code 506              |                                                                                                                                                                                         |
| Response Code 507              |                                                                                                                                                                                         |
| Response Code Other            |                                                                                                                                                                                         |
| Content types from servers:    |                                                                                                                                                                                         |
| Content GIF                    |                                                                                                                                                                                         |
| Content JPEG                   |                                                                                                                                                                                         |
| Content HTML                   |                                                                                                                                                                                         |
| Content CSS                    |                                                                                                                                                                                         |
| Content XML                    |                                                                                                                                                                                         |
| Content PLAIN                  |                                                                                                                                                                                         |
| Content X-COMPONENT            |                                                                                                                                                                                         |
| Content JAVASCRIPT             |                                                                                                                                                                                         |
| Content FLASH                  |                                                                                                                                                                                         |
| Content OCTET-STREAM           |                                                                                                                                                                                         |
| Content MS-WORD                |                                                                                                                                                                                         |
| Content MS-EXCEL               |                                                                                                                                                                                         |
| Content MS-POWERPOINT          |                                                                                                                                                                                         |
| Content Custom-1               |                                                                                                                                                                                         |
| Content Custom-2               |                                                                                                                                                                                         |
| Content Custom-3               |                                                                                                                                                                                         |
| Content Other                  |                                                                                                                                                                                         |
| Content bytes from servers:    | Total number of HTTP response body bytes received with the given content type (excluding chunk headers) before the DX performs its HTTP-level response body processing and compression. |
| Bytes In GIF                   |                                                                                                                                                                                         |
| Bytes In JPEG                  |                                                                                                                                                                                         |
| Bytes In HTML                  |                                                                                                                                                                                         |
| Bytes In CSS                   |                                                                                                                                                                                         |
| Bytes In XML                   |                                                                                                                                                                                         |
| Bytes In PLAIN                 |                                                                                                                                                                                         |
| Bytes In X-COMPONENT           |                                                                                                                                                                                         |
| Bytes In JAVASCRIPT            |                                                                                                                                                                                         |
| Bytes In HTML                  |                                                                                                                                                                                         |
| Bytes In CSS                   |                                                                                                                                                                                         |
| Bytes In XML                   |                                                                                                                                                                                         |
| Bytes In PLAIN                 |                                                                                                                                                                                         |
| Bytes In X-COMPONENT           |                                                                                                                                                                                         |
| Bytes In JAVASCRIPT            |                                                                                                                                                                                         |
| Bytes In FLASH                 |                                                                                                                                                                                         |
| Bytes In OCTET-STREAM          |                                                                                                                                                                                         |
| Bytes In MS-WORD               |                                                                                                                                                                                         |
| Bytes In MS-EXCEL              |                                                                                                                                                                                         |
| Bytes In MS-POWERPOINT         |                                                                                                                                                                                         |
| Bytes In Custom-1              |                                                                                                                                                                                         |
| Bytes In Custom-2              |                                                                                                                                                                                         |
| Bytes In Custom-3              |                                                                                                                                                                                         |
| Bytes In Other                 |                                                                                                                                                                                         |

**Table 12: HTTP Target Host Statistics**

| Field                                                                                                                                                                                                                                                                                                                                                                                                                    | Description                                                                                                                                                            |
|--------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Content bytes to clients:<br>Bytes Out GIF<br>Bytes Out JPEG<br>Bytes Out HTML<br>Bytes Out CSS<br>Bytes Out XML<br>Bytes Out PLAIN<br>Bytes Out X-COMPONENT<br>Bytes Out JAVASCRIPT<br>Bytes Out FLASH<br>Bytes Out OCTET-STREAM<br>Bytes Out MS-WORD<br>Bytes Out MS-EXCEL<br>Bytes Out MS-POWERPOINT<br>Bytes Out Custom-1<br>Bytes Out Custom-2<br>Bytes Out Custom-3<br>Bytes Out Other                             | Total number of HTTP response body bytes with the given content type that are remaining after the DX performs its HTTP-level response body processing and compression. |
| Compressed content bytes to clients:<br>Compressed GIF<br>Compressed JPEG<br>Compressed HTML<br>Compressed CSS<br>Compressed XML<br>Compressed PLAIN<br>Compressed X-COMPONENT<br>Compressed JAVASCRIPT<br>Compressed FLASH<br>Compressed OCTET-STREAM<br>Compressed MS-WORD<br>Compressed MS-EXCEL<br>Compressed MS-POWERPOINT<br>Compressed Custom-1<br>Compressed Custom-2<br>Compressed Custom-3<br>Compressed Other | Total number of HTTP response body bytes with the given content type that were compressed and sent to the client.                                                      |

## SSL Listen Statistics

SSL listen statistics can be shown at the cluster and redirector, and server levels (refer to Table 13).

**Table 13: SSL Listen Statistics**

| Field                                                            | Description                                                                                                           |
|------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------|
| Session New                                                      | Total number of new SSL sessions that clients have established with the DX.                                           |
| Sessions Reused                                                  | Total number of reused SSL sessions that clients have established to the DX.                                          |
| Encryption Strong                                                | Total number of SSL sessions with 128-bit or higher level bulk encryption that clients have established with the DX.  |
| Encryption Export                                                | Total number of SSL sessions with lower than 128-bit level bulk encryption that clients have established with the DX. |
| Version SSLv2<br>Version SSLv3<br>Version TLSv1<br>Version Other | Total number of SSL sessions with the given version that clients have established with the DX.                        |

## SSL Target Host Statistics

SSL target host statistics can be shown at the cluster target host, cluster and server levels (refer to Table 14). To display the SSL statistics for a specific cluster, type the command:

```
dx% show cluster <name> stats ssl
```

To display the SSL statistics for all clusters, type the command:

```
dx% show cluster all stats ssl
```

To display the SSL statistics for a specific target host within a cluster, type the command:

```
dx% show cluster <name> target host <name> stats ssl
```

To display the SSL statistics for all target hosts within a cluster, type the command:

```
dx% show cluster <name> target host all stats ssl
```

To display the SSL statistics for the DX server, type the command:

```
dx% show server stats ssl
```

**Table 14: SSL Target Host Statistics**

| Field                                                            | Description                                                                                                                |
|------------------------------------------------------------------|----------------------------------------------------------------------------------------------------------------------------|
| Session New                                                      | Total number of new SSL sessions that the DX has established with a target host.                                           |
| Sessions Reused                                                  | Total number of reused SSL sessions that clients have established to the DX.                                               |
| Encryption Strong                                                | Total number of SSL sessions with 128-bit or higher level bulk encryption that the DX has established with a target host.  |
| Encryption Export                                                | Total number of SSL sessions with lower than 128-bit level bulk encryption that the DX has established with a target host. |
| Version SSLv2<br>Version SSLv3<br>Version TLSv1<br>Version Other | Total number of SSL sessions with the given version that the DX has established with a target host.                        |

### **DXSHELL Commands for Advanced Statistics**

In the examples below, the `<name>` field represents cluster and target names or numbers (1, 2, 3, etc.).

#### **Cluster Statistics**

To display all the statistics for a specific cluster, type the command:

```
dx% show cluster <name> stats
```

To display all statistics for all clusters, type the command:

```
dx% show cluster all stats
```

To display the I/O statistics for a specific cluster, type the command:

```
dx% show cluster <name> stats io
```

To display the I/O statistics for all clusters, type the command:

```
dx% show cluster all stats io
```

To display the HTTP statistics for a specific cluster, type the command:

```
dx% show cluster <name> stats http
```

To display the HTTP statistics for all clusters, type the command:

```
dx% show cluster all stats http
```

To display the SSL statistics for a specific cluster, type the command:

```
dx% show cluster <name> stats ssl
```

To display the SSL statistics for all clusters, type the command:

```
dx% show cluster all stats ssl
```

**Cluster Target Host Statistics**

To display the I/O statistics for a specific target host within a cluster, type the command:

```
dx% show cluster <name> target host <name> stats io
```

To display the I/O statistics for all target hosts within a cluster, type the command:

```
dx% show cluster <name> target host all stats io
```

To display the HTTP statistics for a specific target host within a cluster, type the command:

```
dx% show cluster <name> target host <name> stats http
```

To display the HTTP statistics for all target hosts within a cluster, type the command:

```
dx% show cluster <name> target host all stats http
```

To display the SSL statistics for a specific target host within a cluster, type the command:

```
dx% show cluster <name> target host <name> stats ssl
```

To display the SSL statistics for all target hosts within a cluster, type the command:

```
dx% show cluster <name> target host all stats ssl
```

**Clearing Cluster Statistics**

To clear statistics for a specified cluster, type the command:

```
dx% clear cluster <name> stats
```

To clear statistics for all clusters, type the command:

```
dx% clear cluster all stats
```

Clearing the statistics resets the counter values to 0.

**Forwarder Statistics**

To display all forwarder statistics, type the command:

```
dx% show forwarder all stats
```

To display all a specific forwarder statistics, type the command:

```
dx% show forwarder <name> stats
```

### **Forwarder's Target Host Statistics**

To display statistics for a specific target host within a forwarder, type the command:

```
dx% show forwarder <name> target host <name> stats
```

To display statistics for all target hosts within a forwarder, type the command:

```
dx% show forwarder <name> target host all stats
```

### **Clearing Forwarder Statistics**

To clear statistics for a specified forwarder, type the command:

```
dx% clear forwarder <name> stats
```

To clear statistics for all forwarders, type the command:

```
dx% clear forwarder all stats
```

### **Redirector Statistics**

To display all the statistics for a specific redirector, type the command:

```
dx% show redirector <name> stats
```

To display all statistics for all redirectors, type the command:

```
dx% show redirector all stats
```

To display the I/O statistics for a specific redirector, type the command:

```
dx% show redirector <name> stats io
```

To display the I/O statistics for all redirectors, type the command:

```
dx% show redirector all stats io
```

To display the SSL statistics for a specific redirector, type the command:

```
dx% show redirector <name> stats ssl
```

To display the SSL statistics for all redirectors, type the command:

```
dx% show redirector all stats ssl
```

### **Clearing Redirector Statistics**

To clear statistics for a specified redirector, type the command:

```
dx% clear redirector <name> stats
```

To clear statistics for all redirectors, type the command:

```
dx% clear redirector all stats
```

## ***DX Appliance Server Statistics***

To display all statistics for the DX server, type the command:

```
dx% show server stats
```

To display a one-line summary of DX server statistics updated every n seconds, type the command:

```
dx% show server stats <n>
```

To display the I/O statistics for the DX server, type the command:

```
dx% show server stats io
```

To display the HTTP statistics for the DX server, type the command:

```
dx% show server stats http
```

To display the SSL statistics for the DX server, type the command:

```
dx% show server stats ssl
```

## ***Clearing DX Appliance Server Statistics***

To clear statistics for the DX server, type the command:

```
dx% clear server stats
```

## **Web Log Configuration**

---

Maintaining a Web Log provides vital information for analyzing your Web site's traffic. The DX provides the ability to enable a Web Log for a cluster:

```
set cluster <name> weblog [enabled |disabled]
```

When it is enabled, the DX generates a log entry for each HTTP request that it handles.

The DX can be configured to transmit the logs to the Syslog server in one of two ways. The default configuration is Immediate mode, where the DX appliance immediately writes a User Datagram Protocol (UDP) packet containing a Web log to the configured Syslog server for each client request. Immediate mode can create a significant amount of extra network activity and does not allow the ability to save logs.

The alternative is Web Log Batch mode. In Web Log Batch mode, Web logs are saved on the DX appliance and then copied off in bulk format. For more information, see "Web Log Batch Mode" on page 408.

The user can select the format for the log from one of these five options:

- Common: This is the Apache Common Logging Format (CLF). The information included in the log is:

```
remotehost remotelogname authuser [date] "request" status bytes
```

- Combined: This is a modification of CLF (common) format and adds the values of the Referer and User-Agent HTTP headers in quotes:

```
remotehost remotelogname authuser [date] "request" status bytes "Referer" "User-Agent"
```

- Common\_cn: This is a modification of CLF (common) format with the cluster name prepended to the CLF format:

```
cclustername remotehost remotelogname authuser [date] "request" status bytes
```

- Combined\_cn: This is a modification of the combined format with the cluster name prepended to the combined format:

```
cclustername remotehost remotelogname authuser [date] "request" status bytes "Referer" "User-Agent"
```

- Perf1: This is a proprietary format that allows you to more easily monitor the performance of DX appliance compression and cache. The information included in the log is:

```
remotehost [date] method url version status request-bytes precomp-bytes postcomp-bytes cachehit
```

- Perf2: This is a proprietary format that allows you to troubleshoot performance problems. The information included in the log is:

```
TransactionID remotehost T1 T2 T3 T4 tcp_outcome method url status from ip_port bytes cachehit host "user-agent"
```

The information fields included in the logs are defined in Table 15.

**Table 15: Web Log Field Definitions**

| Field         | Definition                                                                                       |
|---------------|--------------------------------------------------------------------------------------------------|
| remotehost    | The remote hostname (or IP address if the DNS hostname is not available, or if DNSLookup is Off) |
| remotelogname | The remote logname of the user                                                                   |
| authuser      | The username with which the user authenticated himself                                           |
| [date]        | The date and time of the request inside brackets ([ ])                                           |
| "request"     | The request line exactly as it came from the client inside quotes (" ")                          |
| status        | The HTTP status code returned to the client                                                      |
| bytes         | The content-length of the document transferred for response                                      |
| "referer"     | The value of the Referer header inside quotes (" ")                                              |
| "user-agent"  | The value of the User-Agent header inside quotes (" ")                                           |
| clustername   | The name of the cluster that received the request                                                |
| method        | The request method                                                                               |



**Table 15: Web Log Field Definitions (continued)**

| Field          | Definition                                                                                                                                             |
|----------------|--------------------------------------------------------------------------------------------------------------------------------------------------------|
| url            | The request URL                                                                                                                                        |
| version        | The request version with the format "HTTP/ < major > . < minor > " (without the quotes)                                                                |
| request-bytes  | The length of request content-body. This is applicable for POST, PUT, and certain WebDAV requests.                                                     |
| precomp-bytes  | The content-length of the response document before compression                                                                                         |
| postcomp-bytes | The content-length of the response document after compression                                                                                          |
| cachehit       | The number of Juniper cache hits or cache misses                                                                                                       |
| ip_port        | This is the IP address and port number for the chosen target server.                                                                                   |
| from           | This is the "From" request header.                                                                                                                     |
| result         | This is the download result. For example, received ack of the last byte of the object.                                                                 |
| transactionID  | This is the TransactionID response header.                                                                                                             |
| T1             | T1 is a timestamp of the request's arrival completion time on the DX.                                                                                  |
| T2             | T2 is a count of the seconds from T1 until the DX receives the first byte of response from the target server.                                          |
| T3             | T3 is a count of the seconds from T2 until the DX receives the last byte of response from the target server.                                           |
| T4             | T4 is a count of the seconds from T3 until the DX receives the TCP ACK of the last byte of data for this object.                                       |
| tcp_outcome    | Indicates whether the last TCP ACK was received. Valid values included OK, if the last TCP ACK is received or NG, if the last TCP ACK is not recieved. |
| host           | Value of the Host header.                                                                                                                              |

## Web Log Commands

The Web Log is sent out to a Syslog host (server), and the Syslog host must be configured properly before enabling the Web Log feature.

### Enabling the Web Log

To configure the Syslog host that will receive the Web Log, type the commands:

```
dx% set cluster <name> weblog syslog host [ip address]
dx% set cluster <name> weblog syslog port [port]
```

The first command sets the host ip address for the Web Log Syslog host. The second command sets the destination TCP port for the Web Log Syslog host. The default port is 514. By default, Web Log messages with the destination Syslog use Local3 as their facility.

To enable the Web Log format, type the command:

```
dx% set cluster <name> weblog format <fmt>
```

This command sets the format, which can be one of common, combined, common\_cn, combined\_cn, perf1, or perf2.

The delimiter in the Web Log can be set to be either a comma or a space. To set the delimiter, type the command:

```
dx% set cluster <name> weblog delimiter <comma | space>
```

To enable the Web Log feature, type the command:

```
dx% set cluster <name> weblog enabled
```

To disable the Web Log feature, type the command:

```
dx% set cluster <name> weblog disabled
```

### Showing the Web Log Configuration

To show the configuration of the Web Log Syslog host, use the commands:

```
dx% show cluster <name> weblog syslog
dx% show cluster <name> weblog syslog host
dx% show cluster <name> weblog syslog port
dx% show cluster <name> weblog syslog format
```

### Clearing the Web Log Configuration

To clear the configuration of the Web Log Syslog host, use the command:

```
dx% clear cluster <name> weblog syslog host
```

## Web Log Batch Mode

In Web Log Batch mode, Web Logs are saved on the DX appliance and then copied off to an SCP server in bulk format at specified times. The Web Log information that is sent is identical to the information sent in Immediate mode, however it is sent in a batch instead.

The Web Log is stored in compressed format in one of two data stores. The size of these data stores are user-configurable. The second data store is only used when the first data store fills up and the log file cannot be successfully copied. If both data stores fill up and the copy does not succeed, the first data store is purged and filled with new data.

The Web Log is saved on the DX until one of the following events occurs:

- A successful copy is completed
- Both buffers are full
- The DX is rebooted
- You resize the buffer
- You delete or rename the cluster

The Web Log is transmitted securely to the configured Syslog server using Secure Copy (SCP) when:

- You force the DX to send it using a `set cluster <name> weblog batch copy copynow` command
- Maximum buffer size is reached
- User-configurable time is reached (the Alarm)
- Cluster is deleted or renamed

A log message of type EMERG is logged upon a copy failure. You can optionally configure EMERG events to be sent via E-mail.

Before using Web Log Batch mode, you must configure certain items on a per-cluster basis:

- Size of the compressed file to store Web logs
- Three alarms that set when to copy (HH:MM in 24 hour format)
- A retry interval if a copy fails (default value: 60 seconds; range: 30-1200 seconds)
- Destination server
- Destination directory
- Secure Copy (SCP) username
- Private key

The DX only supports SSH2 for scp file transfers (SSH1 is not supported). You must upload the private key (RSA or DSA) onto the DX. The private key is captured with a `capture file` command, and the key must not be password protected.

The file name for the copied Web logs is set to:

```
<DX hostname>_<cluster_name>_<date><time>.gz
```

The date/time format is YYYYMMDDHHMMSS, and the date and time are based on local time.

Batch Web Logs and Syslog Web Logs are mutually exclusive (only one can be enabled at a time). This feature is enabled with the `set cluster <n> weblog destination` command.

## Web Log Batch Commands

You must configure the Web Log Batch feature on a per-cluster basis.

### Configuring the Web Log Batch Feature

To determine whether Web log entries will be sent to the Syslog server immediately (syslog) or in a batch, type the command:

```
dx% set cluster <name> weblog destination [syslog | batch]
```

The syslog and batch options are mutually exclusive.

To set the size of the compressed file to copy (the size of the two data buffers), type the command:

```
dx% set cluster <name> weblog batch copy size [val in MB]
```

The default value is 10 MBytes, and the range is 1 to 50 MBytes.

To set the times for the Web Log to be transmitted to the configured SCP server, type the commands:

```
dx% set cluster <name> weblog batch copy time 1 [time]
dx% set cluster <name> weblog batch copy time 2 [time]
dx% set cluster <name> weblog batch copy time 3 [time]
```

The format for [time] is HH:MM. Up to three times can be configured for each day.

To configure the Web Log to be transmitted to the configured SCP server at periodic intervals, type the command:

```
dx% set cluster <name> weblog batch copy interval <minutes>
```

To force an immediate copy of the Web Logs, type the command:

```
dx% set cluster <name> weblog batch copy copynow
```

To set the retry interval (in seconds) in case of copy failure, type the command:

```
dx% set cluster <name> weblog batch failure retryinterval [val]
```

To set the remote SCP target directory, type the command:

```
dx% set cluster <name> weblog batch scp directory [directory]
```

To set the remote SCP username, type the command:

```
dx% set cluster <name> weblog batch scp username [user]
```

The private key must be captured using the `capture` command.

To set the (non-password protected) private key, type the command:

```
dx% set cluster <name> weblog batch scp keyfile [choose a file]
```

To test the connection, type the command:

```
dx% set cluster <name> weblog batch scp connecttest
```

This copies a one byte test file.

To set the host where the Web Log will be copied, type the command:

```
dx% set cluster <name> weblog batch host [server]
```

The Web log can either be sent to the SCP server in its native format or in compressed form. To enable or disable compression, type the command:

```
dx% set cluster <name> weblog batch compression [enable | disable]
```

The Web logs are compressed in gzip format.

Configuration commands may be executed by users with roles of Administrator and Network Administrator.

### ***Showing the Configuration of the Web Log Batch Feature***

To show all of the configuration parameters associated with the Web Log batch feature, type the command:

```
dx% show cluster <name> weblog batch
```

To show the size of the compressed file to copy (the size of the two data buffers), type the command:

```
dx% show cluster <name> weblog batch copy size
```

This command also shows the total remaining memory available for weblog batch storage.

To show all three of the times when the Web Log will be transmitted to the configured SCP server, type the command:

```
dx% show cluster <name> weblog batch copy time
```

To show the interval at which the Web Log will be transmitted to the configured SCP server, type the command:

```
dx% show cluster <name> weblog batch copy interval
```

To show the retry interval (in seconds) in case of copy failure, type the command:

```
dx% show cluster <name> weblog batch failure retryinterval
```

To show all of the configuration parameters associated with the remote SCP target directory, type the command:

```
dx% show cluster <name> weblog batch scp
```

To show the remote SCP target directory, type the command:

```
dx% show cluster <name> weblog batch scp directory
```

To show the remote SCP username, type the command:

```
dx% show cluster <name> weblog batch scp username
```

To show the (non-password protected) private key, type the command:

```
dx% show cluster <name> weblog batch scp keyfile
```

To show the host where the Web Log will be copied, type the command:

```
dx% show cluster <name> weblog batch host
```

To show if the Web Log will be sent to the SCP server in compressed form, type the command:

```
dx% show cluster <name> weblog batch compression
```

These commands may be executed by users with roles of Admin, Network Admin, Network Operator.

#### ***Clearing the Configuration of the Web Log Batch Feature***

To clear the times for the Web Log to be transmitted to the configured SCP server, type the commands:

```
dx% clear cluster <name> weblog batch copy time 1
dx% clear cluster <name> weblog batch copy time 2
dx% clear cluster <name> weblog batch copy time 3
```

To clear the (non-password protected) private key, type the command:

```
dx% clear cluster <name> weblog batch scp keyfile
```

These commands can be executed by users with roles of Administrator and Network Administrator.

## Chapter 23

# Troubleshooting

This chapter describes troubleshooting for the DX Application Acceleration Platform discussing the following topics:

- Checking Settings on page 413
- Troubleshooting on page 414
- Technical Service Dump on page 418
- Using tcpdump to Generate a Detailed Report of Network Activity on page 419

### Checking Settings

---

There are several commands that are useful when troubleshooting your DX. Use these commands to look at your system configuration. For a complete overview of configuration settings of the DX, use the command:

```
dx% show config
```

To obtain basic information about the DX, type the command:

```
dx% show system info
```

For more extensive information about the DX and the environment that it is operating in, type the command:

```
dx% show system debug
```

If you need to call Juniper Networks Technical Support, they will frequently ask for the information displayed by this command. For a complete list of all **show** commands and corresponding **set** commands, refer to the *Command Line Reference* manual or type **show** commands at the DXSHELL prompt.

## Troubleshooting

---

### *Slow or Degraded Performance*

#### **Are Media Settings for Ether 0 Correct?**

Mismatched or incorrect media settings will severely impair the performance of the DX. For 1U units, make sure that the media setting for Ether 0 is 100base DX full-duplex. These settings must also match those of the L2 switch port the DX is connected to. **DO NOT USE AUTOSELECT.**

For 2U units with gigabit Ethernet, make sure that the media for Ether 0 and the switch port the DX is connected to are both set to autoselect.

To view media settings, type the command:

```
dx% show ether 0
```

To specify correct media settings for most environments, type the command:

```
dx% set ether 0 media 100baseTX full-duplex
```

#### **Is HTTP 1.1 Enabled on the Target Hosts?**

In order for DX to maintain persistent connections with target hosts, the target hosts must be configured to support HTTP 1.1 with keep-alive enabled.

### ***DX Appliance is Not Responding to Requests for Web Content***

#### **Verify that the DX Appliance is Serving Web Pages**

To make sure that the DX is serving content from the target host, open a browser and enter one of the Virtual IP addresses you set on the DX (remember to enter the port number if it is set to something other than 80). You should see the home page of the target host(s).

**NOTE:** At LAN speed, the pages may not seem noticeably faster. In part, Web I/O Acceleration addresses the inefficiencies of long-haul and final-mile transfer in order to accelerate page download for people with slower modem and broadband connections. Therefore, acceleration may not be noticeable from within your LAN.

If the DX does not respond, double check your settings and consult the troubleshooting steps that follow.

#### **Are target Hosts Configured?**

For the DX to serve content, the DX must be configured with one or more clusters populated with target host(s). To view all configured clusters, enter the command:

```
dx% show cluster all
```

Check the output to verify your cluster and target host configuration.



### Is the DX Appliance Enabled?

If the DX is not responding to requests, check that the DX accelerator is enabled with the command:

```
dx% show server status
```

If the DX is down, bring it up with the command:

```
dx% set server up
```

Be sure to save your change with the command:

```
dx% write
```

### Has the DX Appliance Established TCP Connections to the Target Hosts?

Get a list of connections with the command:

```
dx% show netstat
```

Check the output for ESTABLISHED connections to target hosts.

```
Active Internet connections (including servers)
Proto Recv-Q Send-Q Local Address          Foreign Address        (state)
tcp4    0      0 10.0.11.20.11910      10.0.11.81.80         ESTABLISHED
tcp4    0      0 10.0.11.20.11908      10.0.11.81.80         ESTABLISHED
tcp4    0      0 10.0.11.20.11906      10.0.11.81.80         ESTABLISHED
tcp4    0      0 10.0.11.20.11904      10.0.11.81.80         ESTABLISHED
tcp4    0      0 10.0.11.20.11902      10.0.11.81.80         ESTABLISHED
tcp4    0      0 10.0.11.20.11898      10.0.11.81.80         ESTABLISHED
tcp4    0      0 10.0.11.120.80        *,*                    LISTEN
```

### Are the Target Hosts Visible to the DX Appliance?

From the DXSHELL command line, ping one of the target hosts, by typing the command:

```
dx% ping <IP address of the target host>
```

Pinging will stop after five packets on a DX. If the DX can connect to the target host, you should see something similar to this output:

```
PING 192.168.0.102 (192.168.0.102): 56 data bytes
64 bytes from 192.168.0.102: icmp_seq=0 ttl=128 time=0.228 ms
64 bytes from 192.168.0.102: icmp_seq=1 ttl=128 time=0.193 ms
64 bytes from 192.168.0.102: icmp_seq=2 ttl=128 time=0.186 ms
64 bytes from 192.168.0.102: icmp_seq=3 ttl=128 time=0.213 ms
64 bytes from 192.168.0.102: icmp_seq=4 ttl=128 time=0.237 ms
--- 192.168.0.102 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.186/0.207/0.237/0.020 ms
```

### Is the DX Appliance Visible from the Target Hosts?

From one of the target hosts, try pinging the DX with one of the following commands:

```
dx% ping <IP address of DX ether 0>
```

```
dx% ping <Virtual IP address of DX>
```

Pinging will stop after five packets on a DX. You should see something similar to this output:

```

PING 192.168.0.163 (192.168.0.163): 56 data bytes
64 bytes from 192.168.0.163: icmp_seq=0 ttl=255 time=0.219 ms
64 bytes from 192.168.0.163: icmp_seq=1 ttl=255 time=0.174 ms
64 bytes from 192.168.0.163: icmp_seq=2 ttl=255 time=0.174 ms
64 bytes from 192.168.0.163: icmp_seq=3 ttl=255 time=0.187 ms
64 bytes from 192.168.0.163: icmp_seq=4 ttl=255 time=0.181 ms
--- 192.168.0.163 ping statistics ---
5 packets transmitted, 5 packets received, 0% packet loss
round-trip min/avg/max/stddev = 0.172/0.184/0.219/0.016 ms

```

### Is DNS Working?

Try pinging a Web site from the DX to find out if the DX can resolve the site's domain by typing the command:

```
dx% ping www.google.com
```

If you get the response:

```
ping: cannot resolve www.google.com: Host name lookup failure
```

you can check the DNS settings with the command:

```
dx% show dns
```

You can set the DNS server with the command:

```
dx% set dns server <IP address of DNS server>
```

### Is Traffic Flowing Through the DX Appliance?

You can check that the DX is taking in and sending out data with the command:

```
dx% show server stats 2
```

While the stats are refreshing every 2 seconds, try hitting the DX with your Web browser. You can tell that the DX is handling traffic by watching the number of Sessions, Requests and Bytes In/Bytes Out increase as the statistics refresh.

```

Uptime: 2 days, 15:51
      Sess
      act      tot      act      tot      Bytes In  Bytes Out
      3        131      1        1.09K    1.33MB    1.09MB
      3        131      1        1.09K    1.39MB    1.16MB
      3        131      1        1.09K    1.44MB    1.20MB

```

To stop the stats, type the key sequence:

```
ctrl-c
```

## **Cannot Access the WebUI with your Web Browser**

### **Is the WebUI enabled?**

Check that the WebUI is enabled with the command:

```
dx% show admin webui status
```

If the WebUI server is down, you can bring it up with the command:

```
dx% set admin webui up
```

### **Are you Including the Port When You Enter the Address in your Browser?**

Check which port the WebUI is listening on with the command:

```
dx% show admin webui port
```

Combine the port with the IP address of ether 0 to form the URL you use to access the WebUI. For example, if the IP of ether 0 was 10.0.11.20 and the admin port was 8090, you would use this URL to access the WebUI:

```
http://10.0.11.20:8090
```

**NOTE:** It is possible to configure WebUI administrator to listen on an IP (10.0.20.0, for example) and use port 8090. At the same time, a cluster of target hosts may be configured to use the same IP and port (10.0.20.0:8090). When a configuration change is made that requires a restart of the multiplexing engine, a WebUI administrator page could be displayed. To prevent this from occurring, you should not use the administrator port as a cluster port.

## **Cannot Connect to the DXSHELL Command Line with SSH**

### **Is SSH Service Enabled?**

Connect through the serial console or the WebUI and check that SSH service is enabled. From the DXSHELL command line, type:

```
dx% show admin ssh
```

If SSH service is down, you can enable it with the command:

```
dx% set admin ssh up
```

## Technical Service Dump

---

The DX can create a complete snapshot of its status intended to accompany support requests to help with remote troubleshooting. All information contained in the dump is available to the user through various commands. The dump is provided as a convenience for expediting the resolution of support requests.

### What Information is Collected

- Current configuration
- Data traffic statistics
- System event log information

### What Information is not Collected

- Passwords
- SSL keys
- SSL certificates

### Creating the Technical Service Dump

Before running `tsdump`, you will need to configure a few settings that tell the DX what to do with the `tsdump` file.

1. Choose whether you want to send the `tsdump` file via e-mail or copy it to your TFTP server. You must choose either e-mail or TFTP; you cannot use both.
  - To send the `tsdump` via E-mail using your SMTP server:
2. Configure the DX to output `tsdumps` to E-mail by typing the command:
 

```
dx% set admin tsdump transport smtp
```
3. Specify an SMTP server that the DX can use to relay E-mail by typing the command:
 

```
dx% set admin email server <IP address of SMTP server>
```
4. Set a name for the `tsdump` file by typing the command:
 

```
dx% set admin tsdump filename <filename>
```



**NOTE:** On some operating systems, including most UNIX-like systems, TFTP upload requires an existing, writable file with the same filename on the remote host.

---

5. Specify the from E-mail address that should appear in the E-mail by typing the command:
 

```
dx% set admin email from <e-mail address>
```

- Specify up to two different E-mail addresses to send the tsdump to by typing the command:

```
dx% set admin tsdump mailto1 <e-mail address>
dx% set admin tsdump mailto2 <e-mail address>
```

- To copy the tsdump file to your TFTP server, tell the DX to output tsdumps to a TFTP server by typing the command:

```
dx% set admin tsdump transport tftp
```

- Tell the DX which TFTP server to use with the command:

```
dx% set admin tftp server <IP address of TFTP server>
```

- Set a name for the tsdump file by typing the command:

```
dx% set admin tsdump filename <filename>
```



**NOTE:** On some operating systems, including most UNIX-like systems, TFTP upload requires an existing, writable file with the same filename on the remote host.

- Finally, after completing the required tsdump settings, create and send the tsdump with the command:

```
dx% tsdump
```

## Using tcpdump to Generate a Detailed Report of Network Activity

The tcpdump command-line utility provides a detailed report of network activity that can be useful for troubleshooting. A DX administrator can view who is accessing the DX, what Web sites are being viewed, and so forth by capturing the HTTP traffic between the DX and the target servers.

The tcpdump command is defined as follows:

```
tcpdump [-i etherN] [-print] [filter]
```

It has the following options:

| Option    | Action                                                                                                   |
|-----------|----------------------------------------------------------------------------------------------------------|
| filter    | Filter expression in file.                                                                               |
| -i etherN | Listen on multiple Ethernet interfaces, where N is between zero and the number of the highest interface. |
| -print    | Display the file on-screen. This option is only available when a single interface is specified.          |

The filename of the tcpdump file created uses the following format:

```
tcpdump.MMDDHHMMSS-NN
```

where MM is the month, DD is the day, HH is the hour (in 24 hour format), MM is the minutes, SS is the seconds, and NN is a random number.

### **Guidelines for Using tcpdump**

When using tcpdump to monitor your DX, use the following guidelines:

- If you reboot your DX appliance, all tcpdump files are discarded.
- When running multiple tcpdump commands concurrently, it is possible for one of the processes to fail due to the available storage space.
- The tcpdump file is updated each time 16 KB of data has accumulated.

### **Setting up Your DX for tcpdump**

Before running the tcpdump utility you must configure the DX with information about where to send the tcpdump file—to up to two E-mail addresses, or copy it to your TFTP or SCP server. Optionally, you may set the size of the filesystem to adjust for larger files.

#### **Sending the tcpdump file to an E-mail Address**

To set up Email addresses to receive the tcpdump file:

1. Specify the “from” E-Mail address that should appear in the E-mail:

```
dx% set admin email from <e-mail address>
```

2. Specify up to two different E-Mail addresses to receive the tcpdump file:

```
dx% set admin tcpdump mailto1 <email address>
dx% set admin tcpdump mailto2 <email address>
```

#### **Setting the Amount of Data to Capture**

To set the amount of data captured when you execute the tcpdump utility, you specify the size of the filesystem, from 10 to 75 MB. The default configuration is 10 MB.

To specify the capture size, enter:

```
dx% set admin tcpdump capturesize [value]
```

### **Running the tcpdump Utility**

With the 5.2 release of the DX Application Acceleration Platform, you can capture the network activity on one or more interfaces simultaneously. When you are capturing information for a single interface, you have the option of printing the tcpdump file to your screen.

To capture data for a single interface, use the `tcpdump` command. For example, to capture data on the Ether0 interface and print it to your screen, enter:

```
dx% tcpdump -i ether0 -print
tcpdump: listening on ether0
14:52:11.984715 192.168.9.119.3135 > 10.10.26.16.23: . ack 2301082499 win
64705 (DF)
14:52:11.984740 10.10.26.16.23 > 192.168.9.119.3135: P 1:31(30) ack 0 win
58080 (DF) [tos 0x10]
...
14:52:12.183852 192.168.9.119.3135 > 10.10.26.16.23: . ack 31 win 64675 (DF)
14:49:15.164512 10.10.26.16.23 > 192.168.9.119.3135: P 6032:6293(261) ack 2
win 5
8080 (DF) [tos 0x10]
14:49:15.364248 192.168.9.119.3135 > 10.10.26.16.23: . ack 6293 win 64282
(DF)
^C
80 packets received by filter
0 packets dropped by kernel
```



**NOTE:** If no interface is specified, `tcpdump` defaults to Ether0.

---

To capture data on Ether0 and Ether1 and have it sent to an Email address or other server, enter:

```
dx% tcpdump -i ether0 -i ether1
Byte counter will be refreshed after 16KB chunks are read.
^C
Bytes processed: 4237
Closing tcpdump...
Done.
Successfully wrote 'tcpdump.0601145111-40' (4237B)
```

## Managing tcpdump Files

You can view, copy, and remove `tcpdump` files at any time. These tasks are described in the following sections:

- “Viewing a `tcpdump` File on the DX Appliance” on page 421
- “Viewing a `tcpdump` Outside the DX Appliance” on page 422
- “Removing a `tcpdump` File” on page 423

### Viewing a `tcpdump` File on the DX Appliance

After creating a `tcpdump` file on the DX you can immediately view it. If you know the name of the file, you can view it directly, otherwise you can view the list of `tcpdump` files that have been created and select the one you wish to view. The listing indicates the size of each `tcpdump` file.

For example:

```
dx% show tcpdump
```

Available files:

```
tcpdump.0601145111-40 (4237B)
```

Usage: show tcpdump <tcpdump file> [-s <keyfile>]

```
dx% show tcpdump tcpdump.0601145111-40
```

```
14:49:48.973963 192.168.9.119.3135 > 10.00.26.16.23: . ack 2301079095 win 65165
```

```
14:49:48.973989 10.80.96.16.23 > 172.23.9.119.3135: P 1:80(79) ack 0 win 58080
```

```
14:49:49.175100 192.168.9.119.3135 > 10.00.26.16.23: . ack 80 win 65086 (DF)
```

```
14:49:55.716119 10.10.26.32.1051 > 10.00.26.16.9500: udp 268
```

```
14:50:01.776028 10.10.26.32.1051 > 10.00.26.16.9500: udp 268
```

```
14:50:07.835935 10.10.26.32.1051 > 10.00.26.16.9500: udp 268
```

```
14:50:13.895843 10.10.26.32.1051 > 10.00.26.16.9500: udp 268
```

```
14:50:19.955751 10.10.26.32.1051 > 10.00.26.16.9500: udp 268
```

```
14:50:26.015658 10.10.26.32.1051 > 10.00.26.16.9500: udp 268
```

```
14:50:32.075567 10.10.26.32.1051 > 10.00.26.16.9500: udp 268
```

```
14:50:38.135600 10.10.26.32.1051 > 10.00.26.16.9500: udp 268
```

```
14:50:45.205450 10.10.26.32.1051 > 10.00.26.16.9500: udp 268
```

```
14:50:51.265358 10.10.26.32.1051 > 10.00.26.16.9500: udp 268
```

```
14:50:57.325267 10.10.26.32.1051 > 10.00.26.16.9500: udp 268
```

```
14:51:03.385174 10.10.26.32.1051 > 10.00.26.16.9500: udp 268
```

If the file is encrypted, use the -s option to view the file and enter the keyfile password when prompted.

### Viewing a tcpdump Outside the DX Appliance

**NOTE:** If you are running release 2.3 or later, tcpdump output is already in binary format and you can skip to STEP 2. You can see what release you are running with the command `show version`.

1. Copy the tcpdump file off of the DX to an E-mail, TFTP or SCP server:

```
dx% copy tcpdump <tcpdumpfile> <destination>
```

For example, copy the tcpdump.0601145111-40 file to a pre-defined TFTP server, as follows:

```
dx% copy tcpdump tcpdump.0601145111-40 tftp://myserver/filename
```

2. Decode the base64-encoded tcpdump file using `uudecode` (UNIX) or `base64.exe` (Windows).
3. Once the file is decoded, you can view it using a standard tcpdump utility with the command:

```
dx% tcpdump -r <name of decoded tcpdump file>
```

You can also use a protocol analyzer such as Ethereal to view the decoded tcpdump.



### Removing a tcpdump File

When you are through with a particular tcpdump file or want to clear all tcpdump files to make room for new files, you can delete the one or all files using the `delete tcpdump` command.

To delete a particular file, enter:

```
dx% delete tcpdump <tcpdumpfile>
```

To delete all tcpdump files at once, enter:

```
dx% delete tcpdump all
```

### Sample TCPdump Scenario

In the following scenario, an administrator runs TCPdump and copies the results through SMTP to a “vaulted” location.

```
dx% tcpdump
```

```
Listening on ether0.  
Byte counter will be refreshed after 16KB chunks are read.  
^CBytes processed: 28510  
Closing tcpdump...  
Done.  
Successfully wrote 'tcpdump.0605094357-35' (27KB)
```

```
dx% set admin tcpdump mailto1 username@juniper.net
```

```
Email address changed.
```

```
(* dx% set admin email from admin@juniper.net
```

```
Email address changed.
```

```
(* dx% copy tcpdump tcpdump.0605094357-35 smtp://vault/tcpdump.dat
```



## Part 5

# Reference Information

This part of the *Installation and Administration Guide for DXOS* provides additional information that may be useful when you are configuring or monitoring your DX platform.

These materials can be found in the following appendices:

- Appendix A, “Glossary” on page 427
- Appendix B, “List of Events” on page 433
- Appendix C, “Configuring Failover by Service” on page 437

The Index can also be used to find information within this guide.



## Appendix A

# Glossary

**Table 7: Glossary (Sheet 1 of 6)**

| Term               | Definition                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                      |
|--------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Active-Active      | An Active-Active configuration is a two-DX configuration where both DXs are actively processing client traffic and load-balancing the client requests. One of the DXs is the token “Master” and if the Master DX fails, the remaining DX takes up the master role to take and redistribute the requests from clients.                                                                                                                                                                                           |
| Active-Standby     | An Active-Standby configuration is a two-DX configuration where one DX processes client traffic and load-balances the client requests (the active unit) while the other (standby) unit listens to the active unit’s heartbeat and waits to take over as the active unit in case of the active unit’s failure.                                                                                                                                                                                                   |
| ActiveN            | An ActiveN configuration is an extension of the two-DX Active-Active configuration where up to 64 DXs are actively processing client traffic and load-balancing the client requests. One of the DXs is the token “Master” and if the Master DX fails, one of the remaining DXs takes up the Master role to take and redistribute the requests from clients.                                                                                                                                                     |
| Blade              | A blade is a DX that has been configured as part of a Group.                                                                                                                                                                                                                                                                                                                                                                                                                                                    |
| “Busy” Redirect    | If the Target Web server responds with a “Busy” error, the Web I/O Accelerator will serve the page specified by this URL instead.                                                                                                                                                                                                                                                                                                                                                                               |
| Certfile           | Certification file for SSL traffic.                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Cipher             | Cryptographic algorithm for a server and client to authenticate each other, transmit certificates, and establish session keys.                                                                                                                                                                                                                                                                                                                                                                                  |
| Ciphersuite        | A set of ciphers.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Cluster            | A cluster is a collection of Web servers that are all configured to serve the same content for a single Web site, and to be accelerated by the DX. The DX listens for incoming Web traffic on a specific virtual IP address and port, distributes it over the target hosts (Web servers) in the cluster and then accelerates the outgoing Web traffic. Typically all the Web servers in a particular cluster serve identical content; that is, each cluster usually represents a distinct Web site or property. |
| Convert302protocol | Converts the 302 responses from HTTP to HTTPS or from HTTPS to HTTP.                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Customiplogheader  | A special header to annotate the log; showing the session that is being logged in an easily identifiable way.                                                                                                                                                                                                                                                                                                                                                                                                   |
| Custom Header      | This is custom HTTP header that will be added with the client’s origin IP to the client’s request.                                                                                                                                                                                                                                                                                                                                                                                                              |
| Default Route      | Also known as the “Gateway,” this is the IP address of the machine the Web I/O Accelerator talks with in order to access the outside world.                                                                                                                                                                                                                                                                                                                                                                     |

**Table 7: Glossary (Sheet 2 of 6)**

| Term                                                                               | Definition                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
|------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Direct Server Return (DSR)                                                         | Direct Server Return is a configuration where incoming client packets are sent to the Layer 4 Switch, but outbound target blade packets are sent directly back to the client. This reduces the outgoing traffic channeled through a load balancer by allowing Web servers to send their HTTP responses directly back to the requesting client without passing back through the load balancer. Enable this option on the Web I/O Accelerator if the target Web servers are configured to use DSR.      |
| DNS Domain                                                                         | Also known as the Domain Suffix; this will be used to resolve unqualified host names.                                                                                                                                                                                                                                                                                                                                                                                                                 |
| DNS Nameserver                                                                     | The IP address of the primary name server for the Web I/O Accelerator. This is the machine the Web I/O Accelerator queries to resolve host names into IP addresses.                                                                                                                                                                                                                                                                                                                                   |
| Ethernet 0 (ether0), Ethernet 1 (ether1), Ethernet 2 (ether2), Ethernet 3 (ether3) | Ethernet interfaces 0–3. 1U models of the DX appliance have only two Ethernet interfaces (ether0 and ether1). 2U models have four interfaces.                                                                                                                                                                                                                                                                                                                                                         |
| Farm                                                                               | A farm (also called a Server Farm) is a larger collection Web servers that are configured to serve either a single a several Web sites. Within the farm, the servers are frequently configured in clusters, each serving a single Web site.                                                                                                                                                                                                                                                           |
| Failover                                                                           | A process where two or more DXs monitor each other's health, and if one DX appliance fails, another one takes over the processing of new requests. This specifies whether or not the Web I/O Accelerator should act as a cold-standby fail-over unit for another Web I/O Accelerator on the network. NOTE: Both the active and the stand-by DXs should have this option enabled, and both units should have the same Virtual IP settings                                                              |
| Forwarder                                                                          | A Forwarder is a mechanism for forwarding traffic on to a set of servers. It listens for incoming traffic on a specific virtual IP address and port and distributes it over the target hosts. Unlike a cluster, a forwarder blindly forwards incoming traffic on to its target hosts. These typically are not Web servers, and the forwarder does not attempt to accelerate the outgoing traffic. This is for non-HTTP traffic; the forwarder simply passes the traffic through without examining it. |
| Group                                                                              | A group is a homogenous collection of Juniper DXs (also known as "blades"), that is being serviced by a Layer 4 Switch. Any of the DX appliances is capable of servicing a request. Within the Layer 4 Switch, the concept of a group is similar to the concept of a cluster that exists within the DX.                                                                                                                                                                                               |
| Hostname                                                                           | The fully qualified DNS name for the Web I/O Accelerator.                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Instant Redirect                                                                   | Instant Redirect is a mechanism where the DX monitors the health of the target hosts in a cluster, and diverts traffic from a cluster where all target hosts are down (i.e., a "dead" cluster) to an active cluster somewhere else in the network (world).                                                                                                                                                                                                                                            |
| Keyfile                                                                            | Key file for SSL traffic.                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Keypass                                                                            | Password for SSL key.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |

**Table 7: Glossary (Sheet 3 of 6)**

| Term                                  | Definition                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                 |
|---------------------------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Layer 4 Switch                        | A Layer 4 Switch (L4S) is a packet-based switch based on the OSI "transport" layer. Layer 4 switches identify which application protocols (HTTP, SMTP, FTP, and so forth) are included with each packet and use this information to hand off the packet to the appropriate blade or cluster. Layer 4 switches alleviate server load by balancing traffic across a group of DX appliances (blades) or a cluster of servers based upon individual session information and status. When an L4S is placed in front of cluster of servers running a particular application, and a client makes a request for that application, the switch determines which server should handle the request, often based upon current server loads. Once the forwarding decision is made, the switch binds that session to a particular server. |
| Layer 7 Health Checking               | Checks whether the target hosts are available by periodically sending an HTTP request to a specific URL on the target hosts.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Layer 7 Health Check Request Interval | The number of seconds separating each health check request sent to the target hosts. The valid range of values is 1 - 60 seconds.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Layer 7 Health Check Request URL Path | The URL path that is requested on a target host with each health check. The URL path must begin with a slash "/".                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                          |
| Layer 7 Health Check Retry Threshold  | The number of times a health check must fail before the target host is considered unavailable. The valid range of values is 1 - 20.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Layer 7 Health Check Resume Threshold | The number of times a health check must succeed before the target host is considered available. The valid range of values is 1 - 20.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Layer 7 Health Check Status Code      | The HTTP response status code expected from a target host in response to a health check. For typical use, the status code should be set to 200.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                            |
| Layer 7 Health Check Page Size        | The page size expected from a target host in response to a health check. This is the number of bytes in the body of the HTTP response, as it would be indicated in an HTTP Content-Length header. This is an optional setting; to disable this setting, use the value -1.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Layer 7 Health Check Expect String    | A string expected to appear somewhere in the HTTP response given to a health check. The expect string is searched for in the non-header portion of the HTTP response. It is case-sensitive and must be enclosed in double-quotes if there is whitespace in the string. The maximum length of the string is 64 bytes. This setting only applies to health check responses with the following MIME types: text/html, text/css, text/plain and text/xml. This is an optional setting.                                                                                                                                                                                                                                                                                                                                         |
| Listen Port                           | The port on which the Web I/O Accelerator listens for incoming Web traffic; it is typically set to 80.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Listen IP Address                     | Refer to Virtual IP Address.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Listen IP Netmask                     | Refer to Virtual IP Netmask.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Log Host                              | The IP address of the server to which the Web I/O Accelerator will be sending logging data.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Logging                               | Turns logging on or off. Remember that logging always exacts a performance penalty.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| MAC Address                           | The Media Access Controller (MAC) address is a hardware address that uniquely identifies each node of a network. This address is represented in the form of six hexadecimal numbers, typically separated with colons (For example: 20:4A:3E:44:00:22). This should not be confused with the IP Address.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                    |

**Table 7: Glossary (Sheet 4 of 6)**

| Term                  | Definition                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
|-----------------------|----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| Media                 | Media is the mode in which an Ethernet interface (Ether 0 and Ether 1) operates.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                         |
| MTU                   | Maximum Transmission Unit (MTU) is the largest number of bytes of “payload” data a frame can carry, not counting the frame's header and trailer. The MTU should be set to 1500 for Ethernet. DO NOT change this value unless your switch and network are configured to work with a different MTU.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Netmask               | A mask to filter out addresses that should not access the device.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| NTP                   | Network Time Protocol. Specifies whether or not the Web I/O Accelerator should listen for your NTP server.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| RADIUS                | Remote Authentication Dial In User Service                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |
| Redirector            | A Redirector is mechanism for redirecting requests to a single Web server. It listens for incoming Web requests on a specific virtual IP address and port and redirects the client to that Web server. Unlike a cluster, a redirector does not allow Web traffic to pass through the Web I/O Accelerator. Instead, for every Web request a redirector receives, the redirector sends the client back a redirect URL and forces it to resend its HTTP request to that URL.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Redirector Host       | The host portion of the redirect URL sent by the redirector. That is, this is the Web server to which the client should be redirected. The redirector host may be specified as either a hostname or an IP address.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                       |
| Redirector Port       | The port portion of the redirect URL sent by the redirector.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Redirector Protocol   | The protocol portion of the redirect URL sent by the redirector. Valid values are HTTP and HTTPS.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| Redirector URL Method | <p>The manner by which the redirector specifies the path portion of the redirect URL. If the request method is selected, then the redirector will construct the redirect URL using the same URL path as the original request. If the custom method is selected, then the redirector will construct the redirect URL using a custom URL path. You must specify a custom URL path if the custom method is selected, and the custom URL path must begin with a slash '/'.<br/><br/>For instance, if the request method is selected and the redirector receives a request for a page at '/path/page.html', then the redirect URL will look something like 'http://my.redirect.host/path/page.html'. However, if the custom method is selected and the custom URL path is set to '/custom/script.cgi?a = b', then the redirect URL will look something like 'http://my.redirect.host/custom/script.cgi?a = b' for any request received by the redirector.</p> |
| RMMP                  | The Redundancy Multicast Messaging Protocol (RMMP) is a mechanism where the active Layer 4 Switch sends health messages that the other Layer 4 Switch receives. This messaging protocol enables health checking between DX appliances. If a certain number of health messages are not received within a time window, the second Layer 4 Switch takes over the processing of new requests.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| Route (Default)       | Also known as the “Gateway”. This is the IP address of the machine the Web I/O Accelerator talks with in order to access the outside world.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                              |
| Server                | Web I/O Accelerator service.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                             |
| Server Load Balancer  | A Server Load balancer (SLB) distributes service requests across a group of target hosts, based on their availability to service requests.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                               |



**Table 7: Glossary (Sheet 5 of 6)**

| <b>Term</b>                    | <b>Definition</b>                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
|--------------------------------|---------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| SSL                            | Secure Sockets Layer (SSL) is a protocol that defines a way for two network devices to communicate securely. You can enable SSL on the listen side to communicate with clients securely. You can enable SSL on the target side to communicate with the target hosts securely                                                                                                                                                                                                                                                          |
| SSL Protocol Version           | There are three versions of SSL protocol: SSL version 1 (SSLv1), SSL version 2 (SSLv2) and Transport Layer Security version 1 (TLSv1). There are four SSL protocol modes in which the Web I/O Accelerator can operate:<br>sslv2: Use SSLv2 only<br>sslv3: Use SSLv3 only<br>sslv23: Use SSLv2, SSLv3 and TLSv1<br>tslv1: Use TLSv1 only                                                                                                                                                                                               |
| SSL Ciphersuite                | A collection of cryptographic algorithms used by two network devices to authenticate one another, transmit certificates and establish session keys. There are four categories of cipher suites used by the DX:<br>all: Allow all supported SSL ciphersuites<br>common: Allow only the fastest ciphersuites from both the strong and export groups<br>export: Allow only the low security ciphersuites suitable for export<br>strong: Allow only the highest security ciphersuites suitable for use in the U.S.A.                      |
| SSL Certfile                   | The certificate file used when establishing SSL communication.                                                                                                                                                                                                                                                                                                                                                                                                                                                                        |
| SSL Keyfile                    | The key file used when establishing SSL communication.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                |
| SSL Keypass                    | The password for the SSL Keyfile.                                                                                                                                                                                                                                                                                                                                                                                                                                                                                                     |
| Sticky                         | Ties a client to a server via the cookie or the client's IP address.                                                                                                                                                                                                                                                                                                                                                                                                                                                                  |
| Sticky Load Balancing          | A method of load balancing that binds a client to a server via a cookie or the client's IP address. It ensures that all subsequent requests made by a client are directed to the same server that handled the initial request.                                                                                                                                                                                                                                                                                                        |
| Target Host:Port               | This is the IP address and accompanying port of the Web server that the Web I/O Accelerator will accelerate. Depending upon the Web I/O Accelerator model, you may be able to enter IP addresses and ports for up to eight Target Hosts.                                                                                                                                                                                                                                                                                              |
| Target Name                    | This is the fully-qualified host name which clients use to reach your Web site or the servers you are accelerating.                                                                                                                                                                                                                                                                                                                                                                                                                   |
| Web I/O Accelerator Statistics | The following Web I/O Accelerator Statistics are available:<br>Uptime: The elapsed time since the Web I/O Accelerator was turned on.<br>Sessions (active/total): The number of TCP sessions that the Web I/O Accelerator has handled.<br>Requests (active/total): The number of HTTP requests the Web I/O Accelerator has received.<br>Bytes (in/out): The total amount, in bytes, of data the Web I/O Accelerator has received from target hosts, and the total amount of data that the Web I/O Accelerator has sent out to clients. |
| Virtual IP Address             | This is the IP address to which all incoming Web traffic should be routed. It should be different from the IP address(es) you specified on the Network Settings page.                                                                                                                                                                                                                                                                                                                                                                 |
| Virtual IP Netmask             | The proper subnet mask for a device with the given Virtual IP Address.                                                                                                                                                                                                                                                                                                                                                                                                                                                                |

**Table 7: Glossary (Sheet 6 of 6)**

| Term         | Definition                                                                                                                                                                                                            |
|--------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| VMAC Address | A Virtual MAC address is an address that is assigned by software to override the actual MAC address.                                                                                                                  |
| WebUI Port   | This is the port on which the administration Web server (WebUI) listens. For example, if you set this to 8090, you can connect to the DX by typing something like <code>http://junipername.yourdomain.com:8090</code> |
| WebUI SSL    | Turn SSL on or off for the administration Web server (WebUI). The first time, this must be performed in the CDXSHELL, and you will be prompted to generate a certificate.                                             |

:

| Term                           | Description                                                                                                                                                                             |
|--------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| GSLB master                    | DX that acts as a DNS name server to reorder DNS responses based on the selected load balancing algorithm. Collects performance metrics from other GSLB nodes. May also be a GSLB node. |
| GSLB node                      | Provides performance metrics to the GSLB master, including the Round Trip Time (RTT) to the local DNS server.                                                                           |
| GSLB agent                     | Runs on every DX that collects GSLB performance metrics.                                                                                                                                |
| GSLB group                     | DNS hostname that can resolve to several IP addresses. The returned IP address depends on the selected load-balancing policy.                                                           |
| GSLB resolver                  | Answers DNS requests received by the GSLB master. Can be configured to host DNS records or pass non-loadbalanced requests to another DNS server in the network.                         |
| Local DNS (LDNS)               | A client's master DNS server or its immediate upstream proxy.                                                                                                                           |
| Target DNS                     | DNS server where non-loadbalanced requests are forwarded. May be a standard DNS server in the network, or an internal DNS server on the GSLB master.                                    |
| Metric-based load balancing    | Load balancing based on the current DX performance metrics, including load, network bandwidth, and availability.                                                                        |
| Proximity-based load balancing | Metric-based load balancing using the lowest RTT measured between each DX and the client's LDNS.                                                                                        |

## Appendix B

# List of Events

### EMERG Events

---

- “DX Server was started”
- “Not licensed for this device”

**Table 8: EMERG Events Messages**

| Message                            | Description                                                       |
|------------------------------------|-------------------------------------------------------------------|
| “ntp daemon was started”           | The NTP process was started.                                      |
| “admin server was started”         | The WebUI was started                                             |
| “ssh daemon was started”           | The SSH server was started                                        |
| “telnet daemon was started”        | The telnet process was started.                                   |
| “snmp daemon was started”          | The SNMP process was started.                                     |
| “DX Server was started”            | DX was started.                                                   |
| “Not licensed for this device”     | The pac file is not licensed for this DX.                         |
| “DX Server was started”            | DX was started.                                                   |
| “Warning: License key file failed” | Warning message to indicate that the license key file is missing. |

### ALERT Events

---

**Table 9: ALERT Events Messages**

| Message                                                                                               | Description                                                                                                                                                                 |
|-------------------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------------------------------------------------|
| “admin password changed”                                                                              | The password for the Administrator was changed.                                                                                                                             |
| “Bad HTTP request: client sent an invalid header line: <http_header_line > ”                          | An HTTP request with and invalid head was received.                                                                                                                         |
| “Bad HTTP request: HEAD/0.9”                                                                          | HEAD request cannot be Version HTTP 0.9.                                                                                                                                    |
| “Bad HTTP request: header line longer than allowed or poorly formed”                                  | An HTTP request with a header line longer than allowed or a poorly formed HTTP request was received.                                                                        |
| “Bad HTTP request: POST length is less than zero. Request line: <POST request_line > ”                | An HTTP request with the method POST that has a length less than zero was received.                                                                                         |
| “Bad HTTP request: POST request did not contain content length. Request line: <POST request_line”     | An HTTP request with the method POST that did not contain the content length was received.                                                                                  |
| “Bad HTTP request: POST request specified content length of zero and is not configured to allow this” | An HTTP request with the method POST that specified the content length to be zero was received, but the DX appliance was not configured to allow zero length POST requests. |

**Table 9: ALERT Events Messages (continued)**

| Message                                                                                                                                                     | Description                                                                                                 |
|-------------------------------------------------------------------------------------------------------------------------------------------------------------|-------------------------------------------------------------------------------------------------------------|
| “Bad or missing private key file<br><keypath > ; password not set”                                                                                          | Invalid or missing private key file.                                                                        |
| “Cannot contact Default Gateway<br><gateway > ”                                                                                                             | Cannot ping the gateway.                                                                                    |
| “Cannot contact DNS server<br><dns_server > ”                                                                                                               | Unable to contact the DNS server.                                                                           |
| “Cannot contact E-mail server<br><email_server > ”                                                                                                          | Unable to contact the E-mail server.                                                                        |
| “Cannot contact NTP server<br><ntp_server > ”                                                                                                               | Unable to contact the NTP server.                                                                           |
| “Cannot contact syslog host<br><syslog_host > ”                                                                                                             | Unable to contact the syslog host.                                                                          |
| “Cannot contact Target Server<br><target_server > ”                                                                                                         | Unable to contact the Target server.                                                                        |
| “Cannot contact TFTP server<br><tftp_server > ”                                                                                                             | Unable to contact the TFTP server.                                                                          |
| “Cannot upgrade: archive is<br>< number_of_bytes > Kilo bytes.<br>Flash has < number of bytes ><br>available”                                               | Insufficient space on the Flash to perform the upgrade.                                                     |
| “Cluster not in operation; there is no<br>VIP present”                                                                                                      | The cluster is missing the Virtual IP address.                                                              |
| “Duplicate entry found in the CRL file<br>< crl_file > ”                                                                                                    | Duplicate entries were found in the CRL file.                                                               |
| “DX received excessive bytes from a<br>target < target_server > for request<br>< url_requested > ”                                                          | The DX received more bytes from a target server than is<br>indicated in the HTTP header.                    |
| “Failed to add CA cert to trusted list:<br>< internal error message > ”                                                                                     | Unable to add the CA Certificate to the CA Trusted List.                                                    |
| “Failed to load cacrlfile<br>< ca-crl_file > ; check file format”                                                                                           | Unable to load the CA CRL file. The CA CRL file must be<br>in a base64-encoded format.                      |
| “Failed to add CRL from cacrlfile<br>< ca_crl_file > ”                                                                                                      | Unable to add the CRL to the CA CRL file.                                                                   |
| “Failed to load the complete config”                                                                                                                        | Failed to load the configuration.                                                                           |
| “Illegal Content-Length header of<br>< length > sent from<br>< target_server > for a request<br>< url_requested > ”                                         | Invalid content length sent from the Target server.                                                         |
| “Illegal replay from < target_server ><br>(HTTP < http version > ) for a request<br>< url_requested > (no<br>Content-length/chunking/connection:<br>Close)” | Target server is HTTP1 and does not specify “connection:<br>close” or “content length” or does not chunk.   |
| “Illegal reply from < target_server ><br>(HTTP < http version > ) for a request<br>< url_requested > (no<br>Content-length/keep-alive set)”                 | The HTTP 1.0 Target server wants to do “keep-alive” but<br>not without setting the “content-length” header. |
| “< IP address > transitioning to<br>active                                                                                                                  | The DX has transitioned from a standby role to active<br>role.                                              |

**Table 9: ALERT Events Messages (continued)**

| Message                                                                                       | Description                                                                                                                       |
|-----------------------------------------------------------------------------------------------|-----------------------------------------------------------------------------------------------------------------------------------|
| “Layer 2 Link Down on Main Interface”                                                         | The link was down on the main network interface, ether0.                                                                          |
| “No client authentication CA certfile specified”                                              | Missing CA Certificate file. CA Certificate file specifies the list of acceptable CA Certificates that a client may connect with. |
| “No clusters are in operation due to < configuration > errors”                                | All clusters are disabled.                                                                                                        |
| “Only < number > of clusters out of < number > in operation”                                  | Not all clusters are enabled.                                                                                                     |
| “Rebooted from CLI”                                                                           | The DX was rebooted; initiated from the CLI.                                                                                      |
| “Target server < target_server > disabled through configuration”                              | Target server was disabled through the CLI or Web User interface.                                                                 |
| “Target server < target_server > has been contacted”                                          | Successfully established a TCP connection the Target server.                                                                      |
| “Target server < target_server > passed Layer 7 health check”                                 | Target server passed the Layer 7 health check performed by the DX.                                                                |
| “The administrator password has been changed by pressing the reset button”                    | The reset button was pressed and thus the default administrator password was reset.                                               |
| “The CA Trust file < ca_trust_file > could not be loaded; check file format”                  | Unable to load the CA Trust file. The CA Trust file must be in a base64-encoded format.                                           |
| “The CA Certificate file < ca_cert_file > failed to load; check file format.”                 | Unable to load the CA Certificate file. The CA Certificate file must be in a base64-encoded format.                               |
| “Threshold for the maximum number of connections exceeded”                                    | The DX has reached the threshold configured for the maximum number of connections.                                                |
| “Received excessive bytes from the target < target_server > for a request < url_requested > ” | Target server sent more bytes than what are specified in the “content-length” header.                                             |
| “Rebooted from the CLI”                                                                       | Accelerator was rebooted from the CLI.                                                                                            |
| “VIP < vip > down”                                                                            | The VIP is down because all Target servers are down.                                                                              |
| “VIP < vip > up”                                                                              | The VIP is up.                                                                                                                    |



## Appendix C

# Configuring Failover by Service

This appendix describes how to configure failover for each of the support services. The configuration method described is for use only with the DX Application and Acceleration Platform version 5.0 or earlier.



If you are upgrading to or have just purchased version 5.1, you should instead migrate to or use the new failover method described in Chapter 20, “Configuring Failover”.

This appendix contains the following topics:

- “Configuring Server Failover for the Cluster, Forwarder, and Redirector Services” on page 438
- “Configuring Failover for the ActiveN Service” on page 441



SLB failover can only be achieved using the new failover method.

## Configuring Failover for the SLB Service

You can configure failover for the SLB service with or without a VMAC.

To enable basic failover:

```
dx% set slb failover enable
```



**NOTE:** A warning message appears when you enable SLB failover if the VIP for an activeN group conflicts with the VIP for an SLB group.

```
WARNING: The following vip conflicts were detected!!  
ActiveN group <name> vip <vip> conflicts with SLB group <name>
```

We recommend resolving these conflicts to avoid potential IP conflicts on your network.

If you see this message, verify your activeN group and SLB group VIPs and reconfigure accordingly.

To specify a VMAC to use during SLB failover:

```
dx% set slb failover vmac <enabled|disabled>
dx% set slb failover vmac id <id>
```

## Configuring Server Failover for the Cluster, Forwarder, and Redirector Services

To configure a failover unit, you must enable a “Failover” mode on two DXs configured with the same VIP address. The DX automatically determines which unit is active and which is the failover based on the order in which failover is enabled on the two units. The DX that has failover enabled first becomes the primary unit. The DX with failover enabled second monitors the first DX. If the active DX goes offline, the failover DX activates and takes its place.

1. Ether 1 has a default IP address configured and you can optionally configure with a different IP.
2. OPTIONAL: Because failover uses multicast, the factory default IP and netmask should work without any changes, but you can provide valid IP and Netmask settings for the DX’s Ethernet interfaces. This can be done from either the DXSHELL command line or through a Web browser with the WebUI.

- a. From the DXSHELL:

You can view interface settings from the DXSHELL with the command:

```
dx% show ether 1
```

You can change interface settings from the DXSHELL with the commands:

```
dx% set ether 1 ip <address>
dx% set ether 1 netmask <netmask>
```

where `< address >` and `< netmask >` are the IP Address or Netmask values you wish to enter.

To save changes, use the command:

```
dx% write
```

- b. From the WebUI:

To view and change interface settings for Ether 1 in the WebUI, open the Network Settings page. Be sure to click the SAVE button to save any changes.



3. Make sure that the primary DX and the failover DX have the same Virtual IP address.
  - a. From the DXSHELL, you can view VIP settings from the DXSHELL with the command:

```
dx% show cluster <name> listen vip
```

- b. From the WebUI, you can view VIP settings in the WebUI on the DX Settings page.
4. OPTIONAL: Enable failover with Virtual MAC (VMAC)

With this option enabled, the active unit will use the Virtual MAC as its MAC address. The VMAC ID determines the virtual MAC address. The default for the VMAC ID is zero. When a standby unit becomes active, it will assume the VMAC as its MAC address.

- a. Enable failover with the VMAC

```
dx% set server failover vmac enabled
```

- b. Configure the VMAC ID

```
dx% set server failover vmac id <VMAC ID>
```

5. The active DX will failover in the event of an Ethernet link failure.

When the Ethernet link is determined to be down, the amount of time before the active DX will shut itself down is the `POLL INTERVAL * COUNT`. This value should be less than three seconds.

Poll interval is a value in milliseconds. The default poll interval is 500 milliseconds. The count is an integer, and the default count is 4.

6. Enable failover on BOTH the primary DX and the failover DX.

- a. From the DXSHELL:

You can enable failover from the DXSHELL with the command:

```
dx% set server failover enabled
```

- b. From the WebUI:

To enable failover, open the Administrator Services page and locate the “High Availability Failover” option and check “Enabled”. Be sure to press the SAVE button to save and apply the changes.

7. To activate failover from the DXSHELL, use the following command on both servers:

```
dx% set server up
```

Determination of which server is the active server and which is the backup is negotiated by the DXs, based upon the network address (the unit with the higher network address becomes the active unit).

### ***Initiating a Manual Server Failover***

There are times when you need to take a server off-line for maintenance or debugging purposes. You can initiate a manual failover in an active-standby configuration on the active server by typing the command:

```
dx% set server down
```

When the backup server fails to detect the heartbeat messages coming from the active server, it takes over processing and becomes the active node.

Either the server that you took off-line or a replacement unit can be returned to activity as a backup unit by typing the command:

```
dx% set server up
```

For example, if you modify a configuration on a cluster on an active DX that requires a restart of the multiplexing engine, the process brings the active DX down, and the standby DX takes over and becomes the active DX. This may result in a Web site not processing requests while the standby DX appliance takes over.

If you want to make configuration changes to an active-standby configuration without affecting request processing, use the following sequence:

1. Ensure that DX 1 and DX 2 are in an active-standby configuration (DX 1 is active and DX 2 is the standby).
2. Change the cluster configuration on DX 2 (passive).
3. Move the traffic to DX 2 (set the server down on DX 1).
4. Check the failover status on DX 2 (now active).
5. Check the ActiveN status on DX 2 (now active).
6. Bring DX 1 up (set the server up on DX 1).
7. Check the failover status on DX 1 (now passive).
8. Check the ActiveN status on DX 2 (now standby).
9. Change the cluster configuration on DX 1.

## Configuring Failover for the ActiveN Service

---

ActiveN failover must be enabled on all of the DX appliances configured with the same VIP address, in the ActiveN group. The DX appliances automatically determine which unit is active and which is the failover based upon the order in which failover is enabled on the units. The DX that has failover enabled first becomes the primary unit. The DX with failover enabled second, third, and so on, monitors the first DX. If the active DX goes offline, the next DX appliance is activated and takes its place.

To configure ActiveN failover using the DXSHELL:

```
dx% set activeN failover <enabled|disabled>
```

---



**NOTE:** A warning message appears when you enable activeN failover if the VIP for an activeN group conflicts with the VIP for an SLB group.

```
WARNING: The following vip conflicts were detected!!  
ActiveN group <name> vip <vip> conflicts with SLB group <name>
```

It is advised to resolve the conflicts as there could be IP conflicts on your network!!

If you see this message, verify your activeN group and SLB group VIPs and reconfigure accordingly.

---

The following additional commands are available to configure various attributes of the ActiveN service.

This command is used to enable or disable the “Forcemaster”. Enabling the forcemaster allows a switch to snatch “activeness” from another switch with a higher node ID.

```
dx% set activeN failover forcemaster <enabled|disabled>
```

This command is used to set the multicast address for the failover mechanism. The default address is 239.0.0.2.

```
dx% set activeN failover mcastaddr <Ip addr>
```

This command is used to set the bind address for the failover mechanism.

```
dx% set activeN failover bindaddr <Ip addr>
```

This command is used to set the node ID of the ActiveN failover unit. Setting the node ID to auto will result in the node ID being generated automatically.

```
dx% set activeN failover nodeid <number|auto>
```

This command is used to set the port for failover communication.

```
dx% set activeN failover port peer <port>
```

This command is used to disable or enable the Virtual MAC (default is disabled).

```
dx% set activeN failover port vmac [disabled | enabled]
```

This command is used to assign the Virtual MAC Address to the specified ID.

```
dx% set activeN failover port vmac <id>
```

# Index

## Numerics

|                  |     |
|------------------|-----|
| 1U Chassis ..... | 16  |
| 2U Chassis ..... | 16  |
| 3G Cache .....   | 289 |

## A

|                                                |          |
|------------------------------------------------|----------|
| AAA. See Authentication                        |          |
| Action Statements                              |          |
| AppRule .....                                  | 304      |
| Active Directory .....                         | 261      |
| Active-Active .....                            | 49       |
| Configuring .....                              | 239      |
| Defined .....                                  | 427      |
| Topology .....                                 | 50       |
| ActiveN                                        |          |
| Commands .....                                 | 244, 359 |
| Configuring .....                              | 239      |
| Defined .....                                  | 427      |
| Group Naming Conventions .....                 | 152      |
| Operation .....                                | 55       |
| Sample Configuration .....                     | 243      |
| Statistics .....                               | 249      |
| Topology .....                                 | 51       |
| Active-Standby .....                           | 49       |
| Defined .....                                  | 427      |
| Topology .....                                 | 50       |
| Adding a New User .....                        | 114      |
| Admin Audit Trail .....                        | 128      |
| Administration Rights .....                    | 26       |
| Administrator Rights                           |          |
| User Access Levels .....                       | 26       |
| Advanced Statistics .....                      | 393      |
| ALERT .....                                    | 130      |
| Alert                                          |          |
| Level, Setting .....                           | 130      |
| Apache .....                                   | 374, 406 |
| Configuring Logging .....                      | 274      |
| Importing Existing Keys and Certificates ..... | 197      |
| Append .....                                   | 323      |
| Appliance                                      |          |
| Upgrading .....                                | 146      |
| AppRule                                        |          |
| Action Statements .....                        | 304      |
| Arguments, Arguments                           |          |
| AppRule .....                                  | 303      |

|                                                   |         |
|---------------------------------------------------|---------|
| Cache .....                                       | 293     |
| Header Variables .....                            | 300     |
| Limitations and Implications .....                | 321     |
| Logging .....                                     | 325     |
| Operators .....                                   | 302     |
| Page Translator .....                             | 314     |
| Page Translator (Content) .....                   | 316     |
| PAR Test Operators .....                          | 323     |
| Prepend, Append, Replace (PAR) Conditions .....   | 323     |
| Relationships .....                               | 310     |
| Request Retry Examples .....                      | 331     |
| Request Routing Examples .....                    | 331     |
| Request Sentry .....                              | 310     |
| Request Sentry Examples .....                     | 329     |
| Request Translator .....                          | 312     |
| Request Translator Examples .....                 | 330     |
| Scenarios .....                                   | 325     |
| Variables .....                                   | 299     |
| Assigning Roles to a User .....                   | 115     |
| Audit Trail                                       |         |
| Admin .....                                       | 128     |
| Authentication .....                              | 39, 253 |
| Cache .....                                       | 41      |
| Collecting the Authentication Data .....          | 40      |
| Commands .....                                    | 253     |
| Methods .....                                     | 41      |
| Forward Client Certificate .....                  | 43      |
| LDAP .....                                        | 42      |
| RADIUS .....                                      | 41      |
| Password Change Request .....                     | 45      |
| Authentication, Authorization, and Auditing ..... | 40      |

## B

|                       |     |
|-----------------------|-----|
| Backup Chaining ..... | 33  |
| Bind Address .....    | 50  |
| Blade                 |     |
| Defined .....         | 427 |
| Busy Redirect         |     |
| Defined .....         | 427 |

## C

|                          |     |
|--------------------------|-----|
| Cache .....              | 289 |
| AppRules .....           | 293 |
| Load Balancing .....     | 62  |
| Naming Conventions ..... | 152 |

|                                                         |          |                                                   |          |
|---------------------------------------------------------|----------|---------------------------------------------------|----------|
| Persistence.....                                        | 61       | Configuration Management.....                     | 131      |
| Statistics.....                                         | 62       | Backup.....                                       | 131      |
| Transparency.....                                       | 62       | Configuration Synchronization.....                | 138      |
| Usage Scenarios.....                                    | 61       | Editing a Configuration.....                      | 133      |
| Capacity Planning.....                                  | 383      | Exporting a Configuration.....                    | 131      |
| Certificate Authority                                   |          | Exporting and Importing a Configuration.....      | 131      |
| Certificate Presentation.....                           | 214      | Importing a Configuration.....                    | 132      |
| Trusted Certificate Authority Certificate Storage ..... | 214      | Restoring the Factory Default Configuration ..... | 135      |
| Certificate Revocation List.....                        | 215      | Synchronization Group.....                        | 141      |
| Client IP Sticky.....                                   | 59, 248  | System Snapshot.....                              | 135      |
| Client IP Transparency.....                             | 155      | View the Contents of a Configuration File .....   | 133      |
| Cluster                                                 |          | Configuration Questions.....                      | 103      |
| Defined.....                                            | 427      | Answering.....                                    | 103      |
| Naming Conventions.....                                 | 152      | Configuration Synchronization .....               | 138      |
| Redirection .....                                       | 212      | Synchronization Group.....                        | 141      |
| Command Line Interface.....                             | 81       | Configuring the Login Banner.....                 | 144      |
| Commands                                                |          | Connecting a Terminal.....                        | 97       |
| 3G Cache Configuration.....                             | 289      | Connecting the appliance to Your Network.....     | 97       |
| ActiveN.....                                            | 244, 359 | Connection Binding.....                           | 161      |
| Authentication.....                                     | 253      | Configuring.....                                  | 161      |
| Cipherfile.....                                         | 210      | Layer 7 Health Checking.....                      | 162      |
| Client IP Transparency.....                             | 157      | NTLM Authentication Protocol .....                | 161      |
| Command Abbreviation.....                               | 85       | Connectivity Failover.....                        | 368, 369 |
| Configuration Synchronization.....                      | 140      | Console Port.....                                 | 97       |
| Disable Logging of Show Commands.....                   | 129      | Conventions used in this manual .....             | XVI      |
| Expect.....                                             | 232      | Convert302protocol                                |          |
| Export CSV Statistics.....                              | 392      | Defined.....                                      | 427      |
| Forward Proxy Accelerator.....                          | 285      | Cookie-based Client Stickiness .....              | 251      |
| HTTP(S) Authentication.....                             | 253      | CSV Export Statistics.....                        | 391      |
| Local IP Configuration.....                             | 169      | Customiplogheader                                 |          |
| Making Changes from the Command Line.....               | 83       | Defined.....                                      | 427      |
| OWA Configuration.....                                  | 377      | <b>D</b>                                          |          |
| Remote Authentication Configuration.....                | 122      | Default Route                                     |          |
| Reverse Route Return.....                               | 163      | Defined.....                                      | 427      |
| Scriptable Health Checking.....                         | 236      | Default route.....                                | 96       |
| SLB Statistics.....                                     | 185      | Degraded Performance.....                         | 414      |
| SNAT Configuration.....                                 | 158      | Deleting all Users.....                           | 120      |
| SOAP Server Management.....                             | 142      | Direct Server Return                              |          |
| SSL Client Authentication.....                          | 218      | Defined.....                                      | 428      |
| Synchronization Group Management.....                   | 141      | DNS                                               |          |
| Target Host Pause.....                                  | 168      | Troubleshooting.....                              | 416      |
| Target Server Compression .....                         | 171      | DNS Domain.....                                   | 96       |
| TCL.....                                                | 232      | Defined.....                                      | 428      |
| Technical Service Dump .....                            | 418      | DNS Nameserver                                    |          |
| Troubleshooting.....                                    | 413      | Defined.....                                      | 428      |
| Turning on the WebUI.....                               | 86       | DNS Proxy Filter.....                             | 68       |
| Viewing Server Statistics.....                          | 382      | DNS records.....                                  | 66       |
| VLAN.....                                               | 166      | DNS Server.....                                   | 346      |
| Common Administration Tasks.....                        | 113      | Deleting Domains.....                             | 348      |
| Compression                                             |          | Deleting Resource Records.....                    | 348      |
| Target Server.....                                      | 170      | Domain Name System. See DNS                       |          |
| Configuration                                           |          | Domino.....                                       | 373      |
| Preserving.....                                         | 147      | DSR                                               |          |

- Defined ..... 428
  - Dual Power Supply ..... 97
- E**
- EMERG ..... 130
  - Enabling a User ..... 114
  - Event
    - ALERT ..... 130, 433
    - EMERG ..... 130, 433
    - Logging ..... 129
    - Notification ..... 129
  - Event Logging ..... 129
  - Event, List ..... 433
  - External Server Load Balancer ..... 153
- F**
- Factory Default Configuration ..... 135
  - Failover ..... 60, 97
    - Connectivity ..... 369
    - Defined ..... 428
  - Fast Ethernet ..... 97
  - Firmware Upgrade ..... 26
  - First-time Configuration Screen ..... 102
  - Floating VIP ..... 160
  - Forward Client Certificate ..... 43
  - Forwarder
    - Defined ..... 428
    - Naming Conventions ..... 152
    - SSL ..... 6, 34
  - Fully-qualified Domain Name ..... 96
- G**
- Generating SSL Keys and Certificates ..... 206
  - Global Server Load Balancing. See GSLB
  - Group, SLB ..... 28
  - GSLB
    - DNS Proxy Filter ..... 68
    - Health-Checking ..... 70
    - Load-Balancing ..... 70
    - Statistics ..... 72
- H**
- Health
    - SLB Group ..... 29
  - Health Checking
    - Interval ..... 368
    - Layer 7 ..... 221
    - Scriptable ..... 47, 229
    - SMTP ..... 238
  - Heartbeat ..... 50
  - Historical Rates and Statistics ..... 383
    - Enabling ..... 386
  - HTTP Listen Statistics ..... 395
  - HTTP Target Host Statistics ..... 398
  - HTTP(S) Authentication ..... 39, 253
  - Hyper Terminal ..... 99
- I**
- I/O Listen Statistics ..... 393
  - I/O Physical Target Statistics ..... 395
  - I/O Target Host Statistics ..... 394
  - IIS
    - Configuring Logging ..... 275
  - Initiating a Manual Failover ..... 359, 367, 440
  - Install
    - Command ..... 148
  - Installation Overview ..... 95
  - Instant Redirect ..... 173
    - Defined ..... 428
  - Integer-only names ..... 153
  - Integrating the Appliance Into Your Network ..... 151
  - IP Address ..... 96
  - IP Transparency ..... 155
  - iPlanet
    - Configuring Logging ..... 280
- J**
- JDE OneWorld ..... 373
- L**
- L4S. See Layer 4 Switch
  - L7. See Layer 7 Health
  - Layer 4 Switch ..... 28, 30, 52
    - Concept ..... 52
    - Defined ..... 429
    - Health Checking ..... 56
    - Network Acceleration ..... 53
  - Layer 7 Health Checking ..... 221
    - Connection Binding ..... 162
    - Defined ..... 429
    - Logging System Log Messages ..... 225
    - Receive Notification of Errors ..... 131
    - Using SLB ..... 227
  - LDAP ..... 42, 254, 257, 261
  - Least Connection ..... 32
  - License Agreement ..... 103
  - License Key ..... 146
    - Installing ..... 128
  - Lightweight Directory Access Protocol. See LDAP
  - List of Events ..... 433
  - Load Balancing
    - Cache ..... 62
  - Load Balancing Policies ..... 30
    - Backup Chaining ..... 33
    - Least Connection ..... 32
    - Maximum Connections ..... 32
    - Round Robin ..... 31
    - Weighted Least Connections ..... 32

|                                                    |               |                                               |         |
|----------------------------------------------------|---------------|-----------------------------------------------|---------|
| Weighted Round-robin .....                         | 31            | Default .....                                 | 26, 102 |
| Location .....                                     | 128           | Default Administrator .....                   | 105     |
| Log Entries .....                                  | 128           | Lost .....                                    | 120     |
| Syntax .....                                       | 128           | Reset Button .....                            | 120     |
| Logging                                            |               | Password Change Request, Authentication ..... | 45      |
| AppRules .....                                     | 325           | Pausing a Target Host .....                   | 167     |
| Show Commands .....                                | 129           | PeopleSoft .....                              | 373     |
| with Apache .....                                  | 274           | Performance Monitoring .....                  | 381     |
| with IIS .....                                     | 275           | Round Robin Database Mechanism .....          | 384     |
| with iPlanet .....                                 | 280           | Statistical Data Items .....                  | 385     |
| with NetCache .....                                | 281           | Performance, Degraded .....                   | 414     |
| with Resin .....                                   | 280           | Policies                                      |         |
| Logging In the First Time .....                    | 102           | Load Balancing .....                          | 30      |
| Login Banner .....                                 | 144           | Powering-up the appliance .....               | 97      |
| Capturing .....                                    | 145           | preparation of content .....                  | 95      |
| Configuring .....                                  | 144           | Prepend .....                                 | 323     |
| Configuring from the Command Line Interface .....  | 144           | Preserving Your Configuration .....           | 147     |
| Customizing .....                                  | 144           | Primary Nameserver .....                      | 96      |
| Displaying in the WebUI .....                      | 145           | <b>R</b>                                      |         |
| Parsing HTML .....                                 | 146           | RADIUS .....                                  | 41, 254 |
| Lost Password .....                                | 120           | Rates and Statistics                          |         |
| <b>M</b>                                           |               | Historical .....                              | 383     |
| Managing Users .....                               | 113           | Redirection                                   |         |
| Manual Failover, Initiating .....                  | 359, 367, 440 | Cluster .....                                 | 212     |
| Maximum Connections .....                          | 32            | Redirector                                    |         |
| Monitoring                                         |               | Defined .....                                 | 430     |
| Performance .....                                  | 381           | Naming Conventions .....                      | 152     |
| <b>N</b>                                           |               | Relationships, AppRule .....                  | 310     |
| Naming Conventions .....                           | 152           | Replace .....                                 | 323     |
| NAT Operation                                      |               | Request Retry Examples .....                  | 331     |
| Full NAT .....                                     | 30            | Request Routing Examples .....                | 331     |
| Half Nat .....                                     | 30            | Request Sentry AppRule .....                  | 310     |
| Netcache                                           |               | Request Sentry Examples .....                 | 329     |
| Configuring Logging .....                          | 281           | Request Translator Apprule .....              | 312     |
| Netmask .....                                      | 96            | Request Translator Examples .....             | 330     |
| Network Activity Report .....                      | 419           | Requirements                                  |         |
| NULL .....                                         | 152           | Upgrade .....                                 | 147     |
| <b>O</b>                                           |               | Reset Button .....                            | 120     |
| Operators                                          |               | Resetting the Password .....                  | 120     |
| Apprule .....                                      | 302           | Resin                                         |         |
| Outlook Web Access                                 |               | Configuring Logging .....                     | 280     |
| OverDrive Application Rule Translator. See AppRule |               | Reverse Route Return .....                    | 162     |
| OWA Commands .....                                 | 377           | RMMP                                          |         |
| OWA. See Outlook Web Access                        |               | Defined .....                                 | 430     |
| <b>P</b>                                           |               | Role                                          |         |
| Package Contents .....                             | XVI           | administrator .....                           | 27      |
| Page Translator (Content) AppRule .....            | 316           | Default .....                                 | 26      |
| Page Translator AppRules .....                     | 314           | network_administrator .....                   | 27      |
| PAR Test Operators .....                           | 323           | network_operator .....                        | 27      |
| Password .....                                     | 96, 102       | security_administrator .....                  | 27      |
|                                                    |               | security_operator .....                       | 27      |
|                                                    |               | target_host_operator .....                    | 27      |
|                                                    |               | user .....                                    | 27      |



- Roles ..... 27
  - Assigning to a User ..... 115
- Round Robin ..... 31
- Round Robin Database Mechanism ..... 384
- RSA Secure ID ..... 261
  
- S**
- Sandbox Environment ..... 230
- SCP Server ..... 131
- Scriptable Health Checking ..... 47, 229
- Secure Socket Layer. See SSL
- SecureCRT ..... 97
- Serial Number ..... 128
- Server
  - Statistics ..... 382
- Server Load Balancer ..... 153, 179, 437
  - Defined ..... 430
  - Deploying Behind ..... 153
  - Failover ..... 60
- Server Statistics ..... 405
- Setting the Password for a New User ..... 114
- SLB ..... 153, 179, 437
  - Group ..... 28
  - Group Health ..... 29
  - Layer 7 Health Checking ..... 227
  - Port Symmetry ..... 29
- SMTP Health Checking ..... 238
- SNAT ..... 157
  - Operation ..... 158
- SOAP Server ..... 142
- Source Network Address Translation ..... 157
- SSL
  - Basic Conventions and Terms ..... 34
  - Certificates and Keys ..... 4, 33
  - Cipher Suite Details ..... 209
  - Cluster Redirection ..... 212
  - Configuration Examples ..... 190
  - Configuration Parameters ..... 37
  - Configuring Client Authentication ..... 213
  - Forwarder ..... 6, 34
  - Generating Keys and Certificates ..... 206
  - Importing Existing Keys and Certificates ..... 197
  - Listen Statistics ..... 401
  - Overview ..... 4, 33
  - Setting Up For ..... 33
  - Target Host Statistics ..... 401
- Statistics ..... 382
  - Advanced ..... 393
  - Cache ..... 62
  - CSV Export ..... 391
  - GSLB ..... 72
  - Historical ..... 383
  - HTTP Listen ..... 395
  - HTTP Target Host ..... 398
  - I/O Listen ..... 393
  - I/O Physical Target ..... 395
  - I/O Target Host ..... 394
  - Server ..... 405
  - SSL Listen ..... 401
  - SSL Target Host ..... 401
  - Statistics, ActiveN ..... 249
- Sticky ..... 431
  - Client-IP Based ..... 252
  - Cookie-Based ..... 251
  - Overview ..... 251
  - Traffic ..... 251
- Sticky Load Balancing
  - Defined ..... 431
- Synchronization Group ..... 141
- System Snapshot ..... 135
  
- T**
- Target Host
  - Pausing ..... 167
  - Using a Local IP for Communication ..... 169
- Target Hosts ..... 414
- Target Server
  - Enabling Compression ..... 170
- Target Server Compression ..... 170
- Target Server WebUI ..... 172
- Target Tuning ..... 373
  - AppRules ..... 7, 375
  - OWA Commands ..... 377
  - Tool ..... 373
  - WebDAV ..... 374
- Tcl Scripts ..... 229
- TCP Selective Acknowledgement ..... 164
- tcpdump ..... 419
- Technical Service Dump ..... 418
- Terminal ..... 99
  - Baud Rate ..... 99
  - Emulator ..... 99
  - Flow Control ..... 99
  - Settings ..... 99
- TFTP Server ..... 131
- Timestamp ..... 128
- Tool, Target Tuning ..... 373
- Transparency, Cache ..... 62
- Troubleshooting ..... 413
  - Commands ..... 413
- Tuning the Appliance ..... 373
  
- U**
- Upgrade ..... 26
  - Using the install Command ..... 148
- Upgrade Requirements ..... 147
- Upgrading the Appliance ..... 146
- User

- Adding ..... 114
- Assigning Roles ..... 115
- Enabling ..... 114
- Setting the Password ..... 114
- Username ..... 96, 102, 129
  - Default ..... 26, 102
- Users
  - Managing ..... 113

**V**

- Valid Passwords ..... 27
- Valid User Names ..... 27
- Variables
  - AppRule ..... 299
- VIP
  - Floating ..... 160
- Virtual IP Address
  - Defined ..... 431
- Virtual IP address ..... 13
- Virtual LAN
  - Configuring ..... 164

**W**

- Web Log ..... 405
  - Batch Mode ..... 408
  - Commands ..... 407
  - Configuration ..... 405
  - Field Definitions ..... 406
  - Format
    - Combined ..... 406
    - Combined\_cn ..... 406
    - Common ..... 406
    - Common\_cn ..... 406
    - Perf1 ..... 406
    - Perf2 ..... 406
- WEBDAV ..... 374
- WebUI
  - Troubleshooting ..... 417
- Weighted Least Connections ..... 32
- Weighted Round-robin ..... 31
- white space ..... 152
- Windows HyperTerminal ..... 97