

Media Flow

Media Flow Controller™

Administrator's Guide and CLI Command Reference

Release

2.0.7



Published: 2011-7-7

Juniper Networks, Junos, Steel-Belted Radius, NetScreen, and ScreenOS are registered trademarks of Juniper Networks, Inc. in the United States and other countries. The Juniper Networks Logo, the Junos logo, and JunosE are trademarks of Juniper Networks, Inc. All other trademarks, service marks, registered trademarks, or registered service marks are the property of their respective owners.

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 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.

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 (i) Juniper Networks, Inc. (if the Customer's principal office is located in the Americas) or Juniper Networks (Cayman) Limited (if the Customer's principal office is located outside the Americas) (such applicable entity being referred to herein as "Juniper"), and (ii) 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, for which Customer has paid the applicable license or support fees to Juniper or an authorized Juniper reseller, or which was embedded by Juniper in equipment which Customer purchased from Juniper or an authorized Juniper reseller. "Software" also includes updates, upgrades and new releases of such software. "Embedded Software" means Software which Juniper has embedded in or loaded onto the Juniper equipment and any updates, upgrades, additions or replacements which are subsequently embedded in or loaded onto the equipment.
3. **License Grant.** Subject to payment of the applicable fees and the limitations and restrictions set forth herein, Juniper grants to Customer a 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 Embedded Software solely as embedded in, and for execution on, Juniper equipment originally purchased by Customer from Juniper or an authorized Juniper reseller.
 - 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; provided, however, with respect to the Steel-Belted or Odyssey Access Client software only, Customer shall use such Software on a single computer containing a single physical random access memory space and containing any number of processors. Use of the Steel-Belted or IMS AAA software on multiple computers or virtual machines (e.g., Solaris zones) requires multiple licenses, regardless of whether such computers or virtualizations are physically contained on a single chassis.
 - c. Product purchase documents, paper or electronic user documentation, and/or the particular licenses purchased by Customer may specify limits to Customer's use of the Software. Such limits may restrict use to a maximum number of seats, registered endpoints, concurrent users, sessions, calls, connections, subscribers, clusters, nodes, realms, devices, links, ports or transactions, or require the purchase of separate licenses to use particular features, functionalities, services, applications, operations, or capabilities, or provide throughput, performance, configuration, bandwidth, interface, processing, temporal, or geographical limits. In addition, such limits may restrict the use of the Software to managing certain kinds of networks or require the Software to be used only in conjunction with other specific Software. Customer's use of the Software shall be subject to all such limitations and purchase of all applicable licenses.
 - d. For any trial copy of the Software, Customer's right to use the Software expires 30 days after download, installation or use of the Software. Customer may operate the Software after the 30-day trial period only if Customer pays for a license to do so. Customer may not extend or create an additional trial period by re-installing the Software after the 30-day trial period.
 - e. The Global Enterprise Edition of the Steel-Belted software may be used by Customer only to manage access to Customer's enterprise network. Specifically, service provider customers are expressly prohibited from using the Global Enterprise Edition of the Steel-Belted software to support any commercial network access services.

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, sell, 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 or any product in which the Software is embedded; (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, service, application, operation, or capability without first purchasing the applicable license(s) and obtaining a valid key from Juniper, even if such feature, function, service, application, operation, 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 Embedded Software on non-Juniper equipment; (j) use Embedded Software (or make it available for use) on Juniper equipment that the Customer did not originally purchase from Juniper or an authorized Juniper reseller; (k) disclose the results of testing or benchmarking of the Software to any third party without the prior written consent of Juniper; or (l) 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 for Customer's internal business purposes.
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.** The warranty applicable to the Software shall be as set forth in the warranty statement that accompanies the Software (the "Warranty Statement"). Nothing in this Agreement shall give rise to any obligation to support the Software. Support services may be purchased separately. Any such support shall be governed by a separate, written support services agreement. TO THE MAXIMUM EXTENT PERMITTED BY LAW, JUNIPER SHALL NOT BE LIABLE FOR ANY 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 IN THE WARRANTY STATEMENT TO THE EXTENT PERMITTED BY LAW, 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 NON-INFRINGEMENT. IN NO EVENT DOES JUNIPER WARRANT THAT THE SOFTWARE, OR ANY EQUIPMENT OR NETWORK RUNNING THE SOFTWARE, WILL OPERATE WITHOUT ERROR OR INTERRUPTION, OR WILL BE FREE OF VULNERABILITY TO INTRUSION OR ATTACK. In no event shall Juniper's or its suppliers' or licensors' liability to Customer, whether in contract, tort (including negligence), breach of warranty, or otherwise, exceed the price paid by Customer for the Software that gave rise to the claim, or if the Software is embedded in another Juniper product, the price paid by Customer for such other product. Customer acknowledges and agrees that Juniper has set its prices and entered into this Agreement in reliance upon the disclaimers of warranty and the limitations of liability set forth herein, that the same reflect an allocation of risk between the Parties (including the risk that a contract remedy may fail of its essential purpose and cause consequential loss), and that the same form an essential basis of the bargain between the Parties.

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 payable under this agreement are exclusive of tax. Customer shall be responsible for paying Taxes arising from the purchase of the license, or importation or use of the Software. If applicable, valid exemption documentation for each taxing jurisdiction shall be provided to Juniper prior to invoicing, and Customer shall promptly notify Juniper if their exemption is revoked or modified. All payments made by Customer shall be net of any applicable withholding tax. Customer will provide reasonable assistance to Juniper in connection with such withholding taxes by promptly: providing Juniper with valid tax receipts and other required documentation showing Customer's payment of any withholding taxes; completing appropriate applications that would reduce the amount of withholding tax to be paid; and notifying and assisting Juniper in any audit or tax proceeding related to transactions hereunder. Customer shall comply with all applicable tax laws and regulations, and Customer will promptly pay or reimburse Juniper for all costs and damages related to any liability incurred by Juniper as a result of Customer's non-compliance or delay with its responsibilities herein. Customer's obligations under this Section shall survive termination or expiration of this Agreement.
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 Customer may contain encryption or other capabilities restricting Customer's 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. **Interface Information.** To the extent required by applicable law, and at Customer's written request, Juniper shall provide Customer with the interface information needed to achieve interoperability between the Software and another independently created program, on payment of applicable fee, if any. Customer shall observe strict obligations of confidentiality with respect to such information and shall use such information in compliance with any applicable terms and conditions upon which Juniper makes such information available.
14. **Third Party Software.** Any licensor of Juniper whose software is embedded in the Software and any supplier of Juniper whose products or technology are embedded in (or services are accessed by) the Software shall be a third party beneficiary with respect to this Agreement, and such licensor or vendor shall have the right to enforce this Agreement in its own name as if it were Juniper. In addition, certain third party software may be provided with the Software and is subject to the accompanying license(s), if any, of its respective owner(s). To the extent portions of the Software are distributed under and subject to open source licenses obligating Juniper to make the source code for such portions publicly available (such as the GNU General Public License ("GPL") or the GNU Library General Public License ("LGPL")), Juniper will make such source code portions (including Juniper modifications, as appropriate) available upon request for a period of up to three years from the date of distribution. Such request can be made in writing to Juniper Networks, Inc., 1194 N. Mathilda Ave., Sunnyvale, CA 94089, ATTN: General Counsel. You may obtain a copy of the GPL at <http://www.gnu.org/licenses/gpl.html>, and a copy of the LGPL at <http://www.gnu.org/licenses/lgpl.html>.
15. **Miscellaneous.** This Agreement shall be governed by the laws of the State of California without reference to its conflicts of laws principles. The provisions of the U.N. Convention for the International Sale of Goods shall not apply to this Agreement. 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 terms 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. This Agreement and associated documentation has been written in the English language, and the Parties agree that the English version will govern. (For Canada: Les parties aux présentes confirment leur volonté que cette convention de même que tous les documents y compris tout avis qui s'y rattache, soient rédigés en langue anglaise. (Translation: The parties confirm that this Agreement and all related documentation is and will be in the English language)).

Document History

| Date | Media Flow Controller Version | Comments |
|-----------|-------------------------------|------------------------|
| 2010-4-27 | Release 2.0 | Document Version 2.0 |
| 2010-5-14 | Release 2.0.1 | Document Version 2.0a |
| 2010-6-17 | Release 2.0.2 | Document Version 2.0b |
| 2010-7-21 | Release 2.0.3 | Document Version 2.0c |
| 2010-9-17 | Release 2.0.4 | Document Version 2.0d |
| 2011-3-1 | Release 2.0.6 | Document version 2.0e |
| 2011-7-7 | Release 2.0.7 | Document Version 2.0.7 |

TABLE OF CONTENTS

CHAPTER 1

| | |
|---|----|
| About This Guide | 27 |
| Media Flow Controller Documentation and Release Notes | 27 |
| Objectives | 28 |
| Audience | 28 |
| Supported Platforms | 28 |
| Documentation Conventions | 28 |
| Documentation Feedback | 29 |
| Requesting Technical Support | 29 |
| Self-Help Online Tools and Resources | 30 |
| Opening a Case with JTAC | 30 |
| Terminology | 31 |

PART 1

Media Flow Controller Administration

CHAPTER 2

| | |
|---|----|
| Media Flow Controller Overview | 37 |
| About Media Flow Controller | 37 |
| Media Flow Controller Environment | 38 |
| Media Flow Controller Minimum System Requirements | 38 |
| Understanding Media Flow Controller | 39 |
| Media Flow Controller Management Interfaces | 41 |
| Command Line Interface (CLI) | 41 |
| Web Interface (Management Console) | 41 |
| SNMP Agent Support | 41 |
| E-mail and E-mail2SMS Alerts | 41 |
| Logs | 42 |
| Media Flow Controller Delivery Methods | 42 |

| | |
|--|----|
| Understanding Media Flow Controller Delivery | 42 |
| Media Delivery Using HTTP | 43 |
| Media Delivery Using RTSP | 43 |
| Media Delivery Using RTMP | 44 |
| Adaptive Bit Rate Streaming | 44 |
| Dynamic URI Remapping | 45 |
| Caching and Origin Clustering | 45 |
| Media Flow Controller Hierarchical Caching | 45 |
| Factors Influencing the Cache-ability of an Object | 46 |
| Connection Pooling | 47 |
| Origin Clustering and Origin Escalation | 47 |
| Media Flow Controller AssuredFlow | 47 |
| Session Admission Control | 48 |
| Media Flow Controller SmoothFlow | 49 |
| How SmoothFlow Works | 50 |
| Media Flow Controller Namespaces | 50 |
| Media Flow Controller Virtual Players | 51 |

CHAPTER 3

| | |
|--|----|
| Media Flow Controller Deployment Guidelines | 53 |
| Media Flow Controller Deployments Overview | 53 |
| Reverse Proxy Deployments | 53 |
| About Reverse Proxies | 54 |
| Reverse Proxy Protocol Support | 54 |
| Reverse Proxy Deployment Requirements | 54 |
| Reverse Proxy Deployment Process | 55 |
| Reverse Proxy Cache Tuning CLI Commands | 56 |
| Reverse Proxy Namespace Examples | 59 |
| Transparent Proxy Deployments | 66 |
| About Transparent Proxies | 66 |
| Transparent Proxy Deployment Requirements | 67 |
| Transparent Proxy Deployment Process | 68 |
| Upgrading for New Transparent Proxy Functions | 69 |
| Transparent Proxy Example Configuration—General | 70 |
| Transparent Proxy Example Configuration—YouTube | 71 |
| DMCA Compliance Transparent Proxy Configuration Requirements | 73 |
| DMCA Compliance Transparent Proxy Configuration Details | 73 |
| Example: DMCA Compliance Transparent Proxy Configuration | 74 |

| | |
|---|----|
| Transparent Proxy Cache Tuning CLI Commands | 74 |
| Transparent Proxy Cache Tuning Examples | 76 |
| Mid-Tier Proxy Deployments | 79 |

CHAPTER 4

| | |
|---|-----------|
| Configuring and Administering Media Flow Controller (CLI) | 81 |
| Before You Configure Media Flow Controller | 82 |
| About the Media Flow Controller CLI | 82 |
| Connecting and Logging In. | 83 |
| Using the Command Modes | 83 |
| Prompt and Response Conventions. | 84 |
| CLI Options. | 84 |
| Logging In to Media Flow Controller for the First Time (CLI) | 85 |
| Using SSH in Automated Scripts (CLI) | 85 |
| Setting SSH Keys for Multiple Hosts | 86 |
| Media Flow Controller System Configuration Overview (CLI) | 88 |
| Configuring Interfaces, Hostname, Domain List, DNS, and Default Gateway (CLI) | 88 |
| Cutting and Pasting an Interface Configuration (CLI) | 90 |
| Example: Media Flow Controller Interface Configuration | 90 |
| Configuring Media Flow Controller Clock and Banners (CLI) | 92 |
| Creating and Configuring Link Bonding and Static Routes (CLI) | 93 |
| Configuring Link Bonding and Static Routes (CLI) | 93 |
| Understanding Authentication, Authorization, and User Options | 95 |
| About MD5, SHA1, AES-128, and DES | 95 |
| User Account Defaults and States | 95 |
| Configuring Media Flow Controller User Accounts (CLI) | 96 |
| Applying the Media Flow Controller License (CLI) | 97 |
| Media Flow Controller Policy Configurations Overview | 98 |
| Setting Analytics Options (CLI) | 99 |
| Configuring Caching Analytics (CLI) | 100 |
| Setting Network Connection Options (CLI) | 100 |
| Using Network Connection Assured Flow | 100 |
| Configuring Network Connections (CLI) | 101 |
| Configuring Media Flow Controller Delivery Protocols (CLI) | 102 |
| Managing the Media Flow Controller Disk Cache (CLI) | 103 |
| Controlling Cookie Cache Behavior (CLI) | 104 |
| Analyzing the Disk Cache. | 105 |
| Disk Cache Problem Solving | 106 |

| | |
|--|-----|
| Disk Cache Error Messages | 107 |
| Replacing Bad Disks | 108 |
| Correcting Mis-Labeled Disk Types | 108 |
| Inserting New Disks into a VXA Series Media Flow Engine | 109 |
| Installing and Using FMS in Media Flow Controller (CLI) | 110 |
| Installing FMS on Media Flow Controller (CLI) | 110 |
| Modifying and Restarting the FMS Service (CLI) | 112 |
| Configuring the FMS Admin Console—First Time (CLI) | 113 |
| Configuring FMS on Media Flow Controller for Video On Demand (CLI) | 114 |
| Using Video Directories for FMS | 116 |
| Applying the Adobe Full-Function FMS Server License (CLI) | 116 |
| Administering Media Flow Controller Overview (CLI) | 117 |
| Checking Media Flow Controller Version and Status | 117 |
| Saving and Applying Configurations, Resetting Factory Defaults (CLI) | 118 |
| Rebooting Media Flow Controller (CLI) | 119 |
| Upgrading Media Flow Controller (CLI) | 119 |
| Configuring the Web Interface (CLI) | 120 |
| Configuring the Web Interface Proxy (CLI) | 120 |
| Configuring Caching All Contents for a Website (CLI) | 121 |

CHAPTER 5

| | |
|--|-----|
| Configuring Virtual Players, Media Fetch and Pre-Staging (CLI) | 123 |
| Creating and Configuring Virtual Players (CLI) | 123 |
| Virtual Player Type generic | 124 |
| Virtual Player Type break | 125 |
| Virtual Player Type qss-streamlet | 125 |
| Virtual Player Type yahoo | 125 |
| Virtual Player Type smoothflow | 125 |
| Virtual Player Type youtube | 126 |
| Virtual Player Type smoothstream-pub | 126 |
| Using assured-flow | 126 |
| Using query-string-parm | 127 |
| Using hash-verify | 127 |
| Using virtual-player type qss-streamlet rate-map | 128 |
| Example: Configuring generic Virtual Players (CLI) | 129 |
| Configuring YouTube Video Caching (CLI) | 131 |
| Using Virtual Player Type YouTube | 131 |
| Tunnel YouTube Seeks | 132 |

| | |
|--|-----|
| Configure YouTube Video Caching | 132 |
| Configuring SmoothStream Video Caching (CLI) | 134 |
| About SmoothStreaming | 134 |
| SmoothStreaming Multi-Bit-Rate Assets | 134 |
| Example: SmoothStreaming Workflow | 134 |
| Configure SmoothStreaming Caching and Delivery (CLI) | 135 |
| Configuring FlashStream Video Caching (CLI) | 135 |
| About FlashStreaming | 135 |
| Example: FlashStreaming Workflow | 136 |
| Configure FlashStreaming Caching and Delivery (CLI) | 137 |
| Configuring NFS Fetch for Images (CLI) | 137 |
| Configuring HTTP Fetch for Videos (CLI) | 138 |
| Configuring RTSP Fetch for Videos (CLI) | 138 |
| Pre-Staging Content with FTP (CLI) | 139 |

CHAPTER 6

| | |
|--|-----|
| Configuring Namespaces (CLI) | 141 |
| Creating a Namespace and Setting Namespace Options (CLI) | 141 |
| Using namespace cache-inherit | 142 |
| Using namespace Cookie-Based Authentication and Filtering | 142 |
| Using namespace delivery protocol {http rtsp} origin-fetch cache-age | 143 |
| Using namespace delivery protocol http origin-fetch cache-fill | 143 |
| Using namespace domain regex | 144 |
| Using namespace domain <FQDN:Port> | 144 |
| Using namespace match <criteria> precedence | 145 |
| Using namespace match uri regex | 146 |
| Using namespace match virtual-host | 146 |
| Using namespace object delete list | 146 |
| Using Namespace Forced Tunneled-Transaction Override | 147 |
| Using namespace for Pre-Staging Content via FTP | 148 |
| Using namespace for Live Streaming Delivery Without Caching | 148 |
| Using namespace for Live Streaming Delivery With Caching | 148 |
| Using namespace for Proxy Configurations | 148 |
| Using namespace for Cluster Configurations | 150 |
| Example: Configuring Media Flow Controller Namespaces (CLI) | 150 |
| Using Namespace for Dynamic URI Remapping | 154 |
| Configuring Dynamic URI Mapping | 155 |
| Media Flow Controller Enhancements for Dynamic URI Mapping | 155 |

| | |
|--|------------|
| Dynamic URI Websites' Namespace Configuration Examples | 156 |
| CHAPTER 7 | |
| Configuring Media Flow Controller Load Balancing | 159 |
| Media Flow Controller Load Balancing Overview | 159 |
| Load Balancing with Direct Server Return | 159 |
| CHAPTER 8 | |
| Configuring Media Flow Controller Server Maps | 161 |
| Media Flow Controller Server Map Overview | 161 |
| Server Map Format Types | 161 |
| Origin Clustering Using Media Flow Controllers | 162 |
| Creating the cluster-map XML File | 163 |
| Sample cluster-map XML File for Origin-Based Clusters | 163 |
| Server Map Example: cluster-map DTD | 164 |
| Origin Server Load Distribution and Failover | 164 |
| Creating the origin-escalation-map XML File | 165 |
| Server Map Example origin-escalation-map DTD | 166 |
| Creating the host-origin-map XML File | 167 |
| Server Map Example: host-origin-map DTD | 168 |
| Creating the nfs-map XML File | 169 |
| Configuring Server Maps (CLI) | 170 |
| Example cluster-map Configuration | 171 |
| Example: origin-escalation-map Configuration | 171 |
| Example: HTTP host-origin-map Configuration | 171 |
| CHAPTER 9 | |
| Configuring and Using Media Flow Controller Logs and Alarms | 173 |
| Media Flow Controller Logging Overview | 173 |
| System Baseline and Health | 173 |
| Media Flow Controller Log Codes and Sub-Codes | 174 |
| Status and Error Codes | 174 |
| Status and Error Sub-Codes | 175 |
| Configuring Media Flow Controller Service Logs Overview | 183 |
| About Log Rotation | 183 |
| Using Accesslog Copy With SFTP | 183 |
| Access Log Format Options | 183 |
| Stream Log Format Options | 185 |

| | |
|---|-----|
| Error Log Options | 187 |
| Error Log Module Options | 187 |
| Configuring Media Flow Controller Service Logs (CLI). | 188 |
| Reading Media Flow Controller Service Logs Overview. | 189 |
| Reading the Service Log (accesslog). | 190 |
| Reading the Cache Log (cachelog) | 190 |
| Reading the Error Log (errorlog) | 191 |
| Reading the FMSAccess Log (fmsaccesslog) | 192 |
| Reading the FMSEdge Log (fmsedgelog) | 193 |
| Reading the FMSCollector Log / fuselog | 194 |
| Reading the Stream Log (streamlog) | 194 |
| Reading the Trace Log (tracelog). | 195 |
| Configuring Media Flow Controller System Log. | 198 |
| System Log Severity Levels and Classes | 198 |
| Configuring Media Flow Controller System Logging (CLI) | 199 |
| Reading the Media Flow Controller System Log | 201 |
| Reading the Media Flow Controller Tech-Support Log. | 201 |
| Configuring Media Flow Controller Log Statistics Thresholds (CLI) | 202 |
| Stats Reports Names Options | 204 |
| Measurement Counters (stats samples). | 204 |
| Configuring Media Flow Controller Stats Alarms | 205 |
| Configuring Media Flow Controller Fault Notifications (CLI) | 208 |

CHAPTER 10

| | |
|--|------------|
| Troubleshooting Media Flow Controller | 211 |
| Viewing Information Using Show Commands | 211 |
| Internal Watchdog | 216 |
| Testing Network Connectivity | 217 |
| Testing Media Flow Controller Delivery Functions | 218 |
| Testing HTTP Origin Fetch. | 218 |
| Testing NFS Origin Fetch. | 221 |
| Testing a Specific Transaction | 223 |
| Enabling Debug Operations | 223 |
| Troubleshooting Media Flow Controller Invalid Licenses | 224 |
| Troubleshooting namespace match uri Configuration | 225 |
| Troubleshooting namespace domain Configuration. | 225 |
| Troubleshooting File Not Getting Cached | 225 |
| Troubleshooting Cache Promotion Not Happening | 226 |

| | |
|---|-----|
| Troubleshooting Incoming Requests' URL Length | 226 |
| Troubleshooting Accesslog SFTP | 226 |
| Troubleshooting Lost Admin Password | 227 |
| About the Media Flow Controller Boot Process | 228 |
| Password Database Reset Procedure | 228 |
| Troubleshooting No Web Interface Access | 230 |
| Troubleshooting Accesslog Rotation Intervals | 230 |

CHAPTER 11

| | |
|---|------------|
| Configuring Media Flow Controller (Web Interface) | 231 |
| About the Media Flow Controller Web Interface | 232 |
| Connecting and Logging In | 232 |
| Logging In to Media Flow Controller for the First Time (Web Interface) | 233 |
| Configuring Media Flow Controller for the First Time | |
| (Web Interface EZconfig) | 233 |
| Setting the System Hostname (EZconfig) | 234 |
| Setting Network Parameters (EZconfig) | 234 |
| Creating a Virtual Player (EZconfig) | 235 |
| Adding a Namespace (EZconfig) | 236 |
| Enabling Interfaces (EZconfig) | 237 |
| Restarting Services (EZconfig) | 237 |
| Monitoring Media Flow Controller Statistics (Web Interface) | 238 |
| Viewing Media Flow Controller Summary | 238 |
| Viewing Media Flow Controller Statistics | 238 |
| Viewing Media Flow Controller Bandwidth Usage | 239 |
| Viewing Media Flow Controller Namespace Counters | 240 |
| Viewing Media Flow Controller CPU Load | 240 |
| Viewing Network Usage | 241 |
| Viewing Memory Utilization | 242 |
| System Configuration Overview (Web Interface) | 244 |
| Configuring Interfaces, Default Gateway, Static Routes, DNS and Domain Names, Hostname, and Banners (Web Interface) | 245 |
| Configuring Interfaces (Web Interface) | 245 |
| Setting the Default Gateway and Static Routes (Web Interface) | 248 |
| Configuring DNS and Domain Names (Web Interface) | 249 |
| Setting Hostnames and Banners (Web Interface) | 251 |
| Configuring Static Hosts and ARP (Web Interface) | 253 |
| Configuring Static Hosts | 253 |

| | |
|--|-----|
| Configuring ARP (Web Interface) | 254 |
| Configuring Date, Time, and NTP (Web Interface) | 256 |
| Configuring the System Date and Time (Web Interface) | 256 |
| Configuring NTP (Web Interface) | 257 |
| Configuring RADIUS, TACACS+, and SSH (Web Interface) | 259 |
| Configuring RADIUS (Web Interface) | 259 |
| Configuring TACACS+ (Web Interface) | 262 |
| Configuring SSH (Web Interface) | 264 |
| Configuring Users and AAA (Web Interface) | 265 |
| Configuring Users (Web Interface) | 265 |
| Configuring AAA (Web Interface) | 268 |
| Configuring Faults and Logging (Web Interface) | 269 |
| Configuring Fault Reporting (Web Interface) | 269 |
| Configuring System Logging (Web Interface) | 272 |
| Administering Media Flow Controller Overview | 276 |
| Managing Configuration Files (Web Interface) | 276 |
| Configuration Files | 277 |
| Active Configuration | 278 |
| Upload Configuration File | 278 |
| Execute CLI Commands | 279 |
| Import Configuration | 279 |
| Installing Licenses (Web Interface) | 279 |
| Installed Licenses | 280 |
| Add New Licenses | 280 |
| Restarting Services | 281 |
| Upgrading the System (Web Interface) | 281 |
| Installed Images | 282 |
| Install New Image | 282 |
| Rebooting the System (Web Interface) | 283 |
| Reboot or Shutdown | 283 |
| Configuring the Web Interface (Web Interface) | 284 |
| Configuring the Web Interface Proxy (Web Interface) | 285 |
| Service Configurations Overview | 285 |
| Configuring Network Connections (Web Interface) | 286 |
| Configuring Delivery Protocols (Web Interface) | 287 |
| Set HTTP Listen Port | 287 |
| Configure/Add Selected HTTP Listen Interfaces | 288 |
| HTTP Listen Interfaces | 288 |
| HTTP Listen Ports | 289 |

| | |
|--|-----|
| Configuring Virtual Players (Web Interface) | 289 |
| Add Virtual Player | 289 |
| Configure, Show, or Remove Virtual Players | 290 |
| Virtual Player generic Type Configuration | 290 |
| Virtual Player qss-streamlet Type Configuration | 293 |
| Virtual Player yahoo Type Configuration | 294 |
| Virtual Player youtube Type Configuration | 297 |
| Virtual Player smoothstream-pub Type Configuration | 300 |
| Virtual Player flashstream-pub Type Configuration | 300 |
| Configuring NameSpaces (Web Interface) | 301 |
| Add Namespace | 302 |
| Configuration List | 302 |
| Namespace Configuration | 302 |
| Managing the Media-Cache (Web Interface) | 308 |
| Disk Name | 308 |
| Configuring Service Logging (Web Interface) | 309 |
| Access Log Configuration | 309 |
| Access Log Copy/Auto Download Configuration | 310 |
| Stream Log Configuration | 311 |
| Stream Log Copy/Auto Download Configuration | 311 |
| Configuring Server Maps (Web Interface) | 312 |
| Add Server Map | 312 |
| Configuration List | 313 |
| Refresh Force | 313 |
| Server Map Configuration | 314 |
| Viewing Logs Overview | 315 |
| Viewing the System Log (Web Interface) | 315 |
| Viewing the Service Log (Web Interface) | 316 |
| Viewing the Cache Log (Web Interface) | 316 |
| Viewing the Trace Log (Web Interface) | 317 |
| Viewing the Stream Log (Web Interface) | 317 |
| Viewing the FMSCollector Log (Web Interface) | 317 |
| Viewing the FMSAccess Log (Web Interface) | 317 |
| Viewing the FMSEdge Log (Web Interface) | 318 |
| Viewing the Dashboard Overview | 318 |
| Dashboard | 318 |
| Dashboard: Disk Cache | 321 |
| Dashboard: Cache Hit Rate | 322 |
| Viewing Reports (Interface Statistics) | 322 |

| | |
|--|-----|
| Network Usage Last 24 hours | 322 |
| Network Usage Hourly Stats | 324 |
| Bandwidth Usage Last 24 hours. | 325 |

CHAPTER 12

| | |
|--|------------|
| SNMP Support | 327 |
| About SNMP and Media Flow Controller | 327 |
| SNMP Protocol Support | 327 |
| Media Flow Controller MIB Versions | 328 |
| Configuring the SNMP Agent (Web Interface) | 328 |
| Basic SNMP Agent Configuration. | 329 |
| Configuring Trap Sinks | 330 |
| Adding a New Trap Sink. | 330 |
| Configuring the SNMP Agent (CLI) | 331 |
| Configuring Media Flow Controller SNMP and SNMP Alarms | 331 |
| Configuring SNMP | 331 |
| snmp traps events | 333 |
| SNMP Alarms | 333 |

CHAPTER 13

| | |
|--|------------|
| Deploying SmoothFlow for Media Flow Controller | 337 |
| SmoothFlow Deployment Overview | 337 |
| Evaluating Your Needs. | 338 |
| Encoding Requirements | 338 |
| Configuring Media Flow Controller for SmoothFlow (CLI) | 339 |
| Configuring SmoothFlow Virtual Player (CLI). | 339 |
| Configuring SmoothFlow Namespaces (CLI) | 341 |
| Creating SmoothFlow Media Assets Overview | 343 |
| Creating Assets Using an SaaS | 343 |
| Before You Begin Creating Assets Using an SaaS | 344 |
| Steps for Creating Assets Using an SaaS | 345 |
| Using Scripts to Create Assets Using an SaaS | 345 |
| Initiating Encoding Using an SaaS | 345 |
| Verifying that Encoding has Completed | 349 |
| Preparing Media Flow Controller for Assets Created Using an SaaS | 350 |
| Logs for Assets Created Using an SaaS | 351 |
| Creating On-Demand Assets | 352 |
| Before You Begin Creating On-Demand Assets | 353 |

| | |
|---|-----|
| Steps for Creating On-Demand Assets | 353 |
| Bit-Rate Profiles Naming Conventions for On-Demand Assets | 353 |
| Creating the AssetDescription.dat File for On-Demand Assets | 354 |
| Pre-Staging On-Demand Assets | 355 |
| Initiating SmoothFlow Processing for On-Demand Assets | 355 |
| Deploying the SmoothFlow Reference Client Player | 356 |
| Deployment Checklist | 358 |

PART 2

Media Flow Controller Command Reference

CHAPTER 14

| | |
|---|-----|
| Media Flow Controller CLI Command Reference | 363 |
| aaa | 365 |
| aaa (authentication) | 365 |
| aaa (authorization) | 365 |
| accesslog | 366 |
| analytics | 367 |
| application | 368 |
| arp | 368 |
| banner | 368 |
| bond | 369 |
| boot | 370 |
| bridge | 371 |
| cachelog | 371 |
| clear | 372 |
| cli | 372 |
| clock | 375 |
| cmc | 375 |
| collect counters | 375 |
| configuration | 376 |
| configuration text | 378 |
| configure | 380 |
| debug | 380 |
| delivery | 380 |
| email | 383 |
| email event name Options | 384 |

| | |
|---|-----|
| email class Options | 385 |
| enable | 386 |
| errorlog | 386 |
| exit | 387 |
| file | 387 |
| fmsaccesslog | 388 |
| fmsedgelog | 389 |
| fuselogs | 390 |
| hostname | 391 |
| image | 391 |
| interface | 392 |
| ip | 394 |
| ip filter chain rule arguments | 397 |
| ldap | 398 |
| license | 399 |
| logging | 400 |
| logging severity level | 402 |
| management | 403 |
| media-cache | 404 |
| mfdlog | 407 |
| namespace | 409 |
| (namespace) delivery protocol http client-request | 413 |
| (namespace) delivery protocol http origin-fetch | 414 |
| (namespace) delivery protocol http origin-request | 415 |
| (namespace) origin-server | 416 |
| (namespace) object list delete revalidate | 418 |
| network | 419 |
| ntp | 422 |
| ntpdate | 423 |
| ping | 423 |
| radius-server | 423 |
| ram-cache | 424 |
| reload | 425 |
| reset | 425 |
| scheduler | 426 |
| server-map | 426 |
| server-map Example for NFS Origin | 427 |
| server-map Example for HTTP Origin | 428 |
| service | 428 |

show 429

slogin..... 430

snmp-server..... 430

 snmp traps 431

 snmp traps events 431

ssh 432

ssh client 432

ssh server 434

stats 435

stats alarms 437

 stats alarm States..... 437

 stats alarm rate-limit count 437

stats CHDs 439

stats samples..... 441

streamlog..... 443

tacacs-server..... 444

tcpdump..... 445

tech-support..... 445

telnet 445

telnet-server..... 445

terminal 446

tracelog 446

traceroute 447

upload 447

username 447

virtual-player 449

virtual-player type generic 450

virtual-player type break 452

virtual-player type qss-streamlet 453

virtual-player type yahoo 454

virtual-player type smoothflow 455

virtual-player type youtube 457

virtual-player type smoothstream-pub 458

web 459

web proxy 460

write 461

INDEX IX.463

LIST OF FIGURES

| | | |
|-----------|---|-----|
| Figure 1 | Juniper Networks Media Flow Controller Operations (Reverse Proxy Deployment)..... | 40 |
| Figure 2 | Media Flow Controller Delivery Options | 43 |
| Figure 3 | Media Flow Controller Cache Ingest and Promotion Process..... | 46 |
| Figure 4 | SmoothFlow™ Deployment Overview..... | 49 |
| Figure 5 | Reverse Proxy Call Flow | 54 |
| Figure 6 | Transparent Proxy Call Flow | 67 |
| Figure 7 | Example Connectivity..... | 91 |
| Figure 8 | Direct Server Return..... | 160 |
| Figure 9 | wget Test for Media Flow Controller HTTP Delivery and Cache..... | 220 |
| Figure 10 | Media Flow Controller Boot Process | 228 |
| Figure 11 | Media Flow Controller Login Page | 232 |
| Figure 12 | EZ Config Page System Hostname Area | 234 |
| Figure 13 | EZ Config Page Network Parameters Area..... | 234 |
| Figure 14 | EZ Config Page Virtual Player Area | 235 |
| Figure 15 | EZ Config Page Add Namespace Area..... | 236 |
| Figure 16 | EZ Config Page Enable Interfaces Area | 237 |
| Figure 17 | EZ Config Page Service Restart Area..... | 237 |
| Figure 18 | Monitoring tab left navigation menu | 238 |
| Figure 19 | Monitoring > Bandwidth Usage (Last Hour) Chart Example | 240 |
| Figure 20 | Monitoring > Namespace Counters Page Detail | 240 |
| Figure 21 | Monitoring > CPU Load Page Detail | 241 |
| Figure 22 | Monitoring > Network Usage (Last Hour) Page Detail | 242 |
| Figure 23 | Monitoring > Memory Utilization and Current Memory Statistics Page Detail..... | 243 |
| Figure 24 | System Config tab left navigation menu..... | 244 |
| Figure 25 | Network Interfaces Page Detail (eth0 state and eth0 configuration)..... | 245 |
| Figure 26 | Network Interfaces Page Detail (DHCP Primary Interface)..... | 247 |
| Figure 27 | Network Interfaces Page Detail (Add new interface alias) | 247 |
| Figure 28 | IP Routing Page, Default Gateway..... | 248 |
| Figure 29 | IP Routing Page, Static and Dynamic Routes..... | 249 |

| | | |
|-----------|--|-----|
| Figure 30 | IP Routing Page, Add Static Route | 249 |
| Figure 31 | DNS Page Detail, Static and Dynamic Name Servers | 250 |
| Figure 32 | DNS Page Detail, Add or Modify Name Servers | 250 |
| Figure 33 | DNS Page Detail, Static and Dynamic Domain Names | 251 |
| Figure 34 | DNS Page Detail, Configured Domain Names | 251 |
| Figure 35 | DNS Page Detail, Add New Domain Name | 251 |
| Figure 36 | Hostname and Banners Page Detail, System Hostname | 252 |
| Figure 37 | Hostname and Banners Page Detail, DHCP Hostname | 252 |
| Figure 38 | Hostname and Banners Page Detail, Banners (more Banners options at bottom of page) 253 | |
| Figure 39 | Static Hosts Page Detail, Static Host Entries | 254 |
| Figure 40 | Static Hosts Page Detail, Add New Host | 254 |
| Figure 41 | Address Resolution Page Detail, Static and Dynamic ARP Entries | 255 |
| Figure 42 | Address Resolution Page Detail, Add Static Entry | 255 |
| Figure 43 | Address Resolution Page Detail, Clear Dynamic ARP Cache | 256 |
| Figure 44 | System Config > Date and Time Page Detail (Date and Time) | 256 |
| Figure 45 | NTP Page | 257 |
| Figure 46 | NTP Page | 258 |
| Figure 47 | NTP Page | 258 |
| Figure 48 | NTP Page Detail, Add NTP Server | 259 |
| Figure 49 | RADIUS Page Detail, Default RADIUS Settings | 260 |
| Figure 50 | RADIUS Page Detail, RADIUS Servers | 260 |
| Figure 51 | RADIUS Page Detail, Add New RADIUS Server | 261 |
| Figure 52 | TACACS+ Page Detail, Default TACACS+ Settings | 262 |
| Figure 53 | TACACS+ Page Detail, TACACS+ Servers | 262 |
| Figure 54 | TACACS+ Page Detail, Add New TACACS+ Server | 263 |
| Figure 55 | SSH Page | 264 |
| Figure 56 | Users Page Detail, Active Users | 265 |
| Figure 57 | Users Page Detail, User Accounts | 266 |
| Figure 58 | Users Page Detail, Add New User | 267 |
| Figure 59 | Users Page Detail, admin Password (additional Password areas are displayed for each user) | 267 |
| Figure 60 | AAA Page Detail, Authentication Method List | 268 |
| Figure 61 | AAA Page Detail, Authorization | 268 |
| Figure 62 | Faults Page Detail, Fault Reporting | 270 |
| Figure 63 | Faults Page Detail, Notify Recipients | 271 |
| Figure 64 | Faults Page Detail, Add New Notify Recipients | 272 |

| | | |
|------------|---|-----|
| Figure 65 | Logging Page Detail, Local Log Filtering..... | 273 |
| Figure 66 | Logging Page Detail, Local Log Rotation | 274 |
| Figure 67 | Logging Page Detail, Remote Log Sinks..... | 275 |
| Figure 68 | Logging Page Detail, Add New Remote Sink..... | 275 |
| Figure 69 | Logging Page Detail, Log Format..... | 276 |
| Figure 70 | System Config > Configurations Page Detail, Configuration Files | 277 |
| Figure 71 | System Config > Configurations Page Detail, Active Configuration | 278 |
| Figure 72 | System Config > Configurations Page Detail, Upload Configuration | 278 |
| Figure 73 | System Config > Configurations Page Detail, Import Configuration | 279 |
| Figure 74 | System Config > License Page Detail, Installed Licenses | 280 |
| Figure 75 | System Config > License Page Detail, Add New Licenses | 280 |
| Figure 76 | System Config > Restart Services Page | 281 |
| Figure 77 | System Config > Upgrade Page Detail, Installed Images | 282 |
| Figure 78 | System Config > Upgrade Page Detail, Install New Image..... | 282 |
| Figure 79 | System Config > Reboot Page Detail..... | 283 |
| Figure 80 | Web Settings Page Detail, Web UI Configuration | 284 |
| Figure 81 | Web Settings Page Detail, Web Proxy Configuration | 285 |
| Figure 82 | Service Config tab left navigation menu..... | 285 |
| Figure 83 | Delivery Network Page..... | 286 |
| Figure 84 | Delivery Protocol Configuration Page Detail, Set HTTP Listen Port | 287 |
| Figure 85 | Delivery Protocol Configuration Page Detail, Configure/Add Selected HTTP Listen Interfaces | 288 |
| Figure 86 | Delivery Protocol Configuration Page Detail, HTTP Listen Interfaces | 288 |
| Figure 87 | Delivery Protocol Configuration Page Detail, HTTP Listen Ports..... | 289 |
| Figure 88 | Virtual Player Page Detail, Add Virtual Player..... | 289 |
| Figure 89 | Virtual Player Page Detail, List of Virtual Players Added | 290 |
| Figure 90 | Virtual Player Type generic Configuration Page | 291 |
| Figure 91 | Virtual Player Type qss-streamlet Configuration Page..... | 294 |
| Figure 92 | Virtual Player Type yahoo Configuration Page | 295 |
| Figure 93 | Virtual Player Type youtube Configuration Page | 298 |
| Figure 94 | Virtual Player Type smoothstream-pub Configuration Page | 300 |
| Figure 95 | Virtual Player Type flashstream-pub Configuration Page | 301 |
| Figure 96 | Service Config > Namespace Page Detail, Add Namespace | 302 |
| Figure 97 | Service Config > Namespace Page Detail, Configuration List..... | 302 |
| Figure 98 | Service Config > Namespace Configure Page Detail, Origin Server Configuration..... | 303 |
| Figure 99 | Service Config > Namespace Configure Page Detail, Match Details..... | 304 |
| Figure 100 | Service Config > Namespace Configure Page Detail, Parameters..... | 305 |

| | |
|---|-----|
| Figure 101 Service Config > Namespace Configure Page, Pre-Stage User Configuration | 306 |
| Figure 102 Service Config > Namespace Configure Page, HTTP Origin Fetch Configuration | 307 |
| Figure 103 Service Config > Namespace Configure Page, RTSP Origin Fetch Configuration | 308 |
| Figure 104 Service Config > Media-Cache Page Detail | 309 |
| Figure 105 Log Configuration Page Detail, Access Log Configuration | 309 |
| Figure 106 Log Configuration Page Detail, Access Log Copy/Auto Upload Configuration | 310 |
| Figure 107 Log Configuration Page Detail, Stream Log Configuration | 311 |
| Figure 108 Log Configuration Page Detail, Stream Log Copy/Auto Upload Configuration | 311 |
| Figure 109 Server map Configuration Page Detail, Add Servermap | 312 |
| Figure 110 Server map Configuration Page Detail, Configuration List | 313 |
| Figure 111 Server map Configuration Page Detail, Refresh Force | 313 |
| Figure 112 Server map Configuration Page Detail, Server map Configuration | 314 |
| Figure 113 Logs tab left navigation menu | 315 |
| Figure 114 Media Flow Controller Example Current Log | 316 |
| Figure 115 Dashboard Tab Left Navigation Menu | 318 |
| Figure 116 Media Flow Controller Dashboard Example | 320 |
| Figure 117 Media Flow Controller Disk Cache Graph Example | 321 |
| Figure 118 Media Flow Controller Log Analysis Page Example, Domain Hotness Analysis | 322 |
| Figure 119 Reports tab left navigation menu | 322 |
| Figure 120 Network Usage Report | 323 |
| Figure 121 Network Usage Daily Stats Last 24 Hours of Activity | 324 |
| Figure 122 Bandwidth Usage Report | 325 |
| Figure 123 System Config > SNMP Page | 329 |
| Figure 124 System Config > SNMP Page | 330 |
| Figure 125 System Config > SNMP Page | 330 |
| Figure 126 Publishing Workflow, Typical Steps | 344 |
| Figure 127 On-Demand Workflow, Typical Steps | 353 |

LIST OF TABLES

| | | |
|----------|---|-----|
| Table 1 | Text and Syntax Conventions Used in This Book | 28 |
| Table 2 | Media Flow Controller Recommended Hardware Configuration | 38 |
| Table 3 | Protocol Support Matrix..... | 42 |
| Table 4 | Transparent Proxy versus Reverse Proxy..... | 53 |
| Table 5 | Cache Performance Tuning Settings for Reverse Proxy..... | 56 |
| Table 6 | Namespace origin-server Settings for Transparent Proxy Type* | 69 |
| Table 7 | DMCA Requirements to Media Flow Controller Functionality | 73 |
| Table 8 | Cache Performance Tuning Settings for Transparent Proxy..... | 74 |
| Table 9 | Example Machine Setup of Management and Traffic Ports | 90 |
| Table 10 | Hotness Thresholds | 99 |
| Table 11 | YouTube Formats..... | 131 |
| Table 12 | Example namespace domain regex Entries..... | 144 |
| Table 13 | Example namespace match uri regex Entries..... | 146 |
| Table 14 | Namespace origin-server and origin-request Dependencies per Proxy Mode..... | 149 |
| Table 15 | Logging Status (%s) HTTP Codes | 174 |
| Table 16 | Additional Logging Status Sub-Codes..... | 175 |
| Table 17 | Media Flow Publisher Publish Log, Error Codes..... | 180 |
| Table 18 | Accesslog Format Options | 184 |
| Table 19 | Streamlog format Options | 185 |
| Table 20 | Error Log Levels..... | 187 |
| Table 21 | Error Log Modules..... | 187 |
| Table 22 | Delivery Protocol HTTP Trace Points..... | 196 |
| Table 23 | Stats Samples | 204 |
| Table 24 | Media Flow Controller Stats Alarms | 207 |
| Table 25 | TCP/IP Diagnostic Utilities | 217 |
| Table 26 | SNMP Protocol Support | 327 |
| Table 27 | Media Flow Controller MIB Versions | 328 |
| Table 28 | SNMP Traps Notify-able Events..... | 333 |
| Table 29 | SNMP Alarms, Possible Causes, and Recommended Actions..... | 334 |
| Table 30 | Media Flow Controller Acceptable Containers and Codecs | 338 |

| | | |
|----------|--|-----|
| Table 31 | setup.xml File Parameters..... | 346 |
| Table 32 | asset.xml File Parameters..... | 348 |
| Table 33 | jobid.xml File Parameters..... | 349 |
| Table 34 | segment_config.xml Parameters..... | 350 |
| Table 35 | | 352 |
| Table 36 | | 352 |
| Table 37 | | 352 |
| Table 38 | Required Configurable Nodes and Parameters | 357 |
| Table 39 | Logging Severity Levels | 402 |
| Table 40 | SNMP Traps Notify-able Events..... | 432 |
| Table 41 | Stats Alarms | 438 |
| Table 42 | Stats CHDs..... | 439 |
| Table 43 | Stats Samples | 441 |

CHAPTER 1

About This Guide

This preface provides the following guidelines for using the *Juniper Networks Media Flow Controller Administrator's Guide and CLI Command Reference*:

- [Media Flow Controller Documentation and Release Notes](#)
- [Objectives](#)
- [Audience](#)
- [Supported Platforms](#)
- [Documentation Conventions](#)
- [Documentation Feedback](#)
- [Requesting Technical Support](#)
- [Terminology](#)

Media Flow Controller Documentation and Release Notes

For a list of related Media Flow Controller documentation see [Technical Documentation page for Media Flow](#).

If information in the Release Notes differs from the information in this guide, follow the *Media Flow Controller Release Notes*.

To obtain the most current version of all Juniper Networks® technical documentation, see the product documentation page on the Juniper Networks website at <http://www.juniper.net/techpubs/>.

Juniper Networks supports a technical book program to publish books by Juniper Networks engineers and subject matter experts with book publishers around the world. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration using Junos OS and Juniper Networks devices. In addition, the Juniper Networks Technical Library, published in conjunction with O'Reilly Media, explores improving network security, reliability, and availability using Junos OS configuration techniques. All the books are for sale at technical bookstores and book outlets around the world. The current list can be viewed at <http://www.juniper.net/books>.

Objectives

This guide describes how to use the Media Flow Controller command line interface (CLI) and Web interface, to configure and administer Media Flow Controller media delivery and caching.



NOTE: For additional information about the Junos OS—either corrections to, or information that might have been omitted from this guide—see the software Release Notes for your version at the [Technical Documentation page for Media Flow](#).

Audience

This guide is designed for network system administrators who are configuring and monitoring a Juniper Networks Media Flow Controller media delivery and caching appliance.

To use this guide you need a broad understanding of networks in general, the Internet in particular, networking principles, and networking configuration. You must also be familiar with authentication scheme configurations, query parameter configurations, and media delivery protocols, such as HTTP, RTSP, RTMP, and so forth.

Supported Platforms

For the features described in this guide, Juniper Networks Media Flow Controller currently supports installation on the Juniper Networks VXA Series and generic x86 servers. For details, see the [Media Flow Controller With VXA Series and Media Flow Controller datasheet](#).

Documentation Conventions

Table 1 Text and Syntax Conventions Used in This Book

| Convention | Description | Example |
|--|---|---|
| Plain Text | Ordinary text. | The origin server organizes media content hierarchically. |
| Bold Text | Commands in running text, and screen elements such as page titles, and option labels. | Use the interface command to configure IP addresses. In the Management Console, use the Setup > Date and time page. |
| <i>Italic Text</i> | Book titles, and emphasis. | See the <i>Juniper Networks Media Flow Controller Administrator's Guide and CLI Command Reference</i> |
| Syntax Conventions in the Command Reference Chapter | | |
| Fixed-width Text | Command keywords. Text displayed online at a command line. | <code>interface <interface_name></code> Please enter your IP address |

Table 1 Text and Syntax Conventions Used in This Book (Continued)

| Convention | Description | Example |
|--|--|---|
| Fixed-width Bold Text | Command text that you type. | interface eth0 ip address <IP address> |
| < > (angle brackets) | Text enclosed in angel brackets (< >) is variable and must be replaced by whatever it represents. In the example to the right, the user would replace <file_name> with the name of the specific file. | show file <file_name> |
| [] (box brackets) | Optional commands. Anything not enclosed in brackets must be specified. | web proxy host <IP_address> [port <TCP_port>] |
| { } (braces) | Represent a set of mutually exclusive options, where one option is required. | web proxy auth authtype {none basic} |
| (pipe symbol) | Separates mutually exclusive options. You can enter one of the options separated by the vertical bar, but you cannot enter multiple options in a single use of the command. A vertical bar can be used to separate optional or required options. | analytics last-evict-time diff <1 seconds> |
| ... (ellipsis) | An ellipsis (...) indicates that the previous option can be repeated multiple times with different values. It can be used inside or outside of brackets. | clock timezone <zone> [<zone>] ... |

Documentation Feedback

We encourage you to provide feedback, comments, and suggestions so that we can improve the documentation. You can send your comments to techpubs-comments@juniper.net, or fill out the documentation feedback form at <https://www.juniper.net/cgi-bin/docbugreport/>. If you are using e-mail, be sure to include the following information with your comments:

- Document or topic name
- URL or page number
- Software release version (if applicable)

Requesting Technical Support

Technical product support is available through the Juniper Networks Technical Assistance Center (JTAC). If you are a customer with an active J-Care or JNASC support contract, or are covered under warranty, and need post-sales technical support, you can access our tools and resources online or open a case with JTAC.

- **JTAC Policies**—For a complete understanding of our JTAC procedures and policies, review the *JTAC User Guide* located at <http://www.juniper.net/customers/support/downloads/710059.pdf>
- **Product Warranties**—For product warranty information, visit <http://www.juniper.net/support/warranty/>
- **JTAC Hours of Operation**—The JTAC centers have resources available 24 hours a day, 7 days a week, 365 days a year.

Self-Help Online Tools and Resources

For quick and easy problem resolution, Juniper Networks has designed an online self-service portal called the Customer Support Center (CSC) that provides you with the following features:

- Find CSC offerings:
<http://www.juniper.net/customers/support/>
- Search for known bugs:
<http://www2.juniper.net/kb/>
- Find product documentation:
<http://www.juniper.net/techpubs/>
- Find solutions and answer questions using our Knowledge Base:
<http://kb.juniper.net/>
- Download the latest versions of software and review release notes:
<http://www.juniper.net/customers/csc/software/>
- Search technical bulletins for relevant hardware and software notifications:
<https://www.juniper.net/alerts/>
- Join and participate in the Juniper Networks Community Forum:
<http://www.juniper.net/company/communities/>
- Open a case online in the CSC Case Manager:
<http://www.juniper.net/cm/>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool located at

<https://tools.juniper.net/SerialNumberEntitlementSearch/>

Opening a Case with JTAC

You can open a case with JTAC on the Web or by telephone.

- Use the Case Manager tool in the CSC at <http://www.juniper.net/cm/>
- Call 1-888-314-JTAC
(1-888-314-5822—toll free in the USA, Canada, and Mexico)

For international or direct-dial options in countries without toll-free numbers, visit

<http://www.juniper.net/support/requesting-support.html>

Terminology

This section provides definitions for Media Flow Controller terms and industry-standard terms that may be unfamiliar to the reader.

Absolute URL An absolute URL points to the exact location of a file or directory on the Internet, by name, including the full path. Contrast with [Relative URL](#) and [Base URL](#).

AFR Assured Flow Rate. A Media Flow Controller option that, when enabled, ensures that media content is delivered at a rate that is minimally needed for the video to play smoothly.

ARP Address Resolution Protocol; allows systems to map IP addresses to MAC addresses.

Bit-rate A data rate (the amount of data transferred in one direction over a link divided by the time taken to transfer it) expressed in bits per second. Juniper Networks notation examples: **Kbps** (kilobits per second), **KB/s** (kilobytes per second). See also [Profile \(bit-rate profile\)](#).

Base URL The leading portion of a URL minus the name component. Example: The base-URL of `"/a/b/c/index.html"` is `"/a/b/c"`. Contrast with [Absolute URL](#) and [Relative URL](#).

Broadcast A type of network routing scheme where data is sent to all possible destinations on a network. Contrast with [Multicast](#) and [Unicast](#).

CDN Content Delivery Network. A system of computers networked together across the Internet that cooperate transparently to deliver content most often for the purpose of improving performance, scalability, and cost efficiency, to end users.

CHD Computed Historical Datapoints; traffic samples that have been computed in some manner, such as summation and averaging.

CLI Command-line interface.

Client Node or software program (front-end device) that requests services from a server.

CMC Central Management Console, Juniper Networks management interface that allows you to push configurations to a number of Media Flow Controllers from a central interface. In Release 2.0.7, this function is deprecated.

DHCP Dynamic Host Configuration Protocol.

DSR Direct Server Return. A method of handling TCP traffic in the context of utilizing a proxy.

Edge cache An appliance, placed between the Internet and close to the end user, that caches and delivers content (such as Java Script, CSS, images, and so forth.) freeing up Web servers for other processes. Media Flow Controller as an edge cache is effectively a "reverse proxy," that provides these benefits: reduces the load (network and CPU) on an origin server by servicing previously retrieved content and enhances the user experience due to a decrease in latency.

FMS Flash Media Server.

Full Download An HTTP media delivery mode in which the entire media file is downloaded *before playback*; contrast with [Progressive download \(PDL\)](#).

HDD Hard disk drives.

"Hot" content (short tail versus long tail) When content is often requested it becomes "hot." Media Flow Controller caches content hierarchically based on hotness. Short tail videos are those that are often requested: a few videos requested by many different clients. Long tail videos are those that are seldom requested: many different videos requested by few clients.

Hot swap Disconnecting or connecting peripherals without interrupting system operations. Media Flow Controller supports hot swapping of caching storage drives.

Ingest Data placed on a Media Flow Controller, analyzed, and queued; contrast with [Pre-stage](#).

KB and KiB KB=1000 kilobytes (networking), KiB=1024 kilobytes (storage).

Live TV / Linear TV These terms mean the same thing; this document uses “live TV.”

Local boot This refers to booting from the default boot partition on the system; for example, when the **reboot** command is given.

MB, MiB, and Mbit MB=1,000,000 megabytes (networking). MiB=1,048,576 (1024 x 1024) megabytes (storage). Mbit=1,000,000 x 8 megabits (data transfer).

Mid-tier proxy A **mid-tier proxy** sits between the origin servers and the edge, and serves requests from the edge caches. Mid-tier proxies improve response time for requests because content is closer to the user; and off-load origin servers from repeat requests from the edge.

MTU Maximum transmission unit. The size (in bytes) of the largest packet or frame that a given layer of a communications protocol can pass onwards.

Multicast A type of network routing scheme where data is sent to certain destinations based on address. Contrast with [Broadcast](#), and [Unicast](#).

Namespace A defined collection of delivery policies for different categories of content or domains.

Network prefix An IPv4 network prefix specifying a network. Used in conjunction with a netmask to determine which bits are significant. For example, “192.168.0.0”.

NFS (network file system) A protocol that allows a user on a client computer to access files over a network similarly to how local storage is accessed.

NIC Network interface controller/card.

NTP Network Time Protocol.

Origin library The source of media content, typically a server located at a data center.

Origin server The media content server. Juniper Networks Media Flow Controller can be configured as an origin server.

Player (media player software) Any media player for playing back digital video data from files of appropriate formats such as MPEG, AVI, RealVideo, Flash, QuickTime, and so forth. In addition to VCR-like functions such as playing, pausing, stopping, rewinding, and forwarding, some common functions include zooming/full screen, audio channel selection, subtitle selection, and frame capturing.

Prefix mode When you enter a command that has prefix mode, you enter a mode for just that configuration; see [“Using the Command Modes” on page 83](#) for details.

Pre-stage Data placed on a Media Flow Controller or origin server before an HTTP request comes in for it. Contrast with [Ingest](#).

Profile (bit-rate profile) A media “bit-rate profile” is the bit-rate encoding that allows optimal downloads to different bandwidths.

Progressive download (PDL) An HTTP media delivery mode in which the media file is played while it is being downloaded; contrast with [Full Download](#).

Publishing point A way to distribute content to your users (live or broadcast as live); either through a defined SDP (service delivery protocol) file, or a namespace (**live-pub-point**).

Pull versus Push Pull refers to media fetches from the origin server initiated by Media Flow Controller based on received requests. Push refers to scheduled media deliveries from the origin server to Media Flow Controller.

PXE (Preboot eXecution Environment) boot A way to boot computers using a network interface without needing a CDROM or USB drive; PXE must be properly installed first.

Remote Authentication Dial In User Service A networking protocol that provides centralized access, authorization and accounting management for people or computers to connect and use a network service.

Regex An extended regular expression. Enclose all **regex** entries in double quotes; for example, a regex for **www.example.com** plus **example.com** could be this: “**^.*example.com**”. A good website to learn Regex is [Regular Expressions Info](#).

Relative URL A relative URL points to the location of a file from a point of reference, usually the directory beneath. Preceded by two dots (`../directory_path/file.txt`) for the directory above; one dot (`./directory_path/file.txt`) for the current directory. Contrast with [Absolute URL](#) and [Base URL](#).

Reverse proxy A server processing in-bound traffic, installed in front of origin servers. Reverse proxies are used for scaling origin servers, caching (serving commonly-accessed files), load balancing, and security (denying requests, preventing direct origin server access, and so forth).

RTMP Real-Time messaging protocol. A multimedia streaming and RPC (remote procedure call) protocol primarily used in Adobe Flash. RTMP has three variations: the “plain” protocol which works on top of TCP and uses port 1935, RTMPT which is encapsulated within HTTP requests to traverse firewalls, and RTMPS which works just like RTMPT but over a secure HTTPS connection.

RTP Real-Time transport protocol. A standardized packet format for delivering audio and video over the Internet. It is used in conjunction with other protocols such as RTSP. The RTP standard defines a pair of protocols, RTP and RTCP. RTP is used for transfer of multimedia data and RTCP is used to periodically send control information and quality of service (QoS) parameters between the server and client.

RTSP Real-Time Streaming Protocol. An application-level protocol for the control of real-time streaming data sent over RTP. Typically RTP data is sent over UDP, but it can also be sent over the RTSP channel via an interleaved mechanism or over TCP via port 80 with HTTP-like syntax and operations.

RU (rack unit) A unit of measurement of the height of a rack-mounted device.

RX A communications abbreviation for “receive.”

SAS Serial attached SCSI. A data transfer technology designed to move data to and from computer storage devices such as hard drives and tape drives.

SATA Serial Advanced Technology Attachment. A computer bus technology primarily designed for transfer of data to and from a hard disk.

SCP The SCP (secure channel protocol) pseudo-URL format is:

```
scp://<username>[:<password>]@<hostname>/<path>[</filename>]
```

The path is an absolute path. Paths relative to the user's home directory are not currently supported. You must have an SCP server installed in order to SCP files to your machine.

SFTP Secure File Transfer Protocol. The SFTP pseudo-URL format is:

sftp://<username@<hostname>:<path>[</filename>]

The path is an absolute path. Paths relative to the user's home directory are not currently supported. You must have FTP server installed in order to FTP or SFTP, respectively, files to your machine. SFTP uses SSH.

SSD Solid-state drive. A storage device using solid-state memory to store persistent data.

State The current configurations and operational status of the appliance.

Streaming The process of playing a file while it is still being downloaded. Streaming technology lets a user view and hear digitized content as it is being downloaded.

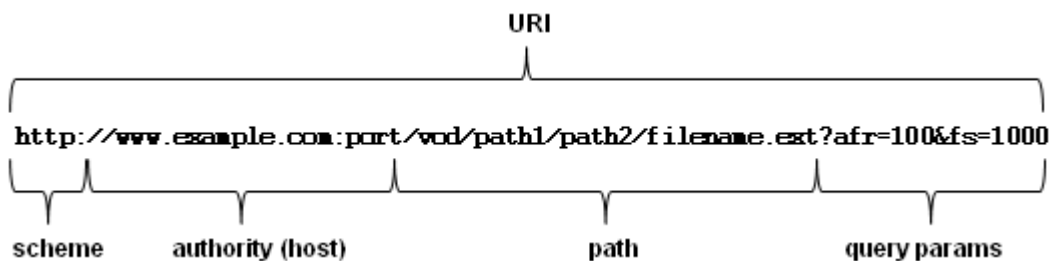
Transparent proxy A proxy that does not modify the request or response beyond what is required for proxy authentication and identification. Transparent proxies help optimize networks transparently (no client configuration required, no modification of traffic done). Media Flow Controller transparent proxy configurations can be done in three ways: origin based on HOST header, origin based on X-NKN or a custom header, and origin based on client destination IP address. Contrast with [Mid-tier proxy](#), and [Reverse proxy](#).

Tunneling When a payload protocol is incompatible with the delivery network, a tunneling protocol can encapsulate it for delivery only; no polices can be applied.

Unicast A type of network routing scheme where data is sent to a single destination host on a network. Contrast with [Broadcast](#), and [Multicast](#).

UOL, URI, URL Uniform Object Locator, Uniform Resource Identifier, Uniform Resource Locator (respectively). When shown as a command variable, for example **<URL>**, this indicates a normal URL, using any protocol that WGET supports, including HTTP, HTTPS, FTP, and TFTP; or a pseudo-URL specifying an SCP file transfer.

uri-prefix This **namespace** argument refines what requests Media Flow Controller accepts. In the URL shown, the **uri-prefix** could be defined as / (slash), **/vod**, or **/vod/path1**. If / (slash) is used, all incoming requests to that domain are honored; if **/vod**, only requests containing “/vod” (and any sub-directory of it) are honored; if **/vod/path1**, requests must include that prefix *and* that sub-directory (sub-sub-directories of **path1** need not be specified).



Virtual host A virtual host is a capability of some computers that can respond to different IP addresses and offer different services appearing to be a distinct host on a distinct machine; a single machine can supply several virtual hosts.

Virtual player This term refers to the server-side player provided by Media Flow Controller to assist in media viewing. Media Flow Controller offers several types of virtual player for use in different scenarios.

VOD Video on-demand.

PART 1

Media Flow Controller Administration

CHAPTER 2

Media Flow Controller Overview

- [About Media Flow Controller](#)
- [Media Flow Controller Environment](#)
- [Media Flow Controller Minimum System Requirements](#)
- [Understanding Media Flow Controller](#)
- [Media Flow Controller Management Interfaces](#)
- [Media Flow Controller Delivery Methods](#)
- [Caching and Origin Clustering](#)
- [Media Flow Controller AssuredFlow](#)
- [Media Flow Controller SmoothFlow](#)
- [Media Flow Controller Namespaces](#)
- [Media Flow Controller Virtual Players](#)

About Media Flow Controller

Juniper Networks Media Flow Controller is a purpose-built appliance that combines media intelligence, storage organization, multi-tier caching, and network optimization to scale media delivery throughput and enhance end-user experience. Media Flow Controller can be deployed by content providers, content delivery networks (CDNs), and network service providers (NSPs). Media Flow Controller can be used for origin acceleration, as well as edge or mid-tier caching, and is designed to:

- Reduce media delivery costs by providing high performance density per server.
- Scale the performance of network storages and origin servers.
- Provide TV-like viewing experience for online video watchers.
- Save transit bandwidth costs, accelerate Web downloads, and improve response time for requests.
- Consolidate servers by delivering mixed-media content using multiple protocols from the same server.

Media Flow Controller Environment

Media Flow Controller software can be deployed in any network that uses the TCP/IP protocol. Media Flow Controller allows you to manage and deploy network bandwidth efficiently, thereby ensuring the highest quality experience for your end users. Media Flow Controller can be easily integrated with existing media servers, and management systems, for incremental and non-disruptive deployment.

- The Media Flow Controller open architecture enables it to integrate easily into existing network and storage infrastructures without requiring disruptive changes. Media Flow Controller supports industry-standard storage interfaces and devices.
- Media Flow Controller supports industry-standard video players, including Flash, QuickTime, SilverLight, and Windows Media Player.
- Media Flow Controller runs on industry-standard x86 64-bit server platforms and Juniper Networks VXA Series appliances.

Media Flow Controller Minimum System Requirements



NOTE: This section provides a high-level overview of Media Flow Controller minimum system requirements. For the most up-to-date and complete information, see the [Media Flow Controller With VXA Series and Media Flow Controller datasheet](#).

Media Flow Controller can run on Juniper Networks VXA Series appliances or standard x86 64-bit servers. When running Media Flow Controller software on generic (non-Juniper Networks) x86 servers, the following are either required or recommended for optimal performance.

Table 2 Media Flow Controller Recommended Hardware Configuration

| Item | Reverse Proxy | Transparent Proxy |
|---|--|---|
| Processor | 64-bit x86, Dual Quad Core or more, minimum of 2.4 GHz speed. | 64-bit x86, Dual Quad Core or more, minimum of 2.4 GHz speed. |
| RAM | 4 GB or more. | 36 GB or more. |
| Direct Attached Storage | Up to 16 direct attached storage (DAS) drives such as SATA, SAS, or SSD, configured as JBODs (just a bunch of disks). | |
| Solid State Drives | Intel Extreme, or Intel Mainstream, for higher performance. | |
| Disk Controllers (RAID should be disabled.) | Dell SAS 6/ir, HP SC44Ge, LSI SAS 3442E-R/3081E-R, LSI 1068/1064/1078, LSI SAS2008, LSI Logic / Symbios Logic SAS1068E PCI-Express Fusion-MPT SAS HP Smart Array G6 controllers (rev01) 3Ware 9690SA | |

Table 2 Media Flow Controller Recommended Hardware Configuration (Continued)

| Item | Reverse Proxy | Transparent Proxy |
|---------------------|---|-------------------|
| Network Controllers | Intel 1GbE (82574, 82580, 82571 (will be EOLed), 82576, 82575eb), Intel 10GbE 82599 Chelsio 10GbE | |
| Network Interfaces | One 100 Mbps or 1 Gbps management interface. Up to 10 x 1 Gbps or 2 x 10 Gbps media delivery interfaces. | |

Understanding Media Flow Controller

Media Flow Controller operates in three different proxy modes: reverse proxy, transparent proxy, and mid-tier proxy.

Media Flow Controller consolidates all streaming protocols (HTTP, RTSP, RTMP) into a single server, reducing the number of servers required to deliver video over multiple protocols.

Media Flow Controller is able to get content from origin servers or origin storages once, and serve it to several users simultaneously.

When a user request is received, Media Flow Controller checks for the presence of the object in its cache file system. If no copy exists in any cache (also known as “cache miss”), Media Flow Controller sends a request to the target origin server, fetches the content, and serves it to the user. Then Media Flow Controller decides whether that content is cache-worthy. Media Flow Controller decides the cache-worthiness based on its intelligent Analytical Engine, request and response headers, and customer-configured policies.

Media Flow Controller keeps track of the popularity of the objects in the cache. The popularity of the object is calculated based on the frequency of hits (i.e., number of repeat requests to an object in a given time period). When objects become “hot” (popular), Media Flow Controller promotes them to a cache tier that supports faster delivery. Promotion in Media Flow Controller can happen starting from the lowest tier; for example, SATA to SAS, SSD, and RAM. This allows Media Flow Controller to scale throughput and meet increased demand.

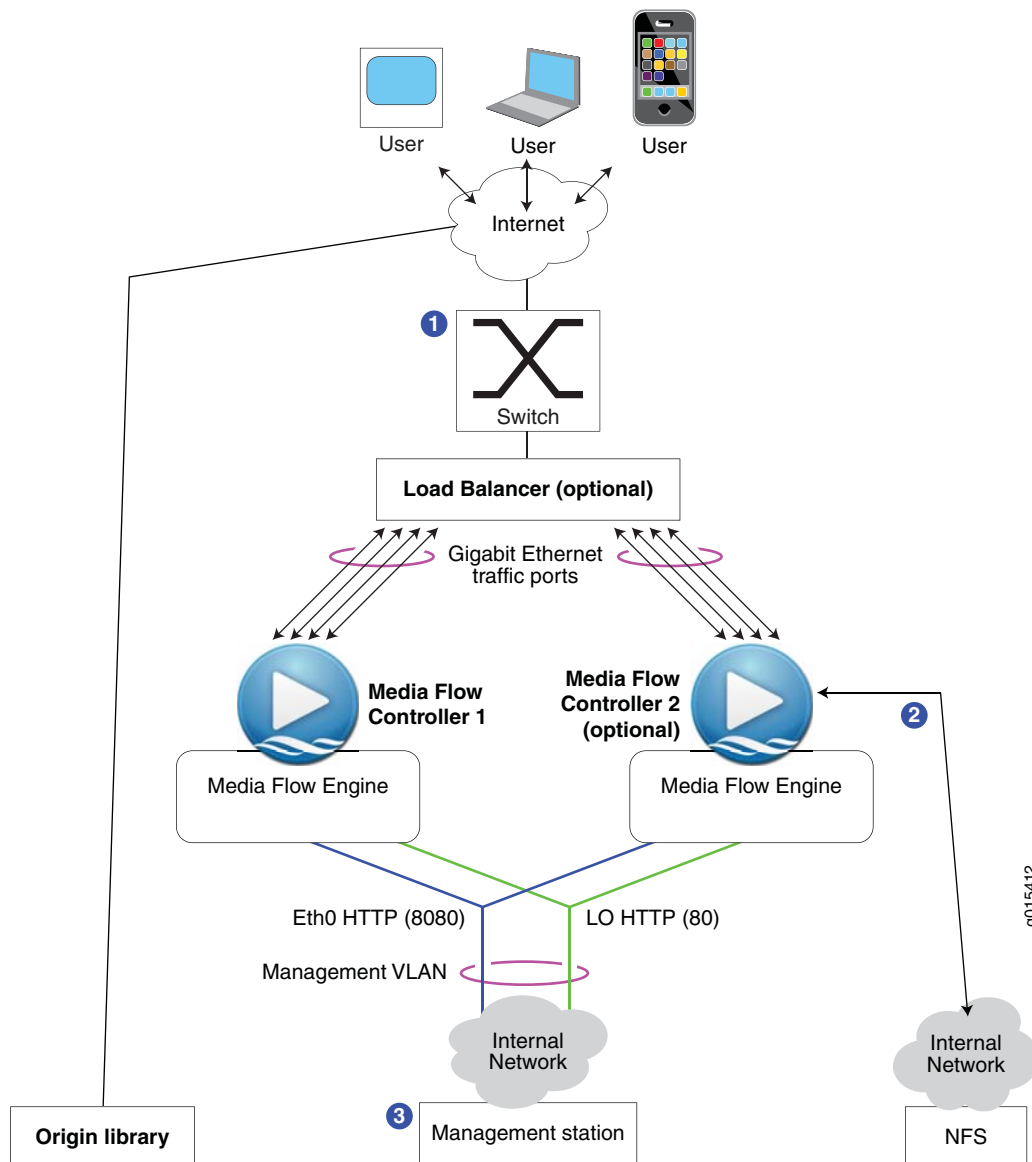


Figure 1 Juniper Networks Media Flow Controller Operations (Reverse Proxy Deployment)

[Figure 1](#) illustrates the media delivery optimization operation.

1. Requests come in from the Internet via HTTP, to (typically) a switch or Load Balancer that redirects the request to Media Flow Controller. Media Flow Controller does certain basic checks, such as URL validation, and parses the URL query string and header fields to identify the content and any associated policies; and then (optionally) calculates the AFR needed to deliver the content, and does a resource check to verify that the content can be delivered in an acceptable manner for that session.
2. Upon a cache miss (content is not cached), Media Flow Controller gets the content from origin, serves it, and caches a copy. For objects larger than 32 KB, serving begins when 32 KB is received; for objects smaller than 32 KB, serving begins at 4 KB or when the object is received. Subsequent requests for the same content are served directly from the cache.
3. Management interfaces monitor activity and allow configuration changes.

Media Flow Controller Management Interfaces

Media Flow Controller can be configured and managed through a variety of interfaces.

- [Command Line Interface \(CLI\)](#)
- [Web Interface \(Management Console\)](#)
- [SNMP Agent Support](#)
- [E-mail and E-mail2SMS Alerts](#)
- [Logs](#)

Command Line Interface (CLI)

You can log in to the Media Flow Controller server via the command-line interface (CLI) by invoking an SSH session. The CLI allows all aspects of system configuration and management.

Web Interface (Management Console)

You can log in to the Media Flow Controller Web interface, also referred to as the Management Console, from a Web browser using HTTP and port 8080 (for example, `http://Media Flow Controller-Hostname:8080/`). HTTPS can also be used if SSH certificates are set up properly. The Web interface makes it easy to configure and manage the system from any remote location. The Web interface provides a powerful dashboard that displays real-time performance data such as number of concurrent connections, cache-hit ration, bandwidth served, and CPU/memory/disk utilization.

SNMP Agent Support

Media Flow Controller includes a Simple Network Management Protocol (SNMP) agent for monitoring various system statistics and parameters. The Media Flow Controller SNMP agent supports only monitoring and does not support any network configuration features.

For more details, see [Chapter 12, "SNMP Support"](#).

E-mail and E-mail2SMS Alerts

Media Flow Controller allows you to be notified via e-mail during events such as high CPU/memory utilization, interface up/down, and threshold crossing on statistics or counters. Media Flow Controller uses SMTP protocol to send e-mails to the administrators. You can use the E-mail 2SMS facility provided by mobile network operators to configure Media Flow Controller to send SMS notifications.

Example:

```
From: "System Administrator" <do-not-reply@mfc.example.com>
Date: December 28, 2009 9:40:47 AM PST
To: admin@example.com
Subject: System event on mfc.example.com: Process exit: ftpd

Hostname: mfc.example.com
Date:      2009/12/28 17:40:47
Description: Unexpected exit of process ftpd.

Uptime:   1h 15m 34.860s
Version:  mfc-1.2.0
```

Logs

Media Flow Controller generates *syslogs* for system events such as interface up/down, user login, configuration logging, software process crashes, and so forth.

Media Flow Controller generates *errorlogs* for debugging and troubleshooting internal system errors.

Media Flow Controller generates W3C/NCSA-compliant *accesslogs* to track all media requests. Media Flow Controller also allows administrators to customize the format of the access log entries. Media Flow Controller can be integrated with third-party reporting tools (such as Sawmill) to generate reports for analyzing usage patterns and for capacity planning.

Media Flow Controller Delivery Methods

- [Understanding Media Flow Controller Delivery](#)
- [Media Delivery Using HTTP](#)
- [Media Delivery Using RTSP](#)
- [Adaptive Bit Rate Streaming](#)
- [Dynamic URI Remapping](#)

Understanding Media Flow Controller Delivery

Media Flow Controller can deliver content simultaneously to a large audience across multiple screens (such as PCs, TVs, and mobile devices), by supporting a wide range of delivery protocols and media formats.

Media Flow Controller consolidates multiple delivery protocols such as HTTP, RTSP, and RTMP; see [Table 3](#). Media Flow Controller can be used for delivering rich media content to users including streaming of videos, both on-demand and live. See [Figure 2](#), for illustration.

Media Flow Controller:

- Efficiently caches objects of all sizes, ranging from small objects (thumbnails) to the largest objects (videos and software downloads)
- Supports delivery via HTTP, RTSP, and RTMP
- Supports various formats required for delivery to different screens
- Supports multi-tenancy to host content belonging to multiple customers

Media Flow Controller supports the entire spectrum of adaptive streaming methods for on-demand and live streaming such as Apple iPhone Streaming, Microsoft Smooth Streaming, Move Adaptive Streaming, and Adobe Dynamic HTTP and RTMP streaming.

Media Flow Controller supports HTTP Progressive Download (PDL).

Table 3 Protocol Support Matrix

| Content Ingest or Publishing Protocol | Delivery Type | Delivery Protocols |
|--|--------------------|------------------------|
| HTTP Fetch | On-demand and live | HTTP, RTP/RTSP*, RTMP* |
| NFS Fetch (file system containing assets mounted on a Media Flow Controller) | On-demand or live | HTTP, RTP/RTSP*, RTMP* |

Table 3 Protocol Support Matrix (Continued)

| Content Ingest or Publishing Protocol | Delivery Type | Delivery Protocols |
|---------------------------------------|--------------------|----------------------|
| FTP (push) | On-demand | HTTP, RTP/RTSP, RTMP |
| RTP and RTSP Ingest | On-demand and live | RTP and RTSP |
| RTMP Ingest | On-demand and live | RTMP |

* Only on-demand media delivery is supported for this combination.

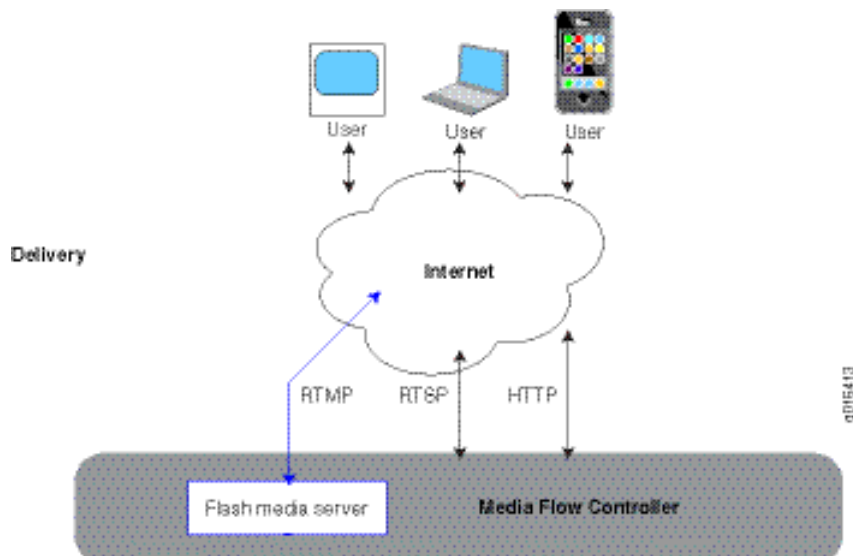


Figure 2 Media Flow Controller Delivery Options

Media Delivery Using HTTP

Media Flow Controller supports delivering rich media content to users with the HTTP protocol. The HTTP protocol can be used for delivering static Web objects such as HTML, images, and thumbnails; and for streaming videos. For HTTP, the supported methods are CONNECT, DELETE, GET, HEAD, OPTIONS, POST, PUT, and TRACE.

Media Delivery Using RTSP

Media Flow Controller improves the performance of media delivery using the real-time streaming protocol (RTSP), by using its caching and network scaling infrastructure. Media Flow Controller supports the trick-play/DVR-like functions based on the RTSP protocol's Pause and Seek operations.



NOTE: Trick-play capability is supported for H.264/AAC encoding and MP4 format, but is not currently supported for WMP/WMV using VC1 encoding or proprietary formats.

Media Flow Controller uses a suite of protocols to deliver videos using RTSP:

- RTSP for establishing and controlling media sessions with the player.
- RTP to deliver the actual audio and video streams.
- RTCP to provide control information for an RTP flow and statistics for a media connection.

Media Flow Controller provides options for you to use both UDP or TCP protocols for transmitting RTP streams. You can use interleaved RTSP to effectively send both control information and data streams on the same transport connection; this reduces the number of ports used for transmitting streams to a player, thereby reducing the resource utilization on firewalls and switches.

The native RTSP implementation in Media Flow Controller Release 2.1 interoperates with Quicktime players and VLC players. The supported methods for RTSP are DESCRIBE, OPTION, PAUSE, PLAY, SETUP, and TEARDOWN; all are required except OPTION and PAUSE, which are recommended.



NOTE: Unlike HTTP, there are multiple implementations of RTSP. The three most popular implementations are from Apple, Real Networks, and Microsoft. Each of these can claim to be RFC compliant and yet be different due to their proprietary features such that they do not interoperate with each other.

Media Delivery Using RTMP

Media Flow Controller supports delivery of on-demand streams using RTMP, RTMPT and RTMPE protocols, by hosting Adobe FMS server (Flash Media Interactive Server Edition). This gives you a full-range of Adobe FMS capabilities including content protection (DRM) and trick-play (DVR) services. Media Flow Controller supports adaptive delivery of live and on-demand videos using Adobe's Dynamic RTMP streaming.

Media Flow Controller scales the delivery capacity of Adobe FMS by leveraging its multi-tier cache file system. In addition, customers can consolidate the delivery of multiple protocols from the same server.



NOTE: You have to buy a license for Adobe FMS 3.5 or FMS 4.0, to deliver streams using RTMP.

Adaptive Bit Rate Streaming

Adaptive Bit Rate Streaming technology requires content to be encoded at multiple bit-rates that are then delivered to clients as a series of small chunks or fragments. This allows the client player to dynamically switch between fragments of different bit-rates depending on the network bandwidth, CPU state, and so forth, allowing viewers to have the best possible viewing experience.

Media Flow Controller can improve the media delivery throughput of other proprietary HTTP-based streaming technologies such as Adobe Flash Streaming, Move Streaming, Apple HTTP Streaming, and Microsoft SmoothStreaming. Media Flow Controller can be used to deliver both on-demand and live video streams to Move player, Apple iPhone/iPod Touch, Adobe Flash, and Microsoft Silverlight players.

SmoothStreaming is an HTTP-based adaptive-streaming technology implemented by Microsoft. The media format defined by Microsoft for smooth streaming supports both storage and on-the-wire delivery, and is based on the ISO/IEC 14496-12 ISO Base Media File Format specification. Similarly, Apple has defined an HTTP Live Streaming (HLS) IETF RFC based on MPEG-2 TS stream segments, along with an extension to their M3U8 playlist format, to support adaptive stream delivery to iPhone Operating System (iOS) devices. Adobe also recently released an HTTP Dynamic Streaming (HDS) variant for their Flash Player, which is also derived from the ISO/IEC 14496-12 Base Media File Format.

Media Flow Publisher provides the capability to publish on-demand and live content to the various adaptive HTTP streaming technologies. Media Flow Controller supports delivering pre-segmented videos over HTTP using adaptive streaming techniques.

Dynamic URI Remapping

Some popular content providers generate dynamically created URIs for the same content for various reasons, including security. This causes caches to have low cache-hit ratios even when the content is in the cache. Media Flow Controller can identify these dynamic URIs as pointing to the same content and can ensure delivery of the correct content.

Media Flow Controller identifies dynamic URIs via regex substring-addressing of matches that allows access to various portions of the matched string, denoted by parenthesis in the regex expression. The complete string match is referred to as **\$0**, the left-most substring is referred to as **\$1**, each subsequent substring being **\$2**, **\$3**, and so on. You configure a regex and mapping string on a per-namespace basis. The mapping string is an ASCII-printable string that describes the mapping from the various substring matches in the regex, to a new URI.

Caching and Origin Clustering

[Media Flow Controller Hierarchical Caching](#)

[Factors Influencing the Cache-ability of an Object](#)

[Connection Pooling](#)

[Origin Clustering and Origin Escalation](#)

Media Flow Controller Hierarchical Caching

When Media Flow Controller fetches data from origin upon cache miss, it caches the data in its local disks. Media Flow Controller has its own optimized storage sub-system in which data is placed intelligently so it can be read back for very fast delivery. Media Flow Controller organizes data in a hierarchical fashion using a cache tier manager that dynamically calculates the “hotness” of the data and places it in the right cache tier (see [“Hot” content \(short tail versus long tail\)” on page 31](#) for explanation of hotness). RAM is the highest tier, followed by SSD, SAS, and SATA, in that order.

Disk speeds are calculated and assigned to a tier in the cache hierarchy as part of the initialization of Media Flow Controller. When data is accessed from origin, it is stored in the lowest cache tier, and promoted to higher cache tiers as the hotness of the data increases. See [“Terminology” on page 31](#) for definition of “ingest.” See [Figure 3](#), for an illustration.

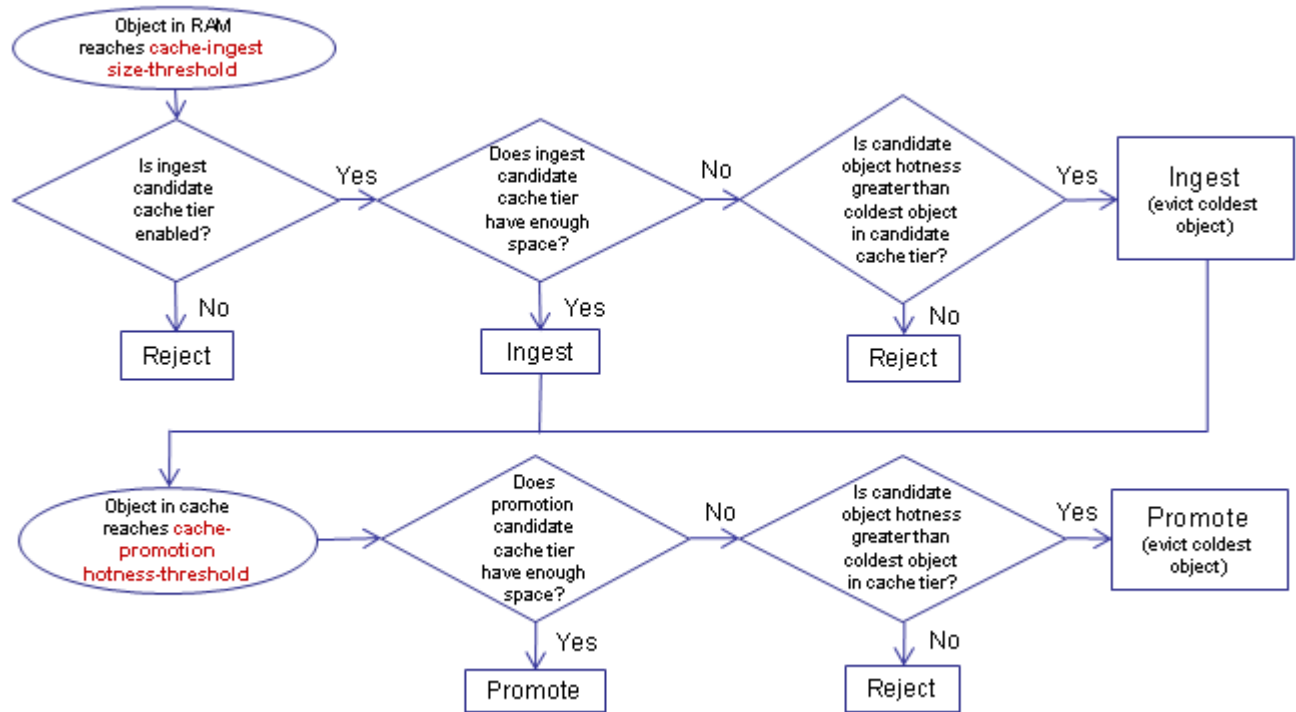


Figure 3 Media Flow Controller Cache Ingest and Promotion Process

Factors Influencing the Cache-ability of an Object

The following is a list of common conditions under which a request or response is *not* cached by Media Flow Controller:

- Origin server returns neither 200 nor 206 response code.
- Request URL has query string and “disable cache on query string” is enabled under **namespace** configuration.
- Origin server returns a 206 response and the content is marked as “chunked encoding.”
- Origin server does not return content-length, or “chunked encoding” is not present in the response.
- Origin server returns a content-length with value 0 (zero).
- Request URL contains hexadecimal characters.
- When the calculation of the object version fails. In order to determine whether a response object is the same as a cached object or not, Media Flow Controller adds version control to each response object. Every object version is calculated based on domain, URI, ETag, Date, and Last-Modified-Date fields from the HTTP transaction. If object versions are different, the object has been modified and is not cached.
- “Cache-control: no-cache” header is present in the request or response.
- Content-Range request and response do not match the offset or length of the object.

- Request or response contains the following headers:**
 - Cache-Control: Private
 - Cache-Control: no-cache
 - Cache-Control: No-Store
 - Cache-Control: Max-Age = 0
 - Pragma: No-Cache
- Object is expired (for example, an object's Expiry time is less than Media Flow Controller's current system time, or an object's Age has exceeded the configured Age time).**
- Response contains cookies (Set-Cookie/Set-Cookie2) and "disable cookie caching" is enabled in **namespace** configuration.**
- Analytics manager decides that the object is not worthy of caching because the size is below that configured by the **namespace** command **delivery protocol http origin-fetch content-store media object-size <size>**.

** Can be overridden by using the **namespace** configuration to forcefully cache objects that are marked as "no-cache."

When Media Flow Controller decides that a request or response is non-cacheable, it tunnels the response through. The response is not stored in the cache file system.

The response is cached when any of the preceding conditions are not encountered.

Connection Pooling

Media Flow Controller fetches content from an origin server, if content is not present in the cache. While fetching content from origin, Media Flow Controller can minimize the load on origin servers by pooling TCP connections. Media Flow Controller establishes fixed number of connections with the origin server and re-uses them for new requests. This minimizes the connection setup overhead and also reduces the load on origin servers. Connection pooling is beneficial when Media Flow Controller is deployed as an origin accelerator.

Origin Clustering and Origin Escalation

Media Flow Controller supports **origin-server** node map configuration for origin escalation: if the target origin server fails, another configured origin server is automatically chosen. These configurations are achieved through the creation of a **server-map (format-type cluster-map** and **format-type origin-escalation-map)** that is then associated with a **namespace**.

For more details, see ["Origin Clustering Using Media Flow Controllers" on page 162](#).

Media Flow Controller AssuredFlow

The AssuredFlow feature ensures that Media Flow Controller provides the required bandwidth for a connection so that media encoded at different bit-rates are delivered at approximately the encoded bit-rate rather than the fastest possible. This helps optimize use of the available bandwidth per session along with contributing to the viewing experience of the end user.

It also may be tied to the end user's service level agreement (SLA). This ensures that bandwidth is not wasted by sending data at a rate higher than the rate at which it is being

consumed (decoded) by the client. Furthermore, it ensures that sufficient bandwidth is available (reserved) for clients that need higher bit-rate video. Examples include:

- Full-screen mode clients (higher bit-rate) versus small window client (lower bit rate).
- Premium content (higher bit-rate) versus free content (lower bit rate).
- Content delivery to a primary site visitor with higher bit-rate versus content delivered to a viewer redirected from a partner site.

Assured-flow rate (AFR) is the parameter through which Media Flow Controller provides customer control of the AssuredFlow feature. AFR is specified in Kbps, and its intent is to ensure that Media Flow Controller reserves at least the configured rate in bandwidth for each delivery session. Clearly, the sum of AFR cannot exceed the aggregate bandwidth of the server. To be more specific, if an interface, say, Gigabit Ethernet, has “n” sessions, AssuredFlow can guarantee that the sum of AFR assigned to each active session does not exceed the capacity of the Gigabit Ethernet port, or 1 Gbps. An active session is one that is sending data at any one particular instance. We recommend that the sum of AFR be configured to 80 to 90 percent of the link speed, for best performance. Further, the configured AFR should reflect the average bandwidth the target origin server is set to deliver media. For example, if a portal delivers video to users at an average rate of 750 Kbps, AFR should be configured to reflect this value (for example, 750 Kbps). Assured-flow rate can be configured globally or through a virtual player configuration. Traffic is served at the configured AFR, or the dynamic AFR set by the virtual player, up to the configured maximum session bandwidth.

It is not uncommon to have portals set the logic in their player to signal the AFR on each session. In that case, the signaled AFR overrides the configured AFR. AFR is disabled by default, which means Media Flow Controller does not assure a delivery rate. When AFR is disabled in Media Flow Controller, player-signaled AFR is still effective.

Session Admission Control

Session admission control provides a mechanism to avoid bandwidth overload. Before a new session is admitted, a series of checks across various resources determines whether the session can be admitted. A new session is defined as the first GET request received within a new network connection. Existing sessions are not subject to this control. The following are the various checks that can reject a new session:

1. A new connection is rejected if the incoming interface is already serving at its bandwidth limit.
2. After a new connection is accepted, the first GET request can be rejected (with an HTTP error code) during various stages of processing:
 - a. The AFR for the request is based on the URL along with other query parameters and Media Flow Controller can determine that this AFR cannot be serviced given the existing bandwidth being served on that port.
 - b. Media Flow Controller can refuse to create the delivery task if it is out of processing resources.
 - c. Media Flow Controller can return an error if it is out of memory resources.
 - d. Media Flow Controller can return an error if meeting this request would exceed the capacity of the internal caches or origin libraries.

Logging and statistics for which Media Flow Controller module refused session admission are provided in the errorlog.

Media Flow Controller SmoothFlow

SmoothFlow™ refers to the Quality of Experience (QoE) feature that Juniper Networks Media Flow Controller can provide to viewers for uninterrupted video viewing.

Last-mile bandwidth fluctuations can cause buffering, or long pauses. Juniper Networks SmoothFlow technology provides viewers a TV-like video viewing experience irrespective of last-mile bandwidth fluctuations, by dynamically detecting available bandwidth and seamlessly switching the bit-rate of a video being progressively downloaded over HTTP.

Viewers with high bandwidth connections receive videos at higher quality resolutions while viewers with lower bandwidth connections receive videos encoded at bit-rates matching their available bandwidth. Media Flow Controller always sends video data at the bit-rate that is appropriate to the available bandwidth between the server and client at any point of time.

Media Flow Controller SmoothFlow receives client-side signals from the client player providing information about real-time resource utilization (for example, if the viewer starts a CPU-intensive application while watching a video).

In addition to SmoothFlow, such signals enable Media Flow Controller to allow viewers to control their media playback experience using flow commands such as fast forward, rewind, frame step, pause, and so forth, on a video that is currently being downloaded.

Media Flow Controller server-side intelligence, coupled with player feedback, allows Juniper Networks Media Flow Controller to deliver a really high quality of viewing experience tailored specifically to each viewer. [Figure 4](#), illustrates a SmoothFlow deployment. See [Chapter 8, "SmoothFlow Deployment,"](#) for more information.

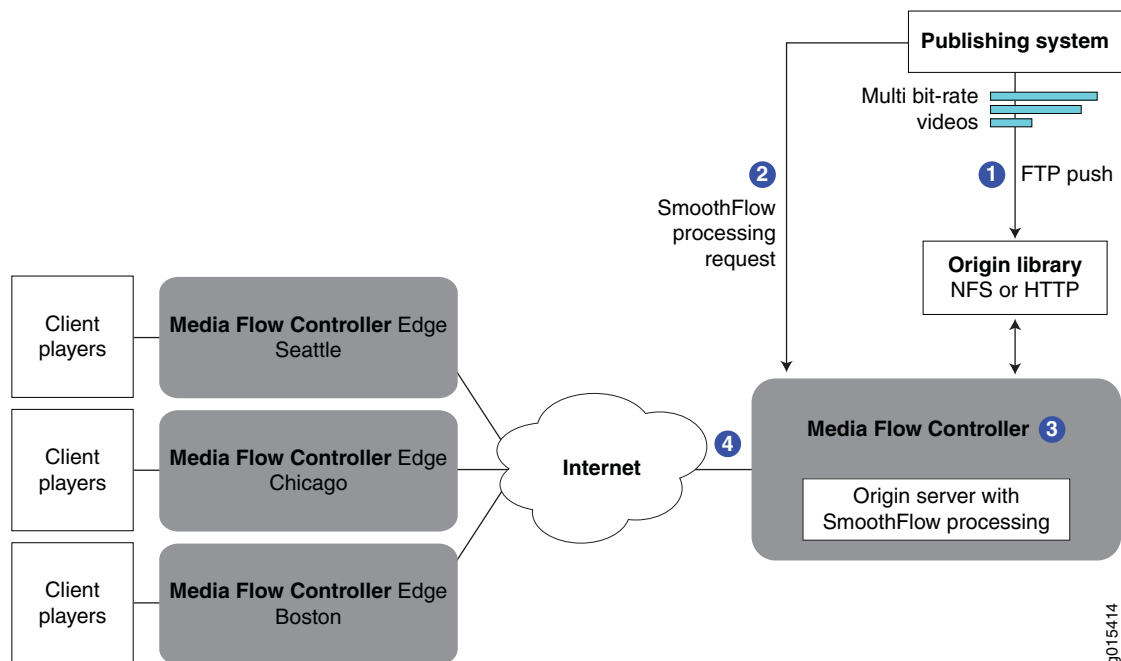


Figure 4 SmoothFlow™ Deployment Overview

g015414

How SmoothFlow Works

SmoothFlow is based on a dual channel approach where one channel is used for video delivery, and the other for control purposes to signal to Media Flow Controller adaptation points for responding to bandwidth fluctuations. SmoothFlow technology is delivered via progressive download over HTTP, providing the additional benefit that the consumer does not need to download a custom player. As shown in [Figure 4](#):

1. Content providers decide how many bit-rate profiles (differently encoded versions) of any one video they want to create. Each set of encoded bit-rate profiles must include a metadata file describing how many bit-rate profiles a video has and where they are stored; this file may be created by the provider, or auto-created, depending on the encoding procedure used. Together, the bit-rate profiles and the description file are the “asset.” After the assets are created, they are pre-staged to an origin server, typically via FTP.
2. Next the publisher or encoder sends a properly configured Media Flow Controller a SmoothFlow processing request for the asset. This may be done with a script, depending on the encoding procedure used.
3. SmoothFlow reads the data file given in the processing request, obtains the video files, and processes them for delivery; this includes chunking the different bit-rate profiles, creating the internal metadata file for Media Flow Controller, and queuing the assets on the origin server.
4. Assets are delivered to the edge either upon cache miss or via a SmoothFlow virtual player. As the assets are delivered to the client players over the delivery channel, feedback hints to SmoothFlow are sent over the control channel telling Media Flow Controller when to switch to a different bit-rate profile.

Media Flow Controller Namespaces

The **namespace** function allows you to classify different types of traffic based on a combination of URL and FQDN, and apply separate delivery policies to each type of classified traffic. This gives you a way to separate your video-delivery traffic characteristics based on any given variable in the stream/request being received by Media Flow Controller from the client. You can create up to 256 namespaces in one Media Flow Controller

At a minimum, **namespace** configuration requires a **domain** (only one per namespace), an **origin-server** (one per namespace unless using **server-map**), and a **match** criteria (to refine delivery of incoming requests). Additional parameters for **origin-fetch**, **cache** options, and so forth, are optional. You can further define control by assigning a configured **virtual-player**. The namespace is referenced via the URL in the HTTP request to Media Flow Controller.

A namespace usually represents one or more Web properties or websites. A CDN service provider may define one or more namespaces for a given CDN customer. For example, a CDN customer may have multiple domains such as `www.foo.com`, `www.bar.com`, and `www.foobar.com`. If the media caching and delivery policies are the same across all the three properties, a CDN service provider can just define one namespace for the CDN customer. If the media caching and delivery policies are different for the three Web properties, then separate namespaces have to be defined in the Media Flow Controller.

A content publisher may define one or more namespaces to represent a website. For example, a content publisher may serve videos and Web objects from the same website. However, the delivery protocols or policies for videos may be different from the delivery

protocols or policies for the Web objects. In such a scenario, a content publisher defines separate namespaces for videos and Web objects delivery.

Media Flow Controller Virtual Players

The virtual player function contains a set of policies that allow you to control the delivery of media. Virtual player provides an infrastructure to perform the following functions:

- Uniquely identify objects in the cache, though the objects may be represented using different URLs
- Provide trick-play functions such as forward, rewind, seek, and pause
- Guarantee bandwidth for each user session
- Validate URLs before serving user requests

You can create any number of virtual players; they are utilized when assigned to a namespace. Namespaces that are not assigned a **virtual-player** use the values configured under **network connection** for the same functions. Virtual players let you implement custom delivery policies.

CHAPTER 3

Media Flow Controller Deployment Guidelines

- [Media Flow Controller Deployments Overview](#)
- [Reverse Proxy Deployments](#)
- [Transparent Proxy Deployments](#)
- [Mid-Tier Proxy Deployments](#)

Media Flow Controller Deployments Overview

Media Flow Controller can be deployed by content publishers, Internet service providers, and content delivery networks (CDNs) for origin acceleration and edge caching as a reverse proxy, transparent proxy or mid-tier proxy.

Table 4 Transparent Proxy versus Reverse Proxy

| Transparent Proxy | Reverse Proxy |
|---|---|
| Requests redirected to the cache using policy-based routing. | Requests redirected to the cache using DNS-based routing. |
| Typically deployed at Internet service providers (ISPs) or campus edges only. | Typically deployed at origin, content publisher, or CDN edges. Can also be deployed at ISP or campus edges. |
| Requires policy changes at all the edge routers handling user requests. | Requires changes in the DNS server. |
| Does not require contracts with the content publisher. | May require contracts with the content publisher. |
| Server and client do not know there is a cache in-between. | Server and client know there is a cache in-between. |

Reverse Proxy Deployments

- [About Reverse Proxies](#)
- [Reverse Proxy Protocol Support](#)
- [Reverse Proxy Deployment Requirements](#)
- [Reverse Proxy Deployment Process](#)
- [Reverse Proxy Cache Tuning CLI Commands](#)
- [Reverse Proxy Namespace Examples](#)

About Reverse Proxies

In a reverse proxy deployment, the user's request is routed to Media Flow Controller using DNS-based redirection or by using an SLB/GSLB module or by using HTTP redirects.

Deploy Media Flow Controller as a reverse proxy when one or more of the following criteria are met:

- You want to do caching at the origin or edge
- You are a Content Publisher, CDN or a ISP/Campus Edge operator
- You do not want to change policies or configuration in the routers/switches
- You are ready to change DNS configuration
- You are fine with the user and origin knowing about the presence of a proxy/cache in-between

See [Figure 5](#) for a graphic depicting a reverse proxy call flow.

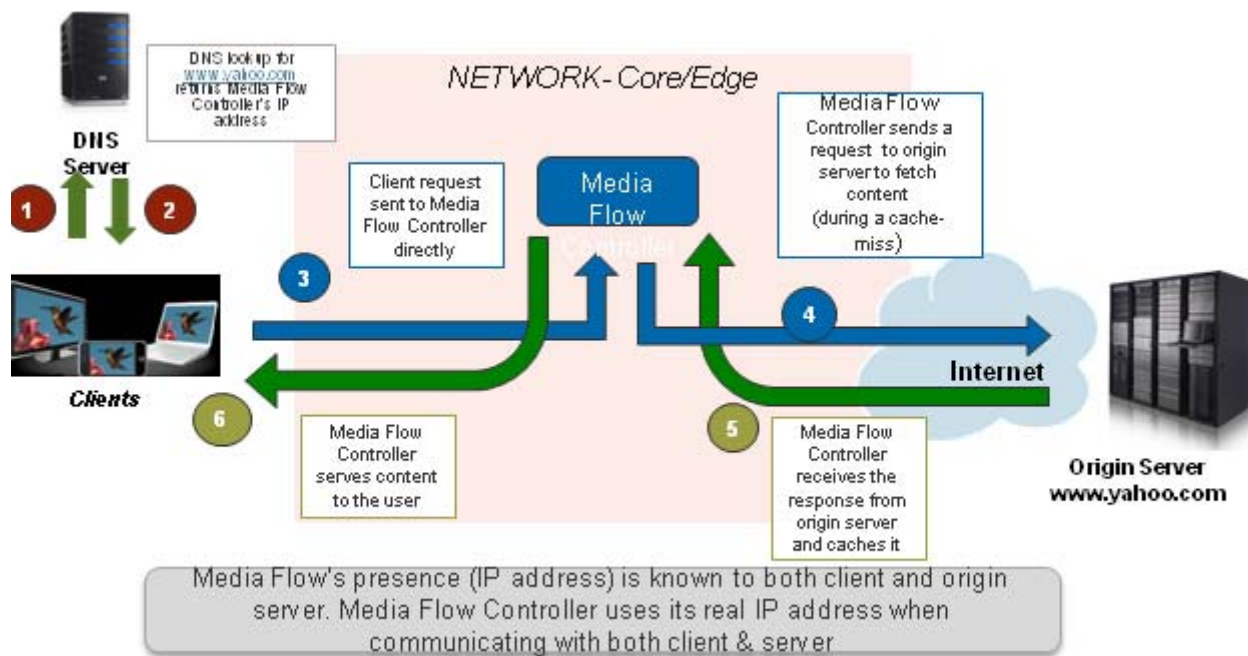


Figure 5 Reverse Proxy Call Flow

Reverse Proxy Protocol Support

Media Flow Controller supports content acquisition or publishing using HTTP, NFS, FTP, RTP/RTSP, and RTMP protocols. Media Flow Controller delivers media to users using the HTTP, RTP/RTSP, and RTMP protocols. Supported combination of protocols for media fetch and delivery are given in [Table 3 on page 42](#).

Reverse Proxy Deployment Requirements

You need the following information to get started with the configuration:

- **Type of delivery**—Live or on-demand video delivery, or Web content delivery
- **Delivery protocol**—HTTP, RTP/RTSP, or RTMP

- **Ingest Protocol**—HTTP, NFS, RTP/RTSP, or RTMP
- **Origin server details**—Hostname, IP address
- **Mapping request to origin policies**—The list of parameters in the URL (domain, URI, and query parameters) for caching and delivery policies

Reverse Proxy Deployment Process

We assume that the Media Flow Controller is configured with the basic settings such as license configuration, interface configuration, IP address assignment, hostname, domain list, system clock, DNS, IP routes, and default gateway configuration. Once the basic configurations are done and the required information gathered, you are ready to begin the deployment process, as outlined:

1. Create a namespace—A namespace contains a set of policies that influence how content is fetched from origin server, cached and delivered to users. You can create multiple namespaces if you want to apply separate set of policies for each of the properties, domains, or customers. Examples of reverse proxy namespace configurations for different deployments are given in [“Reverse Proxy Namespace Examples” on page 59](#).
2. Map requests to namespaces by configuring a **namespace domain** and **namespace match** criteria—See [Chapter 6, “Configuring Namespaces \(CLI\)”](#) for details on using namespaces.
3. Define origin server and ingest protocol settings—Configure origin server settings such as server hostname and IP address, and/or port number for origin fetch, for each **namespace**.

If you have multiple origin servers and you want to distribute load across different origin servers, or if you want configure origin server failover settings, use the **server-map** feature in Media Flow Controller. If you have Media Flow Controllers upstream (mid-tier), you can use the Cluster map type server map.

For information on using the **origin-server** option, see [Chapter 6, “Configuring Namespaces \(CLI\)”](#). For information on using the server-map options, see [Chapter 8, “Configuring Media Flow Controller Server Maps.”](#)

4. Define delivery protocols—Configure the protocol or protocols to deliver media to users; your options are HTTP, RTSP, or RTMP.
For information on for HTTP or RTSP delivery, see [“Configuring Media Flow Controller Delivery Protocols \(CLI\)” on page 102](#).
No namespace configuration is required for RTMP delivery. You can configure Media Flow Controller using the Adobe Flash Media Server (FMS) configuration files.
For information on using FMS for RTMP delivery, see [“Installing and Using FMS in Media Flow Controller \(CLI\)” on page 110](#).
5. Define type of delivery—No special configuration is required to enable on-demand delivery, or HTTP-based live delivery.
You must define a **namespace <name> live-pub-point** to configure settings related to RTSP live streaming. RTMP streaming must be configured using the Adobe Flash Media Server configuration files.
For information on configuring live-streaming, see [“Using namespace for Live Streaming Delivery Without Caching” on page 148](#) and [“Using namespace for Live Streaming Delivery With Caching” on page 148](#).

6. Configure Media Flow Publisher (if applicable)—Configure Media Flow Publishing if you need to deliver on-demand/live videos using Apple HTTP Streaming, Microsoft SmoothStreaming, or Adobe Dynamic HTTP Streaming methods to QuickTime, Silverlight, and Flash players, respectively.
For information on Media Flow Publisher, see [Chapter 11, “Using Media Flow Controller Media Flow Publisher.”](#)
7. Create virtual-players (if applicable)—Virtual Player is the video requests processing engine that performs the following functions:
 - Acts like a media player when fetching content from the origin server
 - Acts like a media server when delivering content to users
 Virtual Players are typically used in a reverse proxy deployment to perform any of the following functions:
 - Rate-limit user sessions by setting an Assured Flow Rate (AFR)
 - Validate URLs by generating a URL hash before delivering video
 - Configure query, or video, “seek” related settings
 - Define how Media Flow Controller should uniquely identify objects in cache
 For information on for HTTP or RTSP delivery, see [“Configuring Virtual Players, Media Fetch and Pre-Staging \(CLI\)” on page 123.](#)
8. Tune cache configuration (if required)—Media Flow Controller’s default configuration is tuned for optimal caching and delivery performance. You can tune the cache configuration to maximize the performance for the given deployment. Refer to the following sections:
 - [“Setting Network Connection Options \(CLI\)” on page 100](#)—For tuning network connection-related parameters such as maximum bandwidth, connection idle timeout, and concurrent sessions.
 - [“Configuring Media Flow Controller Delivery Protocols \(CLI\)” on page 102](#)—For tuning request/response processing-related parameters such as overriding cache options, object size tuning, header insertion/deletion, and enabling connection pooling

Reverse Proxy Cache Tuning CLI Commands

See [Table 5](#) for guidelines on reverse proxy cache performance tuning.

Table 5 Cache Performance Tuning Settings for Reverse Proxy

| Config Parameter | Description | Benefit | Guidelines |
|---|---|--|---|
| namespace <name> delivery protocol {http rtsp} origin-fetch cache- age-default | Set the maximum age of objects when origin did not specify the maximum age of the object. (Default: 8 hours or 28800 secs) | Keeps the content in the cache for a longer duration, thereby improving cache-hit ratio. | Set this configuration when Media Flow Controller is fetching objects from NFS origins or when objects are published via FTP. The higher the cache-age value, the higher the cache-hit ratio will be (due to reduced churn of objects in cache). |

Table 5 Cache Performance Tuning Settings for Reverse Proxy (Continued)

| Config Parameter | Description | Benefit | Guidelines |
|---|--|--|--|
| namespace <name> delivery protocol http origin-fetch cache-directive no-cache follow | Do not cache the object when the origin server says so for example, when origin includes Cache-control: no-cache header. | Prevents dynamically generated or protected content from getting cached. | Set this configuration only in namespaces that serve dynamic content or protected content from origin servers. |
| namespace <name> delivery protocol http origin-fetch cache-revalidation permit | Performs automatic revalidation of the object in cache when it expires or when the object is close to become expired. | Improves cache-hit ratio, by just revalidating the expired object instead of fetching the entire object from origin. | Set this configuration to minimize transit bandwidth usage and to improve cache-hit ratio. |
| namespace <name> delivery protocol http origin-fetch cache-fill aggressive | Fetch the full object from the origin, even though the user requested only partial object. | Improves cache-hit ratio by fetching the entire object, instead of downloading just one chunk requested by the given user. | Set this configuration in namespaces that serve popular objects that are large (such as videos, install packages, and PDF files). |
| namespace <name> delivery protocol http origin-fetch content-store media cache-age-threshold | Short-lived objects are only stored in RAM cache. (Default: Objects of age under 60 seconds, stored in RAM.) | Improves disk performance by minimizing the churn in disk cache. | Keep the value of cache-age-threshold very low, to improve the disk performance. |
| namespace <name> delivery protocol http origin-fetch content-store media object-size | Store objects of given size and above in disk, irrespective of their size. (Default: 0, for example, all objects) | Improves cache-hit ratio by storing all objects in disk. | Increase the value of object-size, to force only large objects to be served from disk. |
| namespace <name> delivery protocol http origin-request connect retry-delay | Retry connection with origin server after a delay of given ms. (Default: 100 ms) | Improves connection management with origin when dealing with slow origin servers. | Increasing the value of retry-delay minimizes the origin server load. Decreasing the value of retry-delay improves the user experience by reducing the wait time for request retries. |
| namespace <name> delivery protocol http origin-request connect timeout | Timeout the connection request sent to origin, if there is no response from origin. (Default: 100 ms) | Improves connection management with origin when dealing with slow origin servers. | Increasing the connect timeout minimizes the load on origin server. Lowering the connect timeout improves the user experience by reducing the wait time for requests. |

Table 5 Cache Performance Tuning Settings for Reverse Proxy (Continued)

| Config Parameter | Description | Benefit | Guidelines |
|---|---|--|--|
| namespace <name> delivery protocol http origin-request host-header inherit incoming-req deny | Do not inherit "Host" header from the client request. | Distribution of load across origin servers, by not inheriting the Host header from client. | Disable this configuration if you have a load balancer to distribute requests across origin servers. Set this configuration only when you want to distribute requests across origins based on "Host" header. |
| namespace <name> delivery protocol http origin-request read interval-timeout | Timeout the read request if there is no response from origin for the given duration. (Default: 100 ms) | Improved throughput, by optimizing the network read operations. | Increasing the read timeout minimizes the load on origin server. Lowering the read timeout improves the user experience by reducing the wait time for requests. |
| namespace <name> delivery protocol http origin-request read retry- delay | Retry read from origin server, after a delay of given duration. (Default: 100 ms) | Improved throughput, by optimizing the network read operations. | Increasing the value of retry-delay minimizes the origin server load. Decreasing the value of retry-delay improves the user experience by reducing the wait time for request retries. |
| namespace <name> delivery protocol http origin-request x-forwarded- for enable | Include X-forwarded-for header with client IP address, in the request sent to origin. | Tracks the source of the request. | Enable this always in reverse proxy mode. |
| namespace <name> delivery protocol http client- request cache-control max- age 0 | Do not serve the client request with the object from cache. | Allows users to fetch content from origin, bypassing the cache. | Enable this always in reverse proxy mode. |
| namespace <name> delivery protocol http client- request cookie action cache | Cache objects though the request contains cookie header | Improved cache-hit ratio, by caching objects with cookies. | Disable this by default in reverse proxy mode. Set this configuration only when you are sure that the same object can be delivered to multiple users, by Media Flow Controller (irrespective of the user cookies). |

Table 5 Cache Performance Tuning Settings for Reverse Proxy (Continued)

| Config Parameter | Description | Benefit | Guidelines |
|---|--|--|---|
| namespace <name> delivery protocol http client- request query-string action no-cache | Do not cache object if the request URL contains query parameters. Presence of query parameters in the URL indicates that the response is dynamically generated, such as the output of CGI. | Improved cache-hit ratio, by not caching content generated by CGIs. | Enable this by default in reverse proxy mode. Disable this configuration only when you are sure that responses to requests with query-parameters can be cached. |
| media-cache disk group- read {sata sas ssd} {disable enable} | Media Flow Controller reads objects from the disk in large chunks (group read). This CLI enables or disables group reads. | Enabling group read, maximizes the disk performance, when caching or delivering large objects. | Disabled, if serving small objects (< 512KB). Enabled, if serving large objects (> 512KB). |

Reverse Proxy Namespace Examples

These configurations provide example configurations; variables are indicated with *italics*.

- [Example: HTTP In and HTTP Out](#)
- [Example: NFS In and HTTP Out](#)
- [Example: FTP In and HTTP Out](#)
- [Example: On-Demand RTSP In and RTSP Out](#)
- [Example: Live RTSP In and RTSP Out](#)
- [Example: Distribute Requests to Different Origin Servers Based on Host Header](#)
- [Example: Failover Across Origin Servers](#)
- [Example: Distribute Load Across Multiple NFS Origins](#)
- [Example: Deliver Streams Using Microsoft SmoothStreaming](#)
- [Example: Deliver Streams Using Adobe Dynamic HTTP Streaming](#)
- [Example: Deliver Streams Using Apple HTTP Streaming](#)
- [Example: Deliver Video Streams With Seek, URL Hash & Assured Flow](#)

Example: HTTP In and HTTP Out

- Type of delivery—On-demand (Web content delivery)
- Delivery protocol—HTTP
- Ingest protocol—HTTP
- Origin server details—web.example.com
- Mapping request to origin policies based on domain (example.com) and URI pattern (/web)
 - Namespace configuration:


```
namespace example
  domain example.com
```

```
match uri /web
origin server http web.example.com
delivery protocol http
status active
```

Example: NFS In and HTTP Out

- Type of delivery—On-demand (Web content delivery)
- Delivery protocol—HTTP
- Ingest protocol—NFS
- Origin server details—web.example.com
- Mapping request to origin policies based on domain (example.com) and URI pattern (/web)
 - Namespace configuration:

```
namespace example
  domain example.com
  match uri /web
  origin server nfs nfs.example.com:/var/www/
  delivery protocol http
  status active
```

Example: FTP In and HTTP Out

- Type of delivery—On-demand (Web content delivery)
- Delivery protocol—HTTP
- Ingest protocol—FTP
- Origin server details—web.example.com
- Mapping request to origin policies based on domain (example.com) and URI pattern (/web)
 - Namespace configuration:

```
namespace example
  domain example.com
  match uri /web
  pre-stage ftp user joe password foobar
  delivery protocol http
  status active
```

Example: On-Demand RTSP In and RTSP Out

- Type of delivery—On-demand (video stream delivery)
- Delivery protocol—RTSP
- Ingest protocol—RTSP
- Origin server details—rtsp.example.com
- Mapping request to origin policies based on domain (example.com) and URI pattern (/video)

- Namespace configuration:

```
namespace example
  domain example.com
  match uri /video
  origin server rtsp rtsp.example.com rtp-rtsp
  delivery protocol rtsp
  status active
```

Example: Live RTSP In and RTSP Out

- Type of delivery—Live (video stream delivery)
- Delivery protocol—RTSP
- Ingest protocol—RTSP
- Origin server details—rtsp.example.com
- Mapping request to origin policies based on domain (example.com) and URI pattern (/livepub)
- Namespace configuration:

```
namespace example
  domain example.com
  match uri /livepub
  origin server rtsp rtsp.example.com rtp-rtsp
  delivery protocol rtsp
  status active
  live-pub-point cusid1-office-allhands
    receive-mode on-demand
  status active
```

Example: Distribute Requests to Different Origin Servers Based on Host Header

- Type of delivery—On-demand (Web content delivery)
- Delivery protocol—HTTP
- Ingest protocol—HTTP
- Origin server details—host1.example.com, host2.example.com, host3.example.com
- Mapping request to origin policies based on domain (example.com) and URI pattern (/web)—Host header in the HTTP request contains values such as web.example.com, images.example.com, and video.example.com
- Namespace configuration:

```
namespace example
  domain example.com
  match uri /web
  origin server http server-map OriginServerDefinitions
  delivery protocol http
  status active
```

- Server map configuration:

```
server-map OriginServerDefinitions
  format-type host-origin-map
  file-url http://example.com/serverdefinitions.xml refresh-interval 60
```

Contents of serverdefinitions.xml:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE HostOriginMap SYSTEM "HostOriginMap.dtd">
<HostOriginMap>
  <Header>
    <Version>1.0</Version>
    <Application>MapXML</Application>
  </Header>
  <HostOriginEntry>
    <Host>web.example.com</Host>
    <Origin>host1.example.com</Origin>
    <Port>80</Port>
  </HostOriginEntry>
  <HostOriginEntry>
    <Host>images.example.com</Host>
    <Origin>host2.example.com</Origin>
    <Port>80</Port>
  </HostOriginEntry>
  <HostOriginEntry>
    <Host>video.example.com</Host>
    <Origin>host3.example.com</Origin>
    <Port>80</Port>
  </HostOriginEntry>
</HostOriginMap>
```

Example: Failover Across Origin Servers

- Type of delivery—On-demand (Web content delivery)
- Delivery protocol—HTTP
- Ingest protocol—HTTP
- Origin server details—host1.example.com, host2.example.com, host3.example.com
- Mapping request to origin policies based on domain (example.com) and URI pattern (/web):
 - Media Flow Controller sends requests to host1.example.com, by default
 - Media Flow Controller fails over to host2.example.com, when host1.example.com returns 404 or 500 error
 - Media Flow Controller fails over to host3.example.com, when host1.example.com and host2.example.com return 404 or 500 error
 - Media Flow Controller sends constant HTTP probes to the configured “heartbeatpath” in the origin server
 - Namespace configuration:

```
namespace example
  domain example.com
  match uri /web
  origin server http server-map OriginServerDefinitions
  delivery protocol http
  status active
```

- Server map configuration:

```
server-map OriginServerDefinitions
  format-type origin-escalation-map
  file-url http://example.com/serverdefinitions.xml refresh-interval 60
  node-monitoring heartbeat
    allowed-fails 3
    connect-timeout 100
    interval 100
    read-timeout 100
```

Contents of serverdefinitions.xml:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE OriginEscalationMap SYSTEM "OriginEscalationMap.dtd">
<OriginEscalationMap>
<Header>
<Version>1.0</Version>
<Application>MapXML</Application>
</Header>
  <OriginEscalationMapEntry>
    <Origin>host1.example.com</Origin>
    <Port>80</Port>
    <Options>heartbeatpath=/hello.html,weight=1,
      http_response_failure_codes=404;500</Options>
  </OriginEscalationMapEntry>
  <OriginEscalationMapEntry>
    <Origin> host2.example.com </Origin>
    <Port>80</Port>
    <Options>heartbeatpath=/hello.html,weight=2,
      http_response_failure_codes=404;500</Options>
  </OriginEscalationMapEntry>
  <OriginEscalationMapEntry>
    <Origin>host1.example.com</Origin>
    <Port>80</Port>
    <Options>heartbeatpath=/hello.html,weight=3,
      http_response_failure_codes=404;500;503</Options>
  </OriginEscalationMapEntry>
</OriginEscalationMap>
```

Example: Distribute Load Across Multiple NFS Origins

- Type of delivery—On-demand (Web content delivery)
- Delivery protocol—HTTP
- Ingest protocol—NFS
- Origin server details (NFS path):
 - \\host1.example.com\html
 - \\host2.example.com\images
 - \\host3.example.com\videos

- Mapping request to origin policies based on domain (example.com) and URI pattern (/web).

- Namespace configuration:

```
namespace example
  domain example.com
  match uri /web
  origin server nfs server-map OriginServerDefinitions
  delivery protocol http
  status active
```

- Server map configuration:

```
server-map OriginServerDefinitions
  format-type nfs-map
  file-url http://example.com/serverdefinitions.xml refresh-interval 60
```

Contents of serverdefinitions.xml:

```
<response scode="0" scode_description="success!">
<PUBLISHINGPOINTS INTERVAL-SEC="3600">
<MISSINGFILE PATH=" "/>
<INVALIDPUBLISHINGPOINT PATH=" "/>
<PUBLISHINGPOINT NAME="name1" PATH="\\host1.example.com\html"/>
<PUBLISHINGPOINT NAME="name2" PATH="\\host2.example.com\images"/>
<PUBLISHINGPOINT NAME="name3" PATH="\\host3.example.com\videos"/>
</PUBLISHINGPOINTS>
</response>
```

Example: Deliver Streams Using Microsoft SmoothStreaming

- Type of delivery—On-demand (video delivery)
- Delivery protocol—HTTP
- Ingest protocol—NFS
- Origin server details—nfs.example.com
- Mapping request to origin policies based on domain (example.com) and URI pattern (/video)

- Sample URL:

```
http://www.example.com/videos/bunny.ism/QualityLevels(400000)/
  Fragments(video=134345672)
```

- Virtual player configuration:

```
virtual-player SmoothStream type smoothstream-pub
  fragment-tag Fragments
  quality-tag QualityLevels
```

- Namespace configuration:

```
namespace example
  domain example.com
  match uri /video
  origin server nfs nfs.example.com:/videos/
```



```

delivery protocol http
status active
virtual-player SmoothStream

```



NOTE: You configure Media Flow Publisher with the Web interface.

Example: Deliver Streams Using Adobe Dynamic HTTP Streaming

- Type of delivery—On-demand (video delivery)
- Delivery protocol—HTTP
- Ingest protocol—NFS
- Origin server details—nfs.example.com
- Mapping request to origin policies based on domain (example.com) and URI pattern (/video)

- Sample URL:

```
http://www.example.com/videos/foo1000Seg1-Frag4
```

- Virtual player configuration:

```

virtual-player FlashStream type flashstream-pub
  fragment-tag Frag
  segment-tag Seg
  seg-frag-delimiter -

```

- Namespace configuration:

```

namespace example
  domain example.com
  match uri /video
  origin server nfs nfs.example.com:/videos/
  delivery protocol http
  status active
  virtual-player FlashStream

```

Example: Deliver Streams Using Apple HTTP Streaming

No virtual player configuration is required other than the standard namespace configuration settings given in the previous sections.

Example: Deliver Video Streams With Seek, URL Hash & Assured Flow

- Type of delivery—On-demand (video delivery)
- Delivery protocol—HTTP
- Ingest protocol—NFS
- Origin server details—web.example.com
- Mapping request to origin policies based on domain (example.com) and URI pattern (/media)

- Sample URL:

```
http://www.example.com/media/
foo.flv?e=3312665958&h=ec41f550878f45d9724776761d6ac416&seek=150
```
- Virtual player configuration:
 - Assured bandwidth of 256 Kbps for each user session
 - Time offset-based seek
 - URL is rejected when the computed hash expires (for example, when the incoming URL request has exceeded the Media Flow Controller's system time)

```
virtual-player player type generic
full-download always
assured-flow rate 256
hash-verify expiry-time-verify query-string-param h
seek-config query-string-param seek
seek-mp4-type time-msec
```

- Namespace configuration:

```
namespace example
domain example.com
match uri /media
origin server nfs nfs.example.com:/videos/
delivery protocol http
status active
virtual-player player
```

Transparent Proxy Deployments

- [About Transparent Proxies](#)
- [Upgrading for New Transparent Proxy Functions](#)
- [Transparent Proxy Example Configuration—General](#)
- [Transparent Proxy Example Configuration—YouTube](#)
- [Transparent Proxy Cache Tuning CLI Commands](#)
- [DMCA Compliance Transparent Proxy Configuration Requirements](#)
- [Example: DMCA Compliance Transparent Proxy Configuration](#)
- [Transparent Proxy Cache Tuning Examples](#)

About Transparent Proxies

A transparent proxy, or “Content Direct,” solution saves transit bandwidth for network service providers (NSP) and improves subscriber’s online experience, through faster downloads. Content Direct saves bandwidth for NSPs, by delivering repeat requests to content from Media Flow Controller. Bandwidth savings achieved using Content Direct varies significantly based on factors such as storage capacity of the Media Flow Controllers, size of objects that get cached (small versus large), type of objects accessed by users (static versus dynamic, cacheable versus non-cacheable), amount of popular content, and number of users accessing

content. Deploy Media Flow Controller as a transparent proxy when one or more of the following criteria are met:

- You do not want users or the origin servers to know about the presence of a proxy/cache
- You want to do caching at the edge
- You are an ISP or Campus Edge operator
- You do not want users to modify their browser configuration to point to a proxy
- You do not want to change the DNS configuration
- You are ready for policy-based routing at edge

See [Figure 6](#) for a graphic depicting a transparent proxy call flow

i **NOTE:** We recommend the most recent Media Flow Controller Release 2.0.x for transparent proxy deployments. These releases contain enhanced transparent proxy capabilities.

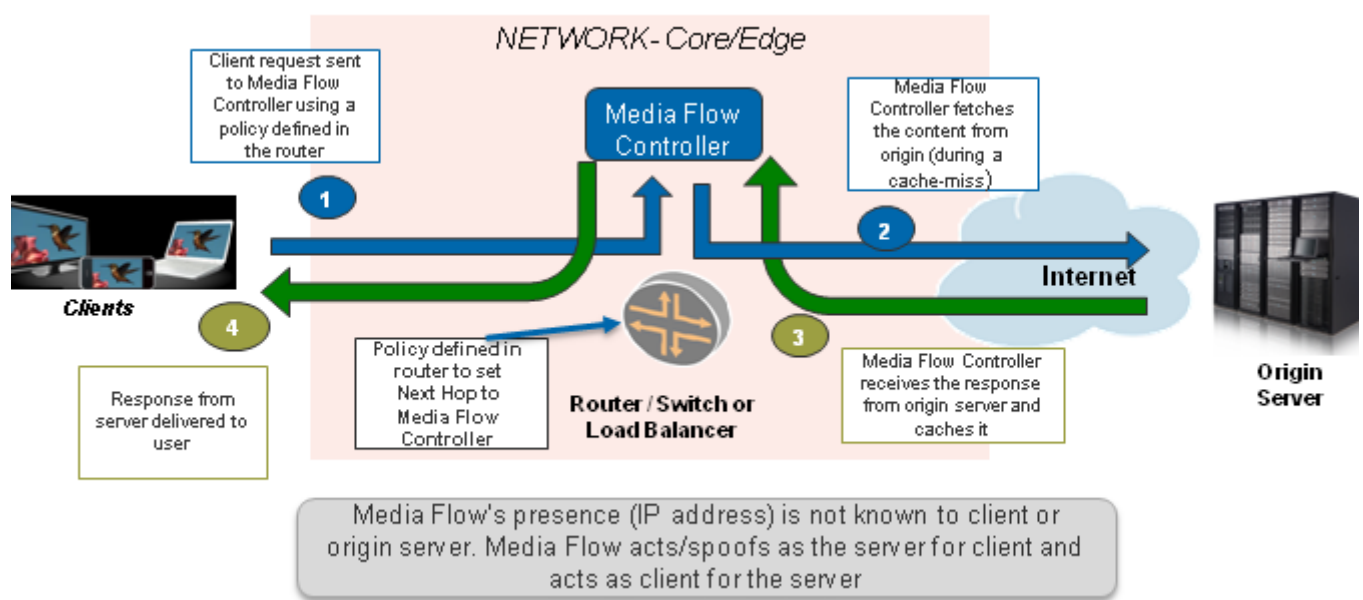


Figure 6 Transparent Proxy Call Flow

Transparent Proxy Deployment Requirements

You need the following information to get started with the configuration:

- **Type of delivery**—Live or on-demand video delivery, or Web content delivery.
- **Delivery protocol**—HTTP is cached, everything else is tunneled.
- **Ingest Protocol**—HTTP only.
- **Origin server details**—Fully Qualified Domain Name (FQDN), IP address.
- **Mapping request to origin policies**—The list of parameters in the URL (domain, URI, and query parameters) for caching and delivery policies.

To achieve maximum performance, do not use the same device as transparent and reverse proxy. The same Media Flow Controller code base is used for both; however transparent proxy is currently limited by the number of client and origin connections rather than throughput. Media Flow Controller currently can support 100,000 subscriber connections. If the number of required connections for a transparent proxy is close to the limit, use separate reverse and transparent proxy Media Flow Controllers.

A combination of reverse proxy and transparent proxy can be deployed together in the same machine as long as the connection and throughput capacity limits are not exceeded.

Transparent Proxy Deployment Process

We assume that the Media Flow Controller is configured with the basic settings such as license configuration, interface configuration, IP address assignment, hostname, domain list, system clock, DNS, IP routes, and default gateway configuration. Once the basic configurations are done and the required information gathered, you are ready to begin the deployment process, as outlined:

1. Create a **namespace**—A namespace contains a set of policies that influence how content is fetched from origin server, cached and delivered to users. You can create multiple namespaces if you want to apply separate set of policies for each of the properties, domains, or customers. Examples of transparent proxy configurations are given in [“Transparent Proxy Example Configuration—General” on page 70](#) and [“Transparent Proxy Example Configuration—YouTube” on page 71](#).
2. Map requests to namespaces by configuring a **namespace domain** and **namespace match** criteria—See [Chapter 6, “Configuring Namespaces \(CLI\)”](#) for details on using namespaces.
3. Define origin server settings—For transparent proxy, you configure **namespace** options to derive the origin IP address from the request from the client. See [Table 6 on page 69](#) for details on the configurations and defaults for transparent proxy deployments.
4. No delivery protocols configuration is needed—Transparent proxy supports caching of HTTP only; it tunnels all other protocols.
5. Create virtual-players (if applicable)—The **virtual-player** is the Media Flow Controller video requests' processing engine that performs the following functions:
 - Acts like a media player when fetching content from the origin server.
 - Acts like a media server when delivering content to users.Virtual players in a transparent proxy are typically used to perform any of the following functions:
 - Configure query, or video, “seek” related settings.
 - Configure URI processing; see [“Using Virtual Player Type YouTube” on page 131](#) for details.
 - Define how Media Flow Controller should uniquely identify objects in cache.
6. Tune cache configuration (if required)—Media Flow Controller's default configuration is tuned for optimal caching and delivery performance. You can tune the cache configuration to maximize the performance for the given deployment. Refer to the following sections:
 - [“Transparent Proxy Cache Tuning CLI Commands” on page 74](#)—For tuning network connection-related parameters such as maximum bandwidth, connection idle timeout, and concurrent sessions.

- [“Transparent Proxy Cache Tuning Examples” on page 76](#)—For tuning request/response processing-related parameters such as overriding cache options, object size tuning, header insertion/deletion, and enabling connection pooling.

Table 6 Namespace **origin-server** Settings for Transparent Proxy Type*

| origin-server setting | Derive origin from the... | Source IP for Cache Miss | Destination IP for Cache Miss |
|--|----------------------------------|----------------------------------|--|
| <code>http follow header HOST</code> | HOST header value | Media Flow Controller IP address | DNS resolved IP address of the origin-server from the HOST header |
| <code>http follow header <name> use-client-ip</code> | Specified X-NKN or custom header | Client IP address | DNS resolved IP address of the origin-server from the given header |
| <code>http follow dest-ip</code> | Client request destination IP | Media Flow Controller IP address | No DNS resolution. Origin IP is client destination IP |
| <code>http follow dest-ip use-client-ip</code> | Client request destination IP | Client IP address | No DNS resolution. Origin IP is client destination IP |

*All transparent proxy **origin-server** settings described automatically set **origin-request host header inherit incoming-req permit**.

Upgrading for New Transparent Proxy Functions

This section describes various Media Flow Controller configurations for transparent proxy deployments in Media Flow Controller Release 2.0.7.



NOTE: All configurations require you be in configuration mode. You reach configuration mode with the commands **enable**, followed by **configure terminal**.

Media Flow Controller Release 2.1 provides many new command defaults and commands to facilitate and improve transparent proxy deployments. However, the new defaults are not automatically instantiated if you do an upgrade and retain your configurations. To instantiate the new defaults, you must either manually make the default changes, or revert to the factory configuration. If you have configurations you want to keep, you need to save them off-box first.

To save your current configuration off-box and upgrade to the new factory defaults without resetting the IP information, see [“Saving and Applying Configurations, Resetting Factory Defaults \(CLI\)” on page 118](#).

Transparent Proxy Example Configuration—General

This section describes configurations recommended for transparent proxy deployments.



NOTE: All transparent proxy configurations need to be customized to your specific pattern of traffic. You may need the assistance of Juniper Networks Support to fine-tune your cache performance.

NOTE: In Release 2.1 Media Flow Controller changed the namespace configuration requirements for transparent proxy deployments. The **namespace <name> proxy-mode** option is deprecated and no longer available. Two new options, **delivery protocol http transparent <client_traffic_NIC> enable**, and **namespace <name> origin-server http follow <option> use-client-ip bind-to <client_traffic_NIC>**, are used instead.

1. Set and enable delivery interfaces for transparent proxy:
 - a. Set HTTP delivery interfaces:
`delivery protocol http interface <client_traffic_NIC>`
 - b. Enable those interfaces for transparent proxy:
`delivery protocol http transparent <client_traffic_NIC> enable`
2. Create a namespace that inherits the host header from the client request when making the origin request for transparent proxy, disable setting the “X-Forwarded-For” header to the value of the client IP address when requests are sent from Media Flow Controller to origin upon a cache miss, and set **match**, **precedence**, and **origin-server** options.


```
namespace tproxy
  delivery protocol http origin-request host-header inherit incoming-req
  permit
  delivery protocol http origin-request x-forwarded-for disable
  match uri / precedence 10
  origin-server http follow header host use-client-ip bind-to <client_traffic_NIC>
```
3. Set an **object-size** limit on storing media objects for namespace **tproxy**; objects below this size go into memory, and never go to any disk or SSD. Also, set the **cache-fill** option so only requested data is fetched. From **namespace** prefix mode:


```
delivery protocol http origin-fetch content-store media object-size
  4096
  delivery protocol http origin-fetch cache-fill client-driven
```
4. Optionally, allow asynchronous, or non-blocking, Domain Name System (DNS) service. In a transparent proxy scenario, when this command is used, Media Flow Controller does the DNS resolution instead of the client doing it. This helps prevent cache poisoning.


```
network resolver asynchronous
```
5. Optionally, set a lower sync interval for hotness data in the cache; default is **14400** seconds. This value should be decreased for transparent proxy caching because the number of objects and the rate of change of those objects is much higher than in the reverse proxy mode.


```
ram-cache sync interval <integer>
```
6. Optionally, disable group read operations. In transparent proxy deployments, most often, Internet objects are small, so disabling group reads improves Media Flow Controller cache performance. By default, **group-read** is disabled for SSD, and enabled for SAS and

SATA; SSD drives support much higher IO (input/output) transaction loads than standard SAS or SATA drives.

```
media-cache disk group-read sata disable
media-cache disk group-read sas disable
```

- Optionally, define the maximum allowed number of TIME_WAIT sockets held by the system simultaneously; this can be useful when running non-persistent connections or performance tests on transparent proxy configurations.

```
network tcp max-tw-buckets <number>
```

- Activate and then exit namespace **tpoxy** and save. From **namespace** prefix mode:

```
status active
exit
write memory
```

Transparent Proxy Example Configuration—YouTube

This section describes configurations found to be beneficial for transparent proxy deployments for YouTube media. You can find additional configurations and information at [“Configuring YouTube Video Caching \(CLI\)” on page 131](#).

- If you install with root cache; disable it (disk **dc_1**) because this is the root disk and Media Flow Controller does a lot of logging on that disk—it shouldn't be used for caching.

```
media-cache disk dc_1 status inactive
```

- Disable cache promotion for the following reasons:

- Hotness promotion algorithm keeps track of only 200k objects. At 1Gbps, we typically see 2.4M objects per day, so 200k is not enough.
- Promotion does not take size into account, so a large object could be promoted and cause a large number of small objects to be evicted from SSD.

```
analytics cache-promotion disable
```

- Disable group read. Group read is used for objects with high locality, which is seldom the case for transparent proxy. Group read is disabled for SSD by default.

```
media-cache disk group-read sas disable
media-cache disk group-read sata disable
```

- Set a cache ingest size threshold; this, combined with a namespace configuration of **object-size 4096** (given in step 12), means that:

```
objects with size with >64K go to SAS
objects with size with >4K and <=64K go to SSD
objects with size with <4K go to RAM
analytics cache-ingest size-threshold 65536
```

- Ensure that the RAM cache size setting is zero to allocate 22 to 23 Gigabyte to RAM in a system with 36 Gigabyte RAM. The default is **0** (zero).

```
ram-cache cache-size-MB 0
```

- Set the queue sizes per thread for each tier. There are four threads per disk. If more than the following number of requests are queued in a disk tier, the object is re-fetched. Default is set at **0** (zero), which means an infinite number of requests can be queued.

```
media-cache disk cache-tier sas admission threshold 20
media-cache disk cache-tier sata admission threshold 12
media-cache disk cache-tier ssd admission threshold 1250
```

7. Set and enable delivery interfaces for transparent proxy:
 - a. Set HTTP delivery interfaces:
`delivery protocol http interface <client_traffic_NIC>`
 - b. Enable those interfaces for transparent proxy:
`delivery protocol http transparent <client_traffic_NIC> enable`
8. To tunnel YouTube seeks, use this **virtual-player** configuration:


```
virtual-player youtube_player type youtube
  cache-name video-id query-string-param "id" format-tag query-string-
    parm "itag"
  exit
```
9. To cache YouTube traffic, use this **namespace** configuration; first, create the namespace with **domain**, **match**, and **origin-server** values; and your **virtual-player**:


```
namespace youtube
  domain regex "^.*\.c\.youtube\.com|^.*\.googlevideo\.com"
  match uri /videoplayback precedence 5
  origin-server http follow header host use-client-ip bind-to <client_traffic_NIC>
  virtual-player youtube_player
```
10. Set the namespace to override cache-control headers, inherit the host header from the client request when making the origin request for transparent proxy, and disable setting the "X-Forwarded-For" header to the value of the client IP address when requests are sent from Media Flow Controller to origin upon a cache miss. From **namespace** prefix mode:


```
delivery protocol http origin-fetch content-store media cache-age-
  threshold 300
  delivery protocol http origin-request host-header inherit incoming-req
  permit
  delivery protocol http origin-request x-forwarded-for disable
```
11. Because YouTube content comes with a "Cache-Control: Private" header, you must set namespace **youtube** to disregard all cache-control headers. From **namespace** prefix mode:


```
delivery protocol http origin-fetch cache-directive no-cache override
```
12. Set an **object-size** limit on storing media objects for namespace **youtube**; objects below this size go into memory, and never to any disk or SSD. This setting may not matter for YouTube since all objects are typically greater in size. From **namespace** prefix mode:


```
delivery protocol http origin-fetch content-store media object-size
  4096
```
13. Allow namespace **youtube** to stop downloading an object after the client stops viewing it. If an object is partially cached, then on a second subscriber request, the remainder object is downloaded via a byte range request. If the origin doesn't support byte-range requests, it sends the whole object and Media Flow Controller discards the part that has already been stored. Once done, exit namespace **youtube** and save. From **namespace** prefix mode:


```
delivery protocol http origin-fetch cache-fill client-driven
```
14. Add the **virtual-player** you configured; activate the namespace, exit, and save your configuration. From **namespace** prefix mode:


```
virtual-player youtube_player
  status active
  exit
write memory
```


DMCA Compliance Transparent Proxy Configuration Requirements

Table 7 shows Media Flow Controller compliance with the Digital Millennium Copyright Act (DMCA).

Table 7 DMCA Requirements to Media Flow Controller Functionality

| DMCA Service Provider Caching Requirements | Media Flow Compliance |
|---|--|
| Content cached must not be modified by the service provider. | Content is delivered as-is—not modified. |
| Provider must comply with rules of “refreshing.” | HTTP 1.1 compliant freshness, also known as expiry, management. |
| Transactions must not interfere with technology that tracks “hits.” | Media Flow Controller maintains complete transparency between client and origin and does not interfere with industry-standard technology to track “hit” information. |
| Limit user's access based on conditions set by the origin server. | HTTP 1.1 compliance ensures that content marked as “Private” is not cached. |
| Service provider must remove cached content when notified. | XML-API and CLI provide for content deletion and content invalidation. |

DMCA Compliance Transparent Proxy Configuration Details

Configuration details for DMCA compliance are:

1. Create a **namespace** that inherits the host header from the client request when making the origin request for transparent proxy. See Table 6 for examples.

```
origin-request host-header inherit incomping-req permit
```

This namespace setting, required for all transparent proxy configurations, tells Media Flow Controller to use the incoming header in the request to the origin server; this achieves full transparency to the client and the origin.
2. Disable setting the “X-Forwarded-For” header to the value of the client IP address when requests are sent from Media Flow Controller to origin upon a cache miss. This is done for full transparency.

```
origin-request x-forwarded-for disable
```
3. By default, all “Cache-Control” headers are followed by Media Flow Controller. For example, if a “Cache-Control” header says “No-Cache”, Media Flow Controller will not cache the content. No configuration is required.
4. By default, Media Flow Controller does not cache content that has query strings. Query strings are typically used for dynamically changing objects; therefore, Media Flow Controller does not cache query strings. No configuration is required.
5. Revalidate-always: For Media Flow Controller to make sure that the object is always ‘fresh’, it needs to be revalidated every time there is a HTTP request for that object.

```
client-request cache-hit action revalidate-always
```
6. By default, Media Flow Controller tunnels authenticated content. No configuration is required.

Example: DMCA Compliance Transparent Proxy Configuration

CLI configuration:

```
namespace tproxy
  delivery protocol http origin-request host-header inherit incoming-req
  permit
  delivery protocol http origin-request x-forwarded-for disable
  delivery protocol http client-request cache-hit action revalidate-always
  match uri / precedence 10
  origin-server http follow header host use-client-ip bind-to
  <client_traffic_NIC>
```

Transparent Proxy Cache Tuning CLI Commands

See [Table 8](#) for guidelines on cache performance tuning for transparent proxy deployments.

Table 8 Cache Performance Tuning Settings for Transparent Proxy

| Config Parameter | Description | Benefit | Guidelines |
|--|--|---|--|
| <code>analytics cache-ingest size-threshold 65536</code> | Set the maximum size of an object that can be optionally ingested into the fastest cache tier in the disk cache. Objects smaller than, or equal to, the configured size are automatically written to the fastest cache tier. The default, 0 , disallows any object being ingested directly to the fastest tier. | Allow objects to be ingested directly to the fastest cache tier. | This configuration means that: <ul style="list-style-type: none"> • Objects with size with >64K go to SAS • Objects with size with >4K and <=64K go to SSD • Objects with size with <4K go to RAM SSD has better performance for smaller objects than SAS, which handles larger objects better. |
| <code>analytics cache-promotion disable.</code> | Alternatively, in an environment where there is constant churn of objects in the cache, disable cache promotion entirely. Constant churn of cached objects may occur due to "short-lived" objects in the cache (due to shorter object expiry time, or highly diversified requests). | More accurate eviction processing so that cold objects do not waste disk space. | The hotness promotion algorithm keeps track of only 200k objects. At 1Gbps, we see 2.4M objects per day, so 200k is not enough. Promotion does not take size into account, so a large object could be promoted and cause a large number of small objects to be evicted from SSD. |

Table 8 Cache Performance Tuning Settings for Transparent Proxy (Continued)

| Config Parameter | Description | Benefit | Guidelines |
|--|--|--|--|
| <code>analytics cache-promotion hotness-threshold</code> | Set a threshold for object "hotness," after which an object is candidate for promotion to a faster cache tier. | Delaying objects from moving to slower cache tiers ensures that "hot" objects are served from the fastest cache tier, improving cache efficiency, and bandwidth savings. | By default, when an object is requested three times, it becomes a candidate for cache promotion. |
| <code>media-cache disk <dc_1> status inactive</code> | Disable the root disk (e.g. <code>dc_1</code>) and Media Flow Controller does a lot of logging on that disk—we recommend not using it for caching. | A transparent proxy accesslog utilizes more disk I/O and should have its own disk. | Recommended for transparent proxy deployments. |
| <code>media-cache disk group-read sas disable</code> <code>media-cache disk group-read sata disable</code> | Disable group read. Group read is used for objects with high locality, which is seldom the case for transparent proxy. Group read is disabled for SSD by default. | Better disk utilization. Group read tends to waste disk bandwidth if objects are not correlated. | Recommended for transparent proxy deployments. |
| <code>media-cache disk cache-tier sas admission threshold 20</code> <code>media-cache disk cache-tier sata admission threshold 12</code> <code>media-cache disk cache-tier ssa admission threshold 1250</code> | Set the queue sizes per thread for each tier. Default is set at 0 (zero), which means an infinite number of requests can be queued. | Better latency when large numbers of requests hit the disk. | There are four threads per disk. If more than the allowed number of requests are queued in a disk tier, the object is fetched from the origin server. However, setting it to these values has been shown to provide better disk utilization. |
| <code>namespace <name> delivery protocol http origin-fetch cache-directive no-cache override</code> | Override the "Cache-Control" header. The "Cache-Control" header influences the caching behavior of Media Flow Controller. By default, Media Flow Controller does not cache objects that are marked as "No-Cache" or "Private". | Achieve better bandwidth savings by caching YouTube videos. | This configuration is a violation of HTTP 1.1 standard and should be applied only for select Web portals whose media delivery methods are well understood by the customer. Required for caching YouTube videos. |

Table 8 Cache Performance Tuning Settings for Transparent Proxy (Continued)

| Config Parameter | Description | Benefit | Guidelines |
|---|--|---|--|
| <code>namespace <name> delivery protocol http origin-fetch cache-fill client- driven</code> | Allow the namespace to stop downloading an object after the client stops viewing it. If an object is partially cached, then, on a second subscriber request, the remainder object is downloaded via a byte range request. If the origin doesn't support byte-range requests, it sends the whole object and Media Flow Controller discards the part that has already been stored. | Optimize the number of bytes that are requested from the origin server, resulting in better bandwidth savings. | Recommended for transparent proxy deployments. |
| <code>namespace <name> delivery protocol http origin-fetch content-store media cache-age-threshold 300</code> | Increase the amount of time objects can be stored in RAM cache. Default: Objects of age under 60 seconds are stored in RAM.) | Save disk I/O bandwidth by storing objects in RAM for a longer period before moving to disk. | Recommended for transparent proxy deployments. |
| <code>namespace <name> delivery protocol http origin-fetch content-store media object-size 4096</code> | Also, set an object-size limit on storing media objects for namespace tproxy ; objects below this size go into memory, and never go to any disk or SSD. | Improve disk-cache performance since small objects need not be written into disk and can be served directly from the RAM cache. | Increase the value of object-size , to force only large objects to be served from disk. |
| <code>namespace <name> delivery protocol http origin-request host-header inherit incoming-req permit</code> | Set the namespace to use the "Host" header from the client request when making the origin request for transparent proxy. | The subscriber's client (the browser or another application) can specify the header to be used for requests to the origin. | Required for transparent proxy deployments. |
| <code>namespace <name> delivery protocol http origin-request x-forwarded-for disable</code> | Set the namespace to disable setting the "X-Forwarded-For" header to the value of the client IP address when requests are sent from Media Flow Controller to origin upon a cache miss. | To be completely transparent, the cache should forward requests as-is from the subscriber's client. | Required for transparent proxy deployments. |
| <code>ram-cache cache- size-MB 0</code> | Ensure that the RAM cache size setting is 0 (zero), the default, to allocate 22 to 23 Gigabyte to RAM in a system with 36 Gigabyte RAM. | Increases the number of TCP connections that Media Flow Controller can support. | Do not change this configuration from the default: 0 (zero). |

Transparent Proxy Cache Tuning Examples

This section provides some guidelines for fine tuning Media Flow Controller's configuration to achieve better bandwidth savings. You can improve bandwidth savings by tuning the following configurations in Media Flow Controller:

- ["Example: Virtual Player Tuning" on page 77](#)
- ["Example: Object Size Tuning" on page 77](#)

- [“Example: Mime-Type Tuning” on page 78](#)
- [“Example: Cache Age Tuning” on page 78](#)
- [“Example: Popularity Tuning” on page 78](#)
- [“Example: Cache-Control Override” on page 78](#)

Example: Virtual Player Tuning

In today's Internet, Web portals represent a single media asset using multiple URLs. A unique URL is generated for every access. Web portals do that for reasons such as prevention of content thefts, securing servers from DOS attacks, and prevention of browser/player caching. However, this poses a number of challenges to NSP edge caching solutions. A media asset may be identified by a query parameter in the URL, instead of the entire URL. Given below is an example of how YouTube generates URLs. The highlighted query parameters represent the uniqueness of a video delivered from YouTube.

```
GET "http://v8.nonxt7.c.YouTube.com/
videoplayback?ip=0.0.0.0&sparams=id%2Cexpire%2Cip%2Cipbits%2Citag%2Calgorith
m%2Cburst%2Cfactor&fexp=904405&algorithm=throttle
factor&itag=34&ipbits=0&burst=40&sver=3&expire=1266310800&key=yt1&signatu
re=66222E9350B9BB5AC68297F12AC1DCB4C53AAFDE.55B33FFFA04EBF001AF39A4F316E657F
C318E0E5&factor=1.25&id=efa3a0434887fdc0&redirect_counter=1"
```

Media Flow Controller uses the **virtual-player** configuration to uniquely identify objects, though the objects may be represented using different URLs, because **virtual-player** allows you to configure parameters in the URL to represent the object identity. For example, to enable YouTube caching, the following configuration (variables italicized) needs to be applied to the Media Flow Controller:

```
virtual-player ytplayerA type youtube
  cache-name video-id querystring-parm id format-id query-string-parm itag
  seek query-string-parm begin
```

You can define virtual players for popular websites such as YouTube, Rapidshare, Dailymotion, and Youko, so Media Flow Controller can cache objects delivered from those Web portals. For a list of complete websites that can be cached by Media Flow Controller and its corresponding **virtual-player** configuration, please contact cmbu-support@juniper.net.

Example: Object Size Tuning

Media Flow Controller stores media objects in a cache file-system that includes RAM-cache as well as disk-cache (SSD, SAS, and SATA drives). There may be deployments where the traffic pattern consists of a mix of small objects (such as Web pages, graphics, icons) and large objects (such as videos, software installation packages).

Storing small objects in disk-cache has a performance overhead, because disk read/write operations are expensive (when compared to RAM read/write operations). You can tune the cache configuration so that small objects are just cached in RAM and delivered to users. Small objects need not be written to disk, thus freeing up the disk-cache for large objects and also reducing the latency for small objects delivery.

Use the namespace **origin-fetch content-store media object-size** CLI command to tune the size of objects that get served from disk. For example, **origin-fetch content-store media object-size 4096** would force Media Flow Controller to store in disk-cache all fetched objects larger than 4 KB.

Use the **analytics cache-ingest size-threshold** CLI command to set the maximum size of an object that can be optionally ingested into the fastest cache tier, such as SSDs. Objects smaller than, or equal to, the configured size are automatically promoted to the fastest cache tier.

Example: Mime-Type Tuning

Media Flow Controller, by default, can cache all static media objects. However, based on the traffic profile of the service provider network, Media Flow Controller can be configured to “selectively” cache objects. For example, Media Flow Controller can be configured to cache only “video” objects and “software installation packages.” To do this, you define namespaces so that Media Flow Controller caches only video files (such as *.mpg, *.wmv, *.mov, and *.flv) and software installation packages (such as *.iso, *.vdf, *.msu, and *.exe). You define the files to be cached by Media Flow Controller using the **namespace match uri** CLI command.

Example: Cache Age Tuning

Media Flow Controller, by default, caches objects of expiry time 60 seconds or less, in RAM. The expectation is that these objects will be evicted soon and they are not worth storing in disk caches (wasting the disk I/O cycles). In transparent proxy deployments, it is recommended to increase this value to 300 seconds or higher, in order to improve the disk cache efficiency. Customers can use the namespace **origin-fetch content-store media cache-age-threshold 300** configuration to tune the cache age.

Example: Popularity Tuning

The Media Flow Controller cache-management algorithm automatically promotes popular or “hot” objects (the most requested objects) to a faster cache tier. “Cold” objects (the least requested objects) remain in slower cache tiers.

You can influence when an object gets promoted to a faster cache tier. Delaying objects from moving to slower cache tiers ensures that “hot” objects are served from the fastest cache tier, improving cache efficiency, which indirectly improves bandwidth savings. By default, when an object is requested three times, it becomes a candidate for cache promotion. Use the **analytics cache-promotion hotness-threshold** CLI command to set a threshold for “hotness” value after which an object is candidate for promotion to a faster cache tier.

In an environment where there is constant churn of objects in the cache, we recommend disabling cache promotion. Constant churn of cached objects may occur due to “short-lived” objects in the cache (due to shorter object expiry time, or highly diversified requests). Disabling cache promotion prevents objects getting promoted to a faster cache tier—especially in circumstances where they get evicted immediately after the promotion. Disable cache promotion with the CLI command **analytics cache-promotion disable**.

Example: Cache-Control Override

The “Cache-Control” header influences the caching behavior of Media Flow Controller. By default, Media Flow Controller does not cache objects that are marked as “No-Cache” or “Private”. However, service providers may wish to selectively override the Cache-Control header to achieve better bandwidth savings. Use the **cache-directive no-cache override** CLI command to override the Cache-Control header. This configuration is a violation of HTTP 1.1 standard and should be applied only for select Web portals whose media delivery methods are well understood by the customer.

Mid-Tier Proxy Deployments

Service providers are using three-tier architectures de-coupling front end servers from back-end storage servers by deploying mid-tier proxies. Mid-tier proxies reduce network latency, save bandwidth costs, offload origin servers, and scale front-end server throughput. Media requests are handled by front-end servers, supporting multiple protocols, that issue HTTP fetch requests to Media Flow Controller mid-tier proxy, instead of to origin servers. Media Flow Controller serves the content just as an origin server would in a network without mid-tier servers.

You can configure Media Flow Controller as a **mid-tier** proxy via a **namespace** setting. In this deployment, you must also configure user browsers to point at the Media Flow Controller; **domain** and **match uri-prefix** settings are generally irrelevant and can be set to **any** and **/** (slash), respectively. If you have multiple namespaces, set a low precedence (**10** is the lowest) so incoming requests can pass to namespaces with specific **uri-prefix** matches (instead of **/** that accepts all incoming requests).

1. Create a namespace with **domain any**, **match uri /** **precedence 10** and **origin-server http absolute-url**.
2. Configure your browser to point at the Media Flow Controller. In Internet Explorer, you do this through the **Tools > Internet Options > Connections > LAN settings** page. In Mozilla Firefox, use the **Tools > Advanced > Network > Connection Settings** page.

In this way, any connections to the Internet go through Media Flow Controller to the appropriate namespace rather than this one.

CHAPTER 4

Configuring and Administering Media Flow Controller (CLI)

- [Before You Configure Media Flow Controller](#)
- [About the Media Flow Controller CLI](#)
- [Logging In to Media Flow Controller for the First Time \(CLI\)](#)
- [Using SSH in Automated Scripts \(CLI\)](#)
- [Setting SSH Keys for Multiple Hosts](#)
- [Media Flow Controller System Configuration Overview \(CLI\)](#)
- [Configuring Interfaces, Hostname, Domain List, DNS, and Default Gateway \(CLI\)](#)
- [Configuring Media Flow Controller Clock and Banners \(CLI\)](#)
- [Creating and Configuring Link Bonding and Static Routes \(CLI\)](#)
- [Understanding Authentication, Authorization, and User Options](#)
- [Configuring Media Flow Controller User Accounts \(CLI\)](#)
- [Applying the Media Flow Controller License \(CLI\)](#)
- [Media Flow Controller Policy Configurations Overview](#)
- [Setting Analytics Options \(CLI\)](#)
- [Setting Network Connection Options \(CLI\)](#)
- [Configuring Media Flow Controller Delivery Protocols \(CLI\)](#)
- [Managing the Media Flow Controller Disk Cache \(CLI\)](#)
- [Installing and Using FMS in Media Flow Controller \(CLI\)](#)
- [Administering Media Flow Controller Overview \(CLI\)](#)
- [Saving and Applying Configurations, Resetting Factory Defaults \(CLI\)](#)
- [Rebooting Media Flow Controller \(CLI\)](#)
- [Upgrading Media Flow Controller \(CLI\)](#)
- [Configuring the Web Interface \(CLI\)](#)
- [Configuring the Web Interface Proxy \(CLI\)](#)
- [Configuring Caching All Contents for a Website \(CLI\)](#)

Before You Configure Media Flow Controller

Media Flow Controller is a network appliance and the network parameters must be configured first, as any network appliance. Before you begin, you need to know:

- Hostnames/IP addresses (including but not limited to, subnet mask, default gateway, DNS servers, and NTP servers) for the Media Flow Controller traffic and management ports (eth0 is default management port, eth1 is the recommended origin-fetch interface).
- Hostnames/IP addresses for external servers such as origin servers/libraries, logging, SNMP, SSH, or storage servers. In order to configure the (required) namespace, you must know the **uri-prefix** (see [uri-prefix](#) for definition), domain name, and origin server FQDN (fully qualified domain name) or IP address, at a minimum. This information tells Media Flow Controller where to fetch media from and how to handle it.
- Domain names for the Media Flow Controller to resolve unqualified hostnames.
- The users you want to be able to administer or monitor Media Flow Controller, their e-mail addresses (for event e-mail notifications), and the authentication / authorization schemes you want to use; these schemes can be complicated and should be prepared by an expert.
- The query parameters that you use in URLs to pass information, if you expect to configure a virtual player (not required). Many content delivery networks (CDNs) have proprietary query parameters (also known as query params) already defined.
- The types of content that you serve and their optimal delivery rates, the protocols that you use for delivery, and the general bandwidths of delivery connections that you want to maintain.

Related Topics

- [“Understanding Media Flow Controller” on page 39](#)
- [“Configuring Virtual Players, Media Fetch and Pre-Staging \(CLI\)” on page 123](#)
- [“Configuring Namespaces \(CLI\)” on page 141](#)

About the Media Flow Controller CLI

The Juniper Networks Media Flow Controller command-line interface (CLI) supports industry-standard commands for configuration and management as well as Media Flow Controller specific commands.

The CLI supports command-line editing: press the up arrow to repeat previous lines, and the left arrow to edit the current line. The CLI also supports command completion when you press the **Tab** key. Commands must terminate with CRLF (carriage return followed by newline).

- [Connecting and Logging In](#)
- [Using the Command Modes](#)
- [Prompt and Response Conventions](#)
- [CLI Options](#)

Connecting and Logging In

You can connect to the CLI with SSH, Telnet (after enabled; Telnet is disabled by default), or serial console using the IP address of your Media Flow Controller. The Media Flow Controller responds with a login prompt. Enter **admin** as the user; there is no default password. After you have connected, you must enter **enable** and then **configure terminal** in order to begin configuring Media Flow Controller.

Likewise, you can log in to the Web-based interface by entering the IP address in a browser window and using **admin** as the login name. The Management Console has a subset of the CLI commands, but is good for simple or First Day configurations.

Each user account has at least one privilege level that determines which commands they can issue and what CLI modes they can access:

- Administrator (**admin**)—Full privileges. Can enter **Enable** mode and **Config** mode.
- Monitor (**monitor**)—Can read data (not including logs) and perform actions, but not change any configuration. Can enter **Enable** mode from **Standard** mode but cannot change configurations.
- Unprivileged (**unpriv**)—Can issue a small subset of commands including debugging and show commands. Can log in to **Standard** mode only.

Using the Command Modes

When you log in to the management shell over SSH (or optionally TELNET, if enabled; this is not recommended) you are in the lowest tier, **Standard** mode; only **show**, **help**, diagnostic commands and a few others can be entered. You get into **Enable** mode by issuing the **enable** command. In **Enable** mode you can view current configurations but not make configurations. You need to enter **Configure** mode to make any changes. The CLI can be in one of three modes, which determine which set of commands are available:

- **Standard** mode—When the CLI is launched, it begins in **Standard** mode. This is the most restrictive mode and only has commands to query a restricted set of state information. You cannot take any actions that would directly affect the system, you cannot change settings.
- **Enable** mode—The **enable** command moves you to **Enable** mode. This has commands to view all state information, and take certain kinds of actions like rebooting the system, but does not allow any configuration to be changed. Its commands are a superset of those in Standard mode. Enter **disable** to exit **Enable** mode.
- **Configure** mode—The **configure terminal** command moves you to **Configure** mode. This has a full unrestricted set of commands to view anything, take any action, or change any configuration. Its commands are a superset of those in **Enable** mode. Enter **exit** to leave **Configure** mode.
- **Prefix** mode—Some commands have a **prefix** mode; that is, when you enter a keyword, you enter a mode for that configuration. For example:

```
MFC (config) # accesslog
MFC (config accesslog) #
```

When in the **prefix** mode, you can only make configurations for that command set and typing **?** (question mark) shows you only the options for those configurations. To leave the prefix mode, type **exit**.

Example:

```
MFC > enable
```

```
MFC #          configure terminal
MFC (config) # namespace test
MFC (config namespace test) # exit
MFC (config) # exit
MFC #          disable
MFC >
```

Prompt and Response Conventions

The prompt always begins with the hostname of the system. What follows depends on what command mode you are in. To demonstrate by example, say the hostname is “MFC”. The prompts for each of the modes would be:

```
MFC >          Standard mode
MFC #          Enable mode
MFC (config) # Configure mode
```

Successful commands do not print any response. The next thing you see after pressing **Enter** is the command prompt. If an error is encountered in executing a command, the response begins with % (percent sign), followed by some text describing the error.



NOTE: All CLI commands allow completion with TAB. For example, typing **en** and then pressing TAB completes the **en** command out to **enable**. Completion (hitting TAB) also shows all commands following the typed letters; for example, typing **e** (in Standard mode) and then pressing TAB shows **enable** and **exit** as the available commands starting with **e**.

CLI Options

There are four groups of commands relating to the CLI itself:

- **cli session** commands change a setting only for the current CLI session. They do not affect any other sessions, and can be performed by any user at any time.
- **cli default** commands change the defaults for the specified setting for all future CLI sessions of all users. They also change the setting for the current session from which they were executed, but not for any other currently active sessions. Since they change configuration, the user must be in configuration mode to run them, so they can only be run by **admin** privilege user.
- Other **cli** commands that take one-time actions, rather than change a setting, and thus do not fall under the session or default umbrellas. For example, **cli clear-history**.
- **terminal** commands are clones of a subset of the **cli session** commands, and are only present for ease-of-use.

See [cli](#) for CLI command details.



NOTE: Some settings, such as the terminal length and width, are inherently session-specific, and there are no corresponding commands to set defaults. Also, some commands are only available in default form.

Logging In to Media Flow Controller for the First Time (CLI)

Before you log in to Media Flow Controller for the first time, see [“Before You Configure Media Flow Controller” on page 82](#).

To log in to the system command line interface (CLI) for the first time, you need the IP address assigned the management interface.

1. Open an SSH session and enter the Media Flow Controller management IP address or hostname, or open a serial console session with the console server IP address and port, to connect.
2. Log in with these default credentials (there is no default password).
User: **admin**

To log in to the Management Console (Web UI), just navigate to the configured Media Flow Controller IP address, specify the management port (**:8080**), and use the same login credentials. The Management Console has a subset of the CLI commands, but is good for simple or First Day configurations. Example:

```
http://192.168.1.100:8080.
```

Related Topics

- [“Understanding Media Flow Controller” on page 39](#)

Using SSH in Automated Scripts (CLI)

SSH clients like, **ssh**, **sftp** and **scp** require authentication for each session. By default, these clients use password authentication. To automate authentication for trusted users and hosts, including Media Flow Controller hosts, you must exchange host keys between each host pair.

For example, **accesslog copy** can use **scp** or **sftp** to auto-upload log files. But to enable Media Flow Controller to automatically upload log files without password authentication, the SSH utilities must be configured to use host key-based authentication. You can identify a machine in the network and its user to be a trusted user who can execute commands without entering a password.

To allow SSH command execution for a user on a trusted host:

1. Identify a trusted machine in the network.
2. Generate an RSA public-private key pair for the client machine. On a Linux host the RSA host key is typically kept in **/etc/ssh/ssh_host_rsa_key.pub** or **/etc/ssh_host_rsa_key.pub**.

```
re01# ssh-keygen -t rsa
Generating public/private rsa key pair.
Enter file in which to save the key (/root/.ssh/id_rsa): /root/joe/.ssh/
mykey
Enter passphrase (empty for no passphrase):
Enter same passphrase again:
Your identification has been saved in /root/joe/.ssh/mykey.
Your public key has been saved in /root/joe/.ssh/mykey.pub.
```

3. Add the public key for the client machine to the Media Flow Controller. For example, to add global SSH authentication:

```
(config) # ssh client global known-host "<TARGET-IP> <HOST-KEY>"
```

Note: The quotation marks are required.

TARGET-IP is the IP address of the trusted host. HOST-KEY is the entire host key entry in the RSA host key file.

4. Check the RSA host key:

```
(config) # show ssh client
```

You can now execute any CLI command from the client machine as the trusted user and you are not prompted for a password.

5. Test the password less authentication. From the trusted client machine, execute the following commands:

- To remotely login,

```
$ ssh <trusted_user>@<Media_Flow_Controller_appliance>
```

- To execute CLI commands in **Enable** mode, use this command line:

```
$ ssh <trusted_user>@<Media_Flow_Controller_appliance> "cli enable\"<command>\""
```

- To execute CLI commands in **Configure** mode, use this command line:

```
$ ssh <trusted_user>@<Media_Flow_Controller_appliance> "cli enable\"configure terminal\"\"<command>\""
```

Enter as many commands as needed following the syntax.

Related Topics

- [“Understanding Media Flow Controller” on page 39](#)
- [“Setting SSH Keys for Multiple Hosts” on page 86](#)

Setting SSH Keys for Multiple Hosts

This procedure shows how to setup SSH keys so that multiple machines can securely communicate.

1. Take the RSA key from the target host that you want to push the logs to and add it to Media Flow Controller. You can get the RSA host key from this location in linux:

```
/etc/ssh/ssh_host_rsa_key.pub
```

2. Add the known host RSA Key value to Media Flow Controller:

```
ssh client global known-host "10.157.42.68 ssh-rsa
AAAAB3NzaC1yc2EAAAABIwAAAQEA2NXZxA2nT7N1pvGAKS73EiN/NFzULNuSQb/
2gnNwMSEWaBBLP/
c94RU7c9CD0ooiHNwz1jdh2SUQIbzuly55e0g30vQZhbu2tDWWi3ESVAzH+Q1GyNmAmM9K
CnRLjb2NKNnA4iwfAkauqkITa6zvW3BA+3dR5wnb8d20m9vNZZhXMnkmFEbR0VKMWQF0cp
796f3hoOBBWniwt+l2V5BMDZg80Nz9UqA7reZR0VkiKwhsQjWRSq+5NmtBBrsAZ+t3YSQm
s4zJY
```

3. Generate the public/private keys in Media Flow Controller:

```
(config) # ssh client user admin identity rsa2 generate
```

4. Get Media Flow Controller's public key:

```
(config) # show ssh client

SSH client Strict Hostkey Checking: ask
No SSH global known hosts configured.
User Identities:
  User admin:
    RSAv2 Public key:
      ssh-rsa
      AAAAB3NzaC1yc2EAAAABIwAAAIEAzsG6puPiaHPrvFp8oTCiiL+JyqdQ2h792sjADfcflq6qK
      jq/
      Tq8pCJM2Hy5R+8vvAIRWASGBMOypaEYE3IiUHUhD+vGUChbgJGRh7HIURcIC3Sy+JtWkH45ox
      KKsuaSmlaPLG1H8KU3jeQ9XDMc4BR293vq5q4Fmvi2MJkkCfb0=
```

5. Paste the Media Flow Controller public key in the following path in the machine where you want to upload the file using this syntax:

```
"/user_name/.ssh/authorized_keys"
```

6. To verify, do a manual upload:

```
(config) # upload accesslog current sftp://user@host:path
```

For example:

```
sftp://root@192.168.1.10:/tmp
```

The first time you do it you'll see something like this:

```
MFC (config) # upload accesslog current sftp://root@10.168.1.10:/tmp
sftp> cd /tmp
sftp> put access2.log access2.log.tmp
Uploading access2.log to /tmp/access2.log.tmp
sftp> -rm access2.log
Couldn't stat remote file: No such file or directory Removing /tmp/
access2.log Couldn't delete file: No such file or directory
sftp> rename access2.log.tmp access2.log
sftp> exit
MFC (config) #
```

Related Topics

- [“Understanding Media Flow Controller” on page 39](#)
- [“Using SSH in Automated Scripts \(CLI\)” on page 85](#)

Media Flow Controller System Configuration Overview (CLI)

You can configure many basic system settings using the Media Flow Controller CLI.



NOTE: Some settings, such as the terminal length and width, are inherently session-specific, and there are no corresponding commands to set defaults. Also, some commands are only available in default form.

NOTE: Save your settings after each configuration by typing **write memory**.

To configure system settings:

- Configure interfaces, the hostname, domain list, DNS, and default gateway. See [“Configuring Interfaces, Hostname, Domain List, DNS, and Default Gateway \(CLI\)” on page 88](#) and [“Example: Media Flow Controller Interface Configuration” on page 90](#)
- Configure the system clock and banner. See [“Configuring Media Flow Controller Clock and Banners \(CLI\)” on page 92](#)
- Configure link bonding and static routes. See [“Creating and Configuring Link Bonding and Static Routes \(CLI\)” on page 93](#)
- Understand authentication, authorization, and user account options. See [“Understanding Authentication, Authorization, and User Options” on page 95](#)
- Configure user accounts. See [“Configuring Media Flow Controller User Accounts \(CLI\)” on page 96](#)
- Apply the Media Flow Controller license. See [“Applying the Media Flow Controller License \(CLI\)” on page 97](#)

Related Topics

- [“Configuring and Using Media Flow Controller Logs and Alarms” on page 173](#)
- [“Configuring Media Flow Controller Load Balancing” on page 159](#)
- [“Troubleshooting Media Flow Controller” on page 211](#)

Configuring Interfaces, Hostname, Domain List, DNS, and Default Gateway (CLI)

Interfaces configured with IP addresses, a hostname for the system that can be used in place of an IP address, the Domain Name System (DNS), and default gateway are typical network connectivity configurations. The domain list helps resolve unqualified hostnames. Before you configure Media Flow Controller interfaces, see [“Before You Configure Media Flow Controller” on page 82](#). See **interface** and **ip** for CLI details.

Tip! You may want to temporarily change the CLI default logout time (**900** = 15 minutes); to do this, use this command:

```
cli session auto-logout <seconds>
```


To configure Media Flow Controller interfaces, hostname, domain list, DNS, and default gateway:

1. Configure interface IP addresses for management (eth0), and origin fetch (eth1). Later, use the **delivery protocol** commands to configure traffic interfaces as needed (described in [“Media Flow Controller Policy Configurations Overview”](#)). It is important to keep the traffic ports separate from the origin fetch ports for proper functioning of assured-flow, if used. Use **show interfaces** to verify.

```
interface <interface_name> ip address <management_IP_address>
    <netmask_or_length>
```

2. Configure the system hostname. Use **show hosts** to verify.
3. Enable or disable (with **no**) use of DHCP on the specified interface. When enabled, DHCP gets the IP address and netmask, so those settings are ignored. Conversely, setting the IP address and netmask disables DHCP implicitly. Use **renew** to force a restart on the DHCP client for the specified interface. Default is **disabled**.

```
interface <interface_name> dhcp
```

4. Optionally, add an **alias** (a secondary address) for this interface; set an **index** (name) to create a pseudo interface with an **ip address** and **netmask**. Use **show interface** to verify.

```
interface <interface_name> alias <index> ip address <IP_address><netmask>
```

5. Configure domain list (to resolve unqualified hostnames), and name server (DNS). Use **show hosts** to verify.

```
ip domain-list <domain_name_for_resolving_hostnames> ...
ip name-server <DNS_server_IP_address>
```

6. Configure the default gateway. Use **show ip default-gateway** to verify.

```
ip default-gateway <default_gateway_IP_address>
```

7. Since delivery changes have been made, restart the delivery service (**mod-delivery**).

```
service restart mod-delivery
```

Sample configuration from unconfigured login prompt:

```
mfc-unconfigured-8a4990 login: admin
mfc-unconfigured-8a4990 > enable
mfc-unconfigured-8a4990 # configure terminal
mfc-unconfigured-8a4990 (config) # interface eth0 ip address 123.45.10.9 /
    24
mfc-unconfigured-8a4990 (config) # hostname MFC
MFC (config) # interface sit0 dhcp
MFC (config) # interface sit0 alias sit0a 123.45.11.9 /24
MFC (config) # ip domain-list example.local
MFC (config) # ip name-server 172.19.172.1
MFC (config) # ip default-gateway 123.45.10.1
MFC (config) # service restart mod-delivery
```



NOTE: Additional interface commands, such as adding a **comment**, using **identify** to flash LED lights, and configuring the **txqueuelen** (transmit queue length), are not included in this configuration example. See [interface](#) for CLI details.

Cutting and Pasting an Interface Configuration (CLI)

You can copy this series of commands, plug in your variables, and save them to a file to re-use as needed.

```
enable
configure terminal
interface eth0 ip address <IP_address> {<netmask> | <mask_length>}
hostname <name>
ip domain-list <domain_name_for_resolving_hostnames> ...
ip name-server <IP_address>
ip default-gateway <IP_address>
service restart mod-delivery
write memory
```

Related Topics

- [“Before You Configure Media Flow Controller” on page 82](#)
- [“Media Flow Controller Policy Configurations Overview” on page 98](#)

Example: Media Flow Controller Interface Configuration

When Media Flow Controller initializes, the on-board Ethernet interfaces are numbered Eth0, Eth1, and so on. When a NIC, dual- or quad- port, is attached to the server, the first NIC (goes by PCI channel number) gets interface names Eth10, Eth11, and so on to Eth19. The second NIC gets the names Eth20, Eth21, and so on to Eth29; it is assumed that only up to 10 Ethernet interfaces per NIC exist. See [Table 9, “Example Machine Setup of Management and Traffic Ports.”](#) for details.



CAUTION: For VXA Series Media Flow Engine appliances, never change the Ethernet name mappings, all interface assignments are handled automatically during manufacturing.

In [Table 9](#), and [Figure 7](#), for example, the wiring logic is:

- **Eth 0**—Running SNMP, sending analytics to another machine, Web management, SSH, and Telnet; connected to your internal network.
- **Eth 1**—Upstream fetching content from origin; connected to the network that connects to the origin server.
- **Eth 2 - 5**—Service traffic; connected to the public Internet. These interfaces must have IP addresses.

Table 9 Example Machine Setup of Management and Traffic Ports

| Interface | Connectivity | IP Address | Subnet Mask | Open Ports | Internet Access | Purpose |
|-----------|--------------|---------------|---------------|------------|-----------------|--------------|
| Eth 0 | onboard | 192.168.1.100 | 255.255.255.0 | 8080, 22 | yes | Management |
| Eth 1 | onboard | 172.20.46.10* | 255.255.255.0 | 80 | yes | Origin fetch |

Table 9 Example Machine Setup of Management and Traffic Ports (Continued)

| Interface | Connectivity | IP Address | Subnet Mask | Open Ports | Internet Access | Purpose |
|-----------|--------------|------------|---------------|------------|-----------------|---------|
| Eth 2 | PCIe card | 10.1.1.11 | 255.255.255.0 | 80 | not applicable | Traffic |
| Eth 3 | PCIe card | 10.1.2.11 | 255.255.255.0 | 80 | not applicable | Traffic |
| Eth 4 | PCIe card | 10.1.3.11 | 255.255.255.0 | 80 | not applicable | Traffic |
| Eth 5 | PCIe card | 10.1.4.11 | 255.255.255.0 | 80 | not applicable | Traffic |

*Eth0 AND Eth1 can be on the same subnet; this examples indicates they are not.

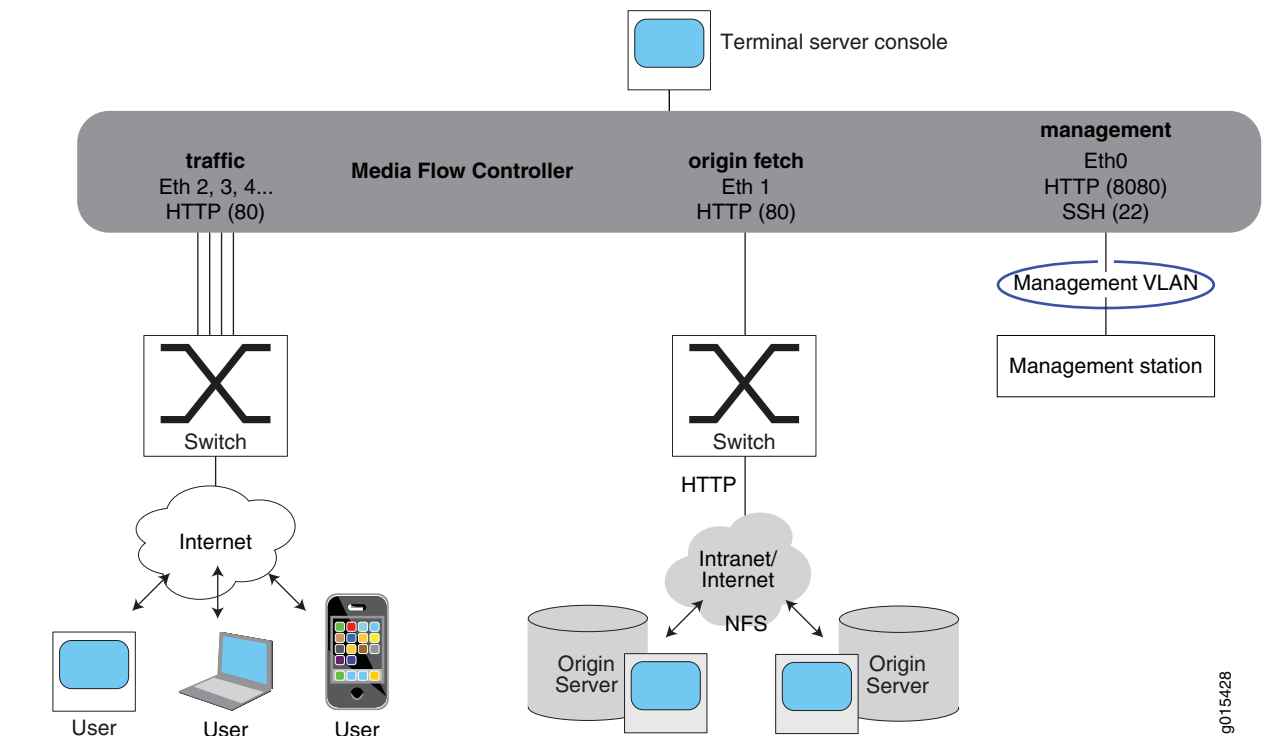


Figure 7 Example Connectivity

Eth0 and Eth1, typically the first two interfaces you use, are usually the first two network ports built into the system—either part of the system board, or the first add-in card or module.

Media Flow Controller supports Lights Out (or "out-of-band") management, which involves the use of a dedicated management channel for device maintenance so you can monitor and manage servers and other network equipment by remote control regardless of whether the machine is powered on. You can configure out-of-band management through the BIOS.

NOTE: Media Flow Controller does not support RAID arrays.

9015428

Related Topics

- [“Before You Configure Media Flow Controller” on page 82](#)
- [“Media Flow Controller Policy Configurations Overview” on page 98](#)

Configuring Media Flow Controller Clock and Banners (CLI)

Proper time configuration is required for accurate functioning. An NTP (network time protocol) server, or a group of NTP servers (peers) is desirable. Banners allow you to display messages to users. Before you configure Media Flow Controller system clock and banners, see [“Before You Configure Media Flow Controller” on page 82](#). NTP is enabled by default.

To configure an NTP (network time protocol) server OR system clock and timezone:

1. Configure NTP server. Use **show ntp** to verify.

```
ntp server <IP_address>
```

Alternately, configure the system clock, and timezone. Use **show clock** to verify.

```
clock set <hh:mm:ss> [<yyyy/mm/dd>]
```

```
clock timezone <zone> [<zone_word>] [<zone_word>] ...
```

2. Optionally, configure NTP peers; if you do not specify version number **3**, the default, version **4**, is used.

```
ntp peer <IP_address>
```

3. Optionally, configure banners. There are two configurable banners: **motd** (message of the day) and **login**. In the CLI, both are displayed at the command line when you log in; in the Management Console, only the **login** message is displayed, on the login page. Multi-word messages must be surrounded by quotes. Use **show host** and **show banner** to verify.

```
banner [login <message_string>] [banner motd <message_string>]
```

Sample configuration:

```
MFC (config) # ntp server 123.45.10.7
```

```
MFC (config) # ntp peer 123.45.10.8
```

```
MFC (config) # clock set 15:51:30
```

```
MFC (config) # clock timezone America North United_States Pacific
```

```
MFC (config) # banner login "Welcome to Media Flow Controller"
```

```
MFC (config) # banner motd "Please note new link bonding commands"
```

Related Topics

- [“Before You Configure Media Flow Controller” on page 82](#)

Creating and Configuring Link Bonding and Static Routes (CLI)

Link bonding allows you to increase link speed by using multiple ports simultaneously. Static routes can be useful for the "stub" parts of your network that need only connect to each other. Before you configure Media Flow Controller link bonding and static routes, see ["Before You Configure Media Flow Controller" on page 82](#).

Media Flow Controller supports three bonding modes:

- **balance-rr**—"Round-robin" mode. Sends TCP/IP packets belonging to the same session across multiple links. Out-of-order TCP packets coming through different links are retransmitted; supports load balancing and failover.
- **balance-xor-layer3+4**— Traffic to a particular network peer goes across multiple links, although packets belonging to a single connection/session do not span multiple links; supports load balancing and failover. Link selection is based on TCP port and IP address.
- **link-agg-layer3+4**—Link Aggregation Control Protocol (LACP). Allows the automatic negotiation of port bundling to form a single logical channel between LACP-enabled links; supports load balancing and failover.

Configuring Link Bonding and Static Routes (CLI)

Bond interfaces to create a port-channel or aggregated link for load distribution across links and increased link availability. Example shows bonding interfaces eth10 and eth11 as a named bonded interface "0". In this way, Layer 2 packets are distributed across the defined links for load distribution; if one of the links fail, the other links take over the media delivery. After you have created the bonded interface, you can use the **delivery protocol** command to assign it as a traffic interface and configure its listen port, if needed.

Individual links under a bond of 1G interfaces cannot be configured for different speed (10/100/1000). Only *1* bond interface is allowed. If a second bond interface is created it does not work well.



NOTE: Do not name a bond interface "0" or "1" as these names can cause problems with the TCP dump function. It is better to use an alphanumeric character like "ae0," "b1," "bond0" or some such.

To configure link bonding and static routes:

1. Create the bond interface with a **name** and specify a **mode**. The CLI lists several options for **bond <bond_interface> mode** that are not supported. Only **balance-rr** ("round-robin"), **balance-xor-layer3+4** (Non-LACP), and **link-agg-layer3+4** (LACP) are supported modes.
`bond <name_for_virtual_interface> mode <bond_mode>`
2. Add up to four interfaces to bond; repeat as needed. Mixed mode bonding (1G and 10G interfaces bonded in a single bond) is not supported .
`interface <interface_name> <virtual_interface_name>`
3. Assign the new bonded interface as a traffic interface and set non-default (80) listen ports, if needed; up to 64 ports can be assigned. After you assign a traffic interface, Media Flow Controller accepts traffic only on those assigned interfaces (up to 10); by default, Media Flow Controller accepts traffic on all interfaces.

```
delivery protocol http interface <bonded_interface_name>
delivery protocol http listen port <port> <port> <port>
```

Example:

```
MFC (config) # bond b1 mode balance-rr
MFC (config) # interface eth10 bond b1
MFC (config) # interface eth11 bond b1
MFC (config) # delivery protocol http interface bond b1 eth12 eth13
MFC (config) # delivery protocol http listen port 80 81 82
MFC (config) # show bonds
```

Bonded Interface b1:

```
Enabled:          yes
Mode:             balance-rr
Link Monitor Time: 100
Interfaces:
  eth10
  eth11
```

4. Configure the IP address for the bonded interface.

```
MFC (config) # interface b1 <IP_address>
```

5. Optionally, configure static routes and ensure a static host mapping for the defined hostname. The **ip route** command only works on devices that already have an IP address assigned. Use **show ip route** to verify.

```
ip route <network_prefix> {<netmask> | <mask_length>} {next_hop_IP_address
| interface_name}
```

```
ip map-hostname
```

Example:

```
MFC (config) # ip route 123.45.10.0 /24 eth0
```

6. Since delivery changes have been made, restart the delivery service.

```
service restart mod-delivery
```



NOTE: Bonded interfaces show Speed and Duplex as UNKNOWN in **show interfaces** output; this is not an error condition.

Related Topics

- [“Before You Configure Media Flow Controller” on page 82](#)
- [“Load Balancing with Direct Server Return” on page 159](#)

Understanding Authentication, Authorization, and User Options

Several configurations or tasks can use an already configured authentication / authorization scheme (AAA, namespace pre-staging, users, file transfers, and so forth). Authentication schemes can be complex to configure—this section does not attempt to guide you through the configuration steps for setting authentication or AAA options, but provides references to the CLI commands. Before configuring any authentication / authorization schemes, you must have this information: the hostname or IP address of the authenticating server, and a shared secret for authentication.

- [About MD5, SHA1, AES-128, and DES](#)
- [User Account Defaults and States](#)

About MD5, SHA1, AES-128, and DES

The first two, **md5** and **sha1**, are cryptographic hash algorithms.

- **md5**—Message-Digest algorithm 5. Considered somewhat faster but less secure than sha1, but still supported for legacy systems. Generates a 128-bit (16 byte) hash.
- **sha1**—Secure Hash algorithm 1. Considered more secure than md5 but still vulnerable to collision attacks. Generates a 160-bit (20-byte) hash.

The second two, **AES-128** and **DES**, are encryption standards used to encrypt and un-encrypt data.

- **AES-128**—Advanced Encryption Standard; 128 is a specific “block cipher.” AES is a newer standard than DES and considered much more secure. Generates a 128 bit encryption key. AES is an asymmetric encryption algorithm which means the sender uses the public key of the receiver to encrypt the message and the receiver uses its private key to decrypt the message.
- **DES**—Data Encryption Standard. This standard is older than AES-128 and considered less secure than AES-128 but still supported for legacy systems using it. Generates 56 bits encryption key. DES is a symmetric encryption algorithm which means that you use the same key to encrypt and decrypt the message.

User Account Defaults and States

The system comes initially with three accounts already created:

- **admin**—Full privileges to do anything on the system.
- **mfc_probe_ftpuser**—The auto-created user for the internal watchdog.
- **cmcrendv**—Not Supported.
- **monitor**—Privileges to read almost everything on the system (does not include logs), and perform some actions, but cannot modify configurations.

These accounts are both enabled, and by default have no password required for login.

There are five states an account may be in:

- “Account disabled” (not listed in /etc/passwd). The **admin** account cannot be disabled.
username foo disable
- “Local password login disabled” (hashed password set to "*"). There is no locally-configured password to permit the user to log in. The user may still log in using an SSH authorized key if one is installed, or remote authentication (for example, RADIUS or TACACS+). The **admin** account may not be in this state unless it has an SSH authorized key installed.
username foo disable password
- “All password login disabled” (hashed password set to "!"). No CLI command for this; the hashed password must be set to "!". Same as "Local password login disabled" except that the user cannot be remotely authenticated (for example, by a RADIUS or TACACS+ server). The user may still log in using an SSH authorized key if one is installed. The **admin** account may not be in this state unless it has an SSH authorized key installed.
- “Local password set”. The user can log in by typing the password whose hashed version we have stored. This is not necessary if an SSH authorized key is installed, or if a remote auth server comes earlier in the authentication order.
username foo password mypassword
- “No password required for login” (hashed password set to ""). Anyone can log in to this account without providing authentication. The **admin** and **monitor** accounts begin in this state (unless overridden by configured defaults), but should be changed for better security.
username foo nopassword

Related Topics

- [“Before You Configure Media Flow Controller” on page 82](#)

Configuring Media Flow Controller User Accounts (CLI)

Configure user accounts to allow multiple administrators to make configuration changes, or to allow certain people to view or monitor the appliance. Before you configure Media Flow Controller user accounts, see [“Before You Configure Media Flow Controller” on page 82](#).

To configure users with the CLI:

1. Configure authentication / authorization; see [radius-server](#) and [tacacs-server](#) for CLI details.
2. After your authentication settings are made, configure authentication and authorization parameters such as setting the default login authentication order and default authorization mapping for local and remote users. See [aaa](#) for CLI details.
3. Configure users. Media Flow Controller provides three capability sets for users: **admin** (full privileges), **monitor** (can view configurations but make no changes), and **unpriv** (very limited command access); see [username](#) for CLI details. In addition to the capabilities, you can configure password options and disable a user account. Use **show usernames** to verify.

- a. Add a user and specify the capability; users are added with admin privileges:

```
username <username> capability <capability>
```

- b. Delete a user:

```
no username <username>
```

- c. Disable a user's password; this does not remove the user or the password:

```
username <username> disable password
```

- d. For a defined user, allow no password:

```
username <username> nopassword
```

- e. For a defined user, configure a password. If no password is specified the user logs in with no password; if **0** is specified, enter a password in cleartext (the system encrypts it using the DES algorithm) and the user logs in with that password; if **7** is specified, you must enter the previously-created, DES encrypted password for that user at the command line. Media Flow Controller default **admin** user does not have a default password; set an **admin** password to secure and restrict administration.

```
username password [ 0 <cleartext_password> | 7 <encrypted_password> | <cleartext_password>]
```

Example:

```
MFC (config) # username joe capability unpriv
MFC (config) # username joe password 12345
MFC (config) # username joe disable password
MFC (config) # username joe nopassword
```

Related Topics

- [“Before You Configure Media Flow Controller” on page 82](#)
- [“Understanding Authentication, Authorization, and User Options” on page 95](#)

Applying the Media Flow Controller License (CLI)

Media Flow Controller comes unlicensed and by default can only support 10 connections at 200 Kbps each; it is a 40 Mbps delivery system. Neither the number of sessions or the session rate (**network connection** parameters) is configurable without the Media Flow Controller license. Contact Juniper Networks to obtain the Media Flow Controller license for normal operations. You need to provide the node ID, which is the MAC address of Eth0 interface; use **show interface eth0** to find the hardware (HW MAC) address. Based on this, Juniper Networks will provide you a license key. After installing the license, you get full feature capability. See **license** for CLI details.

To apply the Media Flow Controller license:

1. Install a license.

```
license install <license_key>
```
2. Delete a license

```
license delete <license_key>
```
3. View installed licenses, including expiration dates.

```
show licenses
```

Example:

```
MFC (config) # show license
No licenses have been configured.
```

```

MFC (config) # show network
Network time out (seconds)           : 60
Maximum concurrent sessions         : 10
Per Session assured flow rate (Kbits/sec) : 0
Per Session Maximum bandwidth (Kbits/sec) : 200
MFC (config) # license install LK2-MFC-413E-5N42-3EE6-4381-GLL8-CE98
MFC (config) # show license
License 1: LK2-MFC-413E-5N42-3EE6-4381-GLL8-CE98
  Feature:           Media Flow Controller
  Valid:             yes
  Start date:        2009/03/15 (ok)
  End date:          2009/06/30 (ok)
  Tied to MAC addr: 00:1E:C9:FF:0C:FA (ok)
  Active:            yes
MFC (config) # show network
Network time out (seconds)           : 60
Maximum concurrent sessions         : 64000
Per Session assured flow rate (Kbits/sec) : 0
Per Session Maximum bandwidth (Kbits/sec) : 0

```

Related Topics

- [“Before You Configure Media Flow Controller” on page 82](#)

Media Flow Controller Policy Configurations Overview

After you have your appliance network connections and basic settings configured, you are ready to start configuring Media Flow Controller policy settings. Any time network configuration changes are made the delivery service (**mod-delivery**) must be restarted with **service restart mod-delivery**. This includes initial configurations after installation. In the Management Console, do this on the **EZconfig** page.

To configure policy settings:

- Set cache analytics options.
See [“Setting Analytics Options \(CLI\)” on page 99](#)
- Set network connection options.
See [“Setting Network Connection Options \(CLI\)” on page 100](#)
- Configure delivery options.
See [“Configuring Media Flow Controller Delivery Protocols \(CLI\)” on page 102](#)
- Create and configure virtual players.
See [Chapter 5, “Configuring Virtual Players, Media Fetch and Pre-Staging \(CLI\)”](#)
- Create and configure namespaces.
See [Chapter 6, “Configuring Namespaces \(CLI\)”](#)
- Manage the disk cache.
See [“Managing the Media Flow Controller Disk Cache \(CLI\)” on page 103](#)

Related Topics

- [Chapter 3, "Media Flow Controller Deployment Guidelines"](#)
- [Chapter 9, "Configuring and Using Media Flow Controller Logs and Alarms"](#)
- [Chapter 7, "Configuring Media Flow Controller Load Balancing"](#)
- [Chapter 8, "Configuring Media Flow Controller Server Maps"](#)

Setting Analytics Options (CLI)

Configure options for handling the cache. Use **analytics** commands to set a **cache-ingest size threshold** limiting, by size, what objects are cached. You can set controls on **cache-promotion**, including disabling it entirely, or adjusting the hotness-threshold for promoting objects to another cache tier; and you can set a **memory-limit** on the amount of cache memory that can be given to storing Header data. See **analytics** for CLI details. Before you configure Media Flow Controller cache **analytics** options, see ["Before You Configure Media Flow Controller" on page 82](#).

- Media Flow Controller uses the concept of "hotness" to determine when to promote an object to various cache tiers. Tiers are comprised of RAM, SSD, SAS, and SATA. Hotness is defined as three times (3 x) the number of requests for an object. The hotness computation is based on the hit frequency (i.e., number of hits as a function of time). For example, an object with 400 hits in an hour is considered hotter than an object with 500 hits occurring over a period of one day. [Table 10](#) shows the relation between the hotness threshold and the number of hits required for the object to be ingested.

In a transparent proxy deployment, we recommend the default setting, **3**, for hotness threshold for two reasons:

- The analytics manager only keeps track of 200,000 objects; billions of objects will be stored in a typical transparent proxy deployment.
- The hotness algorithm does not take object size into account and can promote a large hot object into the SSD, causing a large number of smaller objects to be evicted and therefore the SSD to be under-utilized.

If necessary, use the following command to reset the default value:

```
no analytics cache-promotion hotness-threshold <number>
```

Table 10 Hotness Thresholds

| Hotness Threshold | Number of Hits Required for Object Ingestion |
|-------------------|--|
| 3* | 1* |
| 4 | 2 |
| 7 | 3 |
| 10 | 4 |

*If the **hotness-threshold** is the default value, **3**, objects are promoted to the lowest tier after the first hit, regardless of tier type. If the hotness of an object (always calculated as the number of hits x 3) becomes 2 x the **hotness-threshold**, it is promoted to SAS; and at 6 x the **hotness-threshold**, it is promoted to SSD. The default hotness thresholds for promotion into SATA, SAS, or SSD are 1:2:6. If you change the value of the **hotness-threshold** to **4**, the hotness thresholds for promotion to the higher tiers is then calculated as 4:8:24.

Configuring Caching Analytics (CLI)

To configure caching analytics with the CLI:

1. Configure a **cache-ingest size threshold**, limiting which objects can be cached. Objects smaller than, or equal to, the configured size are automatically written to the fastest cache tier. Default is **0** (zero), no objects are directly promoted to the fastest tier. Maximum allowed value is **4294967295** (4 GB).
`analytics cache-ingest size-threshold {0 | <bytes>}`
2. Configure **cache-promotion**. Set a threshold for "hotness" value after which an object is candidate for promotion to a higher tier in disk cache. Default is **3**, an object requested three times becomes a candidate for cache promotion.
`analytics cache-promotion hotness-threshold <integer>`
3. Disable or re-enable **cache-promotion**. Default is enabled; if **disable** is used, no cache promotion occurs.
`analytics cache-promotion {disable | enable}`
4. Configure a cache **memory-limit** for Header data. Use either a value between **0** and **2048**, in storage MiB (1,048,576); or set to **auto** to use the built-in algorithm. Default is **500** MiB.
`analytics memory-limit <MiB>`

Related Topics

- [“Before You Configure Media Flow Controller” on page 82](#)
- [“Caching and Origin Clustering” on page 45](#)

Setting Network Connection Options (CLI)

Configure network connection settings (global Media Flow Controller defaults) to limit and control connections. The **assured-flow-rate** and **max-bandwidth** options are available in **virtual-player** configurations, which override **network connection** configurations. Before you configure Media Flow Controller **network connection** options, see [“Before You Configure Media Flow Controller” on page 82](#).

Using Network Connection Assured Flow

Assured Flow is a function using the values configured for **max-bandwidth**, **concurrent session**, and **assured-flow rate** (AFR). AFR is the rate that Media Flow Controller provisions at the *network* level. For example, if a video encoded at 800 Kbps needs to be transferred over HTTP that uses TCP/IP over Ethernet, you must account for the overheads of the HTTP, TCP, IP, and Ethernet protocols. Usually, HTTP + TCP + IP + Ethernet overheads amount to 10 to 15 percent. With this in mind, AFR for a video encoded at 800 Kbps should be set to 900 Kbps or slightly higher. See [“Media Flow Controller AssuredFlow” on page 47](#) for more information. See **network** for CLI details.

Configuring Network Connections (CLI)

To configure network connections with the CLI:

1. Configure global network assured-flow-rate (minimum rate for a given session); default, **0** (zero), means assured flow is disabled (no minimum rate is provisioned).
`network connection assured-flow-rate {0 | <kbps>}`
2. Configure global network concurrent session limit; default is **10** (without Media Flow Controller license), **64000** (with Media Flow Controller license). The Media Flow Controller license changes the default. Maximum allowed is **256,000** in Release 2.1.
`network connection concurrent session {64000 | <integer>}`
3. Configure global network socket idled-out time in seconds; this is the time the network waits before closing a connection when there is no data in session; default is **60** seconds.
`network connection idle timeout <seconds>`
4. Configure global network maximum allowed bandwidth (burst rate); even if there is available bandwidth in the link, Media Flow Controller does not allocate more than this value for a session. When there is a full download, Media Flow Controller tries to allocate this value to the session; default is **200** without Media Flow Controller license, **0** kbps (unbounded) with Media Flow Controller license. The Media Flow Controller license changes the default.
`network connection max-bandwidth {0 | <kbps>}`
5. To verify configurations.
`show network`
6. Save configuration settings.
`write memory`

Example:

```
MFC (config) # network connection assured-flow-rate 2000
MFC (config) # network connection concurrent session 64000
MFC (config) # network connection idle timeout 900
MFC (config) # network connection max-bandwidth 2000
MFC (config) # show network

Network time out (seconds)           : 900
Maximum concurrent sessions         : 64000
Per Session assured flow rate (Kbits/sec): 2000
Per Session Maximum bandwidth (Kbits/sec): 2000

Network Access-Control PERMIT list: NONE
Network Access-Control DENY list: NONE
```

Related Topics

- [“Before You Configure Media Flow Controller” on page 82](#)
- [“Media Flow Controller AssuredFlow” on page 47](#)

Configuring Media Flow Controller Delivery Protocols (CLI)

These are the ports on the Media Flow Controller that receive and deliver media. These ports typically have Internet access, and should be connected with highest-quality cables. See [Table 9, “Example Machine Setup of Management and Traffic Ports”](#) for an example.

The **delivery protocol** command lets you specify what protocols to use for media delivery and manipulate headers; in Release 2.0.7 **http** and **rtsp** are allowed values. By default, Media Flow Controller listens on all interfaces; if you set specific traffic interfaces, only those are used for traffic. See [delivery](#) for CLI details; see [“Media Flow Controller Delivery Methods” on page 42](#); for background information. Before you configure Media Flow Controller **delivery protocol** options, see [“Before You Configure Media Flow Controller” on page 82](#).

To configure delivery protocol options with the CLI:

1. Media Flow Controller supports known HTTP methods (GET, POST, TRACE, CONNECT, OPTIONS, DELETE, PUT) always. To add support for **http** request methods, use **allow-req** and specify up to 16 custom request methods. Use **all** to permit Media Flow Controller to tunnel any request. Default is **none**, only the known methods listed are allowed. Use **no** to remove the specified method or methods.

```
delivery protocol http allow-req all
```

2. Optionally, disable/enable connection pooling (enabled by default) and configure parameters; for delivery protocol **http** only.

```
delivery protocol http conn-pool origin {enable | disable}
```

- a. Use **max-conn** to limit the maximum allowed pooled connection; default is **4096**, maximum allowed is 128000.
- b. Use **timeout** to configure a pooled connection timeout; default is **90** seconds, maximum allowed is 86,4000 seconds (24 hours).

3. Optionally, configure interfaces for Media Flow Controller traffic; after configured, Media Flow Controller accepts traffic on those interfaces only. Applies to both **http** and **rtsp** delivery protocols. Up to 10 space-separated interfaces can be specified.

```
delivery protocol [http | rtsp] interface <interface>,<interface>...
```

- a. To configure interfaces for transparent proxy, set HTTP interfaces:

```
delivery protocol http interface <interface>,<interface>...
```

- b. Enable them for transparent proxy with this command:

```
delivery protocol http transparent <interface>,<interface>... enable
```

4. Optionally, configure listen ports for the traffic interfaces as needed; default is port **80** for **http**, port **554** for **rtsp**. By default, Media Flow Controller listens on port 80 for HTTP and port 554 for RTSP on all interfaces.

```
delivery protocol http listen port <port>
```

5. Optionally, set the maximum request length (domain + URI + Query Params + Headers), in characters/bytes, for incoming requests (**http** delivery protocol only). Default is **16384** bytes; maximum allowed value is **32768**. Incoming requests with lengths exceeding the configured value are rejected.

```
delivery protocol http req-length maximum <bytes>
```

6. Optionally, enable origin side revalidation requests. The default is **no**; HEAD revalidation requests are more efficient than GET revalidation requests, however, some content websites do not support HEAD requests.

```
delivery protocol http revalidate-get
```

7. Since delivery changes have been made (steps [3](#) and [4](#)), restart the delivery service (**mod-delivery**).

```
service restart mod-delivery
```

8. To verify configurations.

```
show delivery protocol http
```

9. Save configuration settings.

```
write memory
```

Example:

```
MFC (config) # delivery protocol http conn-pool origin max-conn 128000
```

```
MFC (config) # delivery protocol http interface eth11 eth12 eth13
```

```
MFC (config) # delivery protocol http listen port 80
```

```
MFC (config) # delivery protocol http req-length maximum 23576
```

```
MFC (config) # delivery protocol http revalidate-get
```

```
MFC (config) # service restart mod-delivery
```

```
MFC (config) # write memory
```

Related Topics

- [“Before You Configure Media Flow Controller” on page 82](#)
- [“Media Flow Controller Delivery Methods” on page 42](#)

Managing the Media Flow Controller Disk Cache (CLI)

The media caches, or disks, are active and enabled by default and typically require no configuring. Media Flow Controller support "hot swapping" (system operation is not disrupted); however, you must deactivate and disable disks and caching to change disks. Before you activate or enable a cache, run **show media-cache disk list** and get the name assigned to the disk to use in configuration. See [media-cache](#) for CLI details.

Media Flow Controller supports three cache tiers corresponding to SSD (tier 1), SAS HDD (tier 2), and SATA HDD (tier 3). "Hot" content generally stays in tier 1 (the highest). Media Flow Controller promotes contents between the cache tiers based on content hotness (see [“Hot” content \(short tail versus long tail\)” on page 31](#) for definition). As content gets hotter, it is promoted to the next higher tier. First time content is always put in the lowest cache tier. The default values are: **tier1** weight = **6**, **tier2** weight = **2**, **tier3** weight = **1**.

- [Controlling Cookie Cache Behavior \(CLI\)](#)
- [Analyzing the Disk Cache](#)
- [Disk Cache Problem Solving](#)
- [Replacing Bad Disks](#)
- [Correcting Mis-Labeled Disk Types](#)
- [Inserting New Disks into a VXA Series Media Flow Engine](#)

Controlling Cookie Cache Behavior (CLI)

Media Flow Controller can be configured to act on cookies in the (incoming) request path or the (outgoing) response path.

- [About HTTP State Management and Cookies](#)
- [Controlling Cookies In the Request Path](#)

About HTTP State Management and Cookies

HTTP servers respond to each client request without relating that request to previous or subsequent requests; the state management mechanism allows clients and servers to exchange state information by placing HTTP requests and responses within a larger context termed as a session. HTTP servers provide each (user) session with a “cookie” to track the current state of the session. To better understand HTTP cookies, we recommended that you have prior knowledge of RFC 2965.



NOTE: The default behavior of Media Flow Controller is to follow the directives of the **Cache-Control** header in the HTTP request or the HTTP response. However, these can be overridden by using the CLI or the policy engine as outline below.

Media Flow Controller provides configurable options via the command-line interface (CLI) and the Policy engine to control caching for objects with cookies. Filtering behavior control is provided by the Policy Engine interface since it allows for building a rich set of rules via a programmable interface; see [Chapter 13, “Using Media Flow Controller Policy Engine.”](#) for more information.

Controlling Cookies In the Request Path

On the request path, Media Flow Controller can be configured to make a decision on whether to cache or not cache an associated response containing a cookie.

```
namespace <name>
  delivery protocol http
    client-request cookie action cache
```

This CLI command configures Media Flow Controller to cache responses when clients send a **Cookie** or **Cookie2** header in the request. For objects that are already cached in Media Flow Controller, no object revalidation is done as long as the object is still “fresh” (within configured limits). Moreover, if the cached response has a **Set-Cookie** or **Set-Cookie2** header, the response to the client would also contain the same headers.

Use this to cache objects that have cookies in the response.

```
namespace <name>
  delivery protocol http
    client-request cookie action no-cache
```

This CLI command configures Media Flow Controller to tunnel the request directly without performing any cache lookups within Media Flow Controller.

Use this to ignore caching any objects that are likely to have cookies in the request.

Analyzing the Disk Cache

When Media Flow Controller is having caching issues, you want to check the disk cache. To analyze the media disk cache:

1. Get system-assigned disk names to use in configuration. Use **list** to view all disk drives, their names, physical location, serial number, type, and capacity. Use **cache_name** to view information on the specified cache.

```
show media-cache disk {list |<cache_name>}
show media-cache disk list
```

Example:

```
MFC (config) # show media-cache disk list

show media-cache disk list
Device  Type    Tier    Active  Cache  Free Space  State
-----  ----    -
dc_1    SATA    Tier-3  yes     yes    38890 MiB   cache running
dc_2    SATA    Tier-3  yes     yes    68668 MiB   cache running
Total Free Space:  107486 MiB
```

```
MFC (config) # show media-cache disk dc_1
Disk Cache Configuration & Status:
```

```
Device Name/Type: dc_1/SATA
Cache Tier: Tier-3
Activated: yesCache Enabled: yes
Free Space: 2304 MiB
Disk State : cache running
-----
```

2. Determine the free block thresholds of the disk caches.


```
show media-cache free-block threshold
```
3. Disable a disk if you need to pull the disk for any maintenance purposes; for example, to upgrade to a higher capacity disk, replace a SATA disk with a SAS disk, replace a failed disk, reformat the disk, or if contents should not be cached.


```
media-cache disk <cache_name> cache disable
```
4. Deactivate a disk cache. Media Flow Controller allows OIR (On-line Insertion and Removal) of HDD (Hard Disk Drives). However, **the HDD MUST be made inactive to be removed**. When a new HDD is in the disk, it must be made active and (if so decided) enabled for caching.


```
media-cache disk <cache_name> status inactive
```
5. Put in a new drive and mount it.


```
media-cache disk mount-new
```
6. Find the new (inactive) disk's name.


```
show media-cache disk list
```
7. Activate or re-activate a media-cache disk.


```
media-cache disk <cache_name> status active
```
8. Format the disk if it is newly inserted and empty or you do not want to use its contents. Do this after you issue the **mount** command.


```
media-cache disk <cache_name> format
```
9. Enable or re-enable a disk for caching.


```
media-cache disk <cache_name> cache enable
```

Example:

```

MFC (config) # show media-cache disk list
Device  Type   Tier   Active  Cache  Free Space   State
-----  ----   ----   -
dc_1    SATA   Tier-3  yes     yes    38890 MiB    cache running
dc_2    SATA   Tier-3  yes     yes    68668 MiB    cache running
Total Free Space: 107486 MiB

MFC (config) # media-cache disk dc_2 cache disable
Disk Cache Disabled

MFC (config) # media-cache disk dc_2 status inactive
Disk Deactivated

MFC (config) # media-cache disk mount-new
Message sent to detect for new disks
Please check disk state for status

MFC (config) # show media-cache disk list
Device  Type   Tier   Active  Cache  Free Space   State
-----  ----   ----   -
dc_1    SATA   Tier-3  yes     yes    38890 MiB    cache running
dc_2    SATA   Tier-3  no      no     -            disk has been deactivated
Total Free Space: 107486 MiB

MFC (config) # media-cache disk dc_2 status active
Disk Activated

MFC (config) # media-cache disk dc_2 format
Disk Formatted

MFC (config) # media-cache disk dc_2 cache enable
Disk Cache Enabled

MFC (config) # show media-cache disk list
Device  Type   Tier   Active  Cache  Free Space   State
-----  ----   ----   -
dc_1    SATA   Tier-3  yes     yes    38890 MiB    cache running
dc_2    SATA   Tier-3  yes     yes    68668 MiB    cache running
Total Free Space: 107486 MiB

```

Disk Cache Problem Solving

When a disk cache error is displayed, a first step to take is bringing down the disk and bringing it back up; to do, first find cache names, then act on the problem cache:

```

show media-cache disk list
media-cache disk <cache_name> status inactive
media-cache disk <cache_name> status active
media-cache disk <cache_name> enable

```

See [“Disk Cache Error Messages” on page 107](#) for messages you might get at the command line or in a log.

Disk Cache Error Messages

These are the disk state messages you might get when managing the **media-cache** disk:

- DM2_MGMT_STATE_CACHEABLE = "disk cacheable, but cache not enabled"
- DM2_MGMT_STATE_INVALID_FORMAT_BEFORE_MOUNT and DM2_MGMT_STATE_FORMAT_UNKNOWN_AFTER_MOUNT = "disk has wrong format hence not cacheable"
- DM2_MGMT_STATE_DEACTIVATED = "disk has been deactivated"; DM2_MGMT_STATE_ACTIVATED = "disk has been activated";
- DM2_MGMT_STATE_IMPROPER_UNMOUNT and DM2_MGMT_STATE_IMPROPER_MOUNT = "soft disk error, try to clear"
- DM2_MGMT_STATE_CACHE_RUNNING = "cache running";
- DM2_MUST_FORMAT = "Disk Cache Enable Failed - Disk Cache must be formatted before enabling"
- DEFAULT = "unknown state, please try again a little later";

These messages may display in a log:

- NKN_DM2_DISK_ADMIN_ACTION: A previous command has been given to cause the disk to be in the "cache enabled" state.
- NKN_DM2_EMPTY_CACHE_TIER: A request was sent to a disk tier which has no enabled disks.
- NKN_DM2_EVICTION_SKIPPED: An eviction request was made but the request had some type of error in it.
- NKN_DM2_WRONG_CACHE_VERSION: Attempting to enable a cache which has the wrong version number. This should never happen because older versions should be converted to the current version. Media Flow Controller doesn't downgrade cache versions.
- NKN_DM2_DISK_CACHE_NOT_FOUND: An unknown disk cache name was given to the CLI, and Media Flow Controller returns this error.
- NKN_DM2_INVALID_MGMT_REQUEST: The command attempted was not valid for the given state of the disk. This is a generic error.
- NKN_DM2_DISK_DEVICE_NOT_FOUND: During a 'status active' command, the system could not find the requested device.
- NKN_DM2_CONVERT_FAILED: When Media Flow Controller is starting or a drive is made 'status active', a possible format conversion is done. If that conversion fails, Media Flow Controller returns this error.
- NKN_DM2_MUST_CACHE_DISABLE: If a drive is in the 'cache enabled' state and a 'status inactive' command is given, we return this error.
- NKN_DM2_MUST_CLEAR: If a drive is in an error state of any kind and a 'cache enable' command is given, Media Flow Controller returns this error.
- NKN_DM2_MUST_STATUS_ACTIVE: If a drive is in the 'status inactive' state and a 'cache enable' command is given, we return this error.

Replacing Bad Disks

In order to replace a bad disk, first disable and inactivate the disk, and then add a new disk. This procedure uses **dc_bad** for the disk name; replace appropriately.

1. Discover the bad disk's name.
`show media disk list`
2. Disable the bad disk by name.
`media-cache disk dc_bad cache disable`
3. Make the bad disk inactive.
`media-cache disk dc_bad status inactive`
4. Pull the drive, put in a new drive and mount the new disk.
`media-cache disk mount-new`
5. Find the inactive disk's name.
`show media-cache disk list`
6. Activate the new disk.
`media-cache disk dc_new status active`
7. Format the new disk; takes approximately 5 minutes.
`media-cache disk dc_new format`
8. Enable the new disk.
`media-cache disk dc_new cache enable`



CAUTION: This procedure will not work when adding a new disk or disks to a Juniper Networks VXA Series to a slot that was never configured. In such a case, follow the instructions in the VXA Series hardware guides for adding a new disk using the Adaptec Utility, or see [“Inserting New Disks into a VXA Series Media Flow Engine” on page 109](#).

Correcting Mis-Labeled Disk Types

Occasionally, SSD disk types may be mistakenly labeled as SATA disk types. If this happens, you can correct the problem by re-labeling the disks.

To correct the disk type as SSDs for those disks mistakenly labeled as SATA:

1. Run this command and note down the label for the SATA disks; for example **dc_n**.
`show media-cache disk list`
2. Disable any mis-labeled disks; repeat as needed:
`media-cache disk <dc_n> cache disable`
3. Configure the correct disk type for all mis-labeled disks; repeat as needed:
`media-cache disk <dc_n> disk-type SSD`
4. Restart the delivery service:
`service restart mod-delivery`
5. Wait for 1 minute. Then, format the re-labeled disks; repeat as needed:
`media-cache disk <dc_n> format`
6. Enable the re-labeled disks; repeat as needed:
`media-cache disk <dc_n> cache enable`
7. Save your work:
`write memory`

Inserting New Disks into a VXA Series Media Flow Engine

The task [“Replacing Bad Disks” on page 108](#) is incorrect for VXA Series hardware when adding one or more new disks to a slot that was never used (configured) before because Media Flow Controller only searches for new disks at manufacture time. Use this procedure instead.

To add a disk, or disks, on a VXA Series Media Flow Engine to an un-configured slot:

1. From Enable or Configuration mode in the CLI, check the current drive list; you may want to note the drives you have installed:

```
show media-cache disk list
```

2. Insert the new disks and reboot:

```
reload
```

3. When you are prompted to open the Adaptec Utility, press **Ctrl+a** to open the utility. The Adaptec Utility menu is displayed.

4. Select **Array Configuration Utility**.

The Configuration Change confirmation window is displayed.

5. Select **Accept**.

The Configuration Utility Main menu is displayed.

6. Use the arrow keys to select **Initialize Drives**, and press **Enter**.

A list of discovered drives is displayed.

7. Select the drive or drives that you added, and press **Enter**.

A warning message is displayed.

If you are unsure which drives to select, you can use the **Esc** key to exit to the Adaptec Utility menu and use the **Disk Utilities** function to identify the drives.

8. Enter **Yes**.

An **Initializing is Done** message is displayed.

9. Press any key to continue.

The Configuration Utility Main menu is displayed.

10. Select **Create JBOD**, and press **Enter**.

A drives list is displayed.

11. Select the new drives, and press **Enter**.

A confirmation window is displayed.

12. Enter **Yes**.

The Adaptec Utility menu is displayed.

13. Press **Esc** to exit the Adaptec Utility.

The system reboots automatically.

14. Using the Media Flow Controller CLI, mount the new disks and activate them:

```
media-cache disk mount-new
show media-cache disk list
media-cache disk <disk_name> activate
```

15. Verify the new disks are properly mounted:

```
show media-cache disk list
```



NOTE: The documented disk replacement procedure for replacing a bad disk on VXA Series works correctly provided that Media Flow Controller can still identify the disk (**show media-cache disk list**).

Related Topics

- [“Before You Configure Media Flow Controller” on page 82.](#)
- [“Media Flow Controller Minimum System Requirements” on page 38.](#)

Installing and Using FMS in Media Flow Controller (CLI)

Media Flow Controller has the ability to work with Adobe Flash Media Server (FMS) to stream Flash videos over Real Time Messaging Protocol (RTMP). Before you configure FMS in Media Flow Controller, see [“Before You Configure Media Flow Controller” on page 82](#). See **application** for CLI details. See Adobe Flash Media Server documentation for administering the server once installed.

FMS RTMP delivery service listens on Transmission Control Protocol (TCP) port 1935; the FMS administration service listens on TCP port 1111. RTMP is a proprietary protocol developed by Adobe Systems for streaming audio, video and data over the Internet, between a Flash player and a server.

- [Installing FMS on Media Flow Controller \(CLI\)](#)
- [Modifying and Restarting the FMS Service \(CLI\)](#)
- [Configuring the FMS Admin Console—First Time \(CLI\)](#)
- [Configuring FMS on Media Flow Controller for Video On Demand \(CLI\)](#)
- [Applying the Adobe Full-Function FMS Server License \(CLI\)](#)

Installing FMS on Media Flow Controller (CLI)

To install FMS, you must first download the image; you can do this directly from Adobe (the “Development Server” version, available for free, is adequate for this procedure, but it is limited to 10 simultaneous streams). First download the image to a Web or FTP/SCP server to which you have access. Use these CLI commands at the **(config) #** prompt (must have admin privileges and enter **enable** and then **configure terminal** first):

```
application fms download <URL_of_FMS_image>
application fms install <name_of_FMS_image>
```

The **download** command downloads the image given in the URL (SCP, HTTP, or FTP) to the directory **/nkn/adobe/downloads** directory that is visible in the FMS shell. See [“Terminology” on page 31](#) for the **scp** URL format).

The **install** command shows files in the FMS download directory and, when the install image is given, interactively installs FMS.



CAUTION: It is critical that a few of the installation questions are answered with the correct values; be sure to use these values as indicated in the following step procedure.

1. Accept the license agreement and enter your FMS server serial number or leave blank and press **Enter** to install the FMS Development Server. The FMS Development Server is free, but limited to 10 simultaneous streams.

```
Do you agree with the license agreement? (y/n): y  
Please enter your Flash Media Server 3.5 serial number.
```

```
You have not entered a serial number. Falling back to the Adobe Flash  
Media Development Server!
```

```
Would you like to try again? y/n: Default [n]: n  
Adobe Flash Media Server 3.5 requires approximately 200MB of disk space.
```

2. Enter the correct directory for the installation, **/nkn/adobe/fms**.

```
The installer will install Adobe Flash Media Server 3.5 in the following  
directory. Default [/opt/adobe/fms]: /nkn/adobe/fms
```

3. If you get a previous installation warning, overwrite it.

```
WARNING: The installer has detected a previous installation.  
Do you want the installer to overwrite the previous installation with this  
installation or quit this installer? (y/q) Default [y]: y
```

4. Enter the port on which the FMS is to communicate, **1935**.

```
The Adobe Flash Media Server communicates on the IANA-assigned port of  
1935, which is the port most Flash applications expect, and can also  
communicate on port 80, both for tunneling Flash over HTTP, and for  
proxying HTTP to a webserver.
```

```
Please enter the Adobe Flash Media Server port(s), comma-separated Default  
[1935,80]: 1935
```

5. Enter the port on which the FMS Admin service is to run, **1111**.

```
Please enter the port to use for the Admin service. You can only specify  
one admin port. Default [1111]: 1111
```

6. Enter credentials for using the FMS Admin Console. Example uses **admin / admin**.

```
The administrative user name and password you provide here is required to  
use the Adobe Flash Media Server Management Console for administration,  
monitoring, and debugging.
```

```
Please enter the administrative username: admin  
Please enter the administrative password:  
Please do not enter a blank password.  
Please enter the administrative password: *****  
Confirm password: *****
```

7. Enter a username for the user the FMS is to run as; Media Flow Controller requires this to be **admin**.

```
When the Adobe Flash Media Server service is started, the service can be  
run as a user other than "root". The server would change to this user when  
the server is started and has acquired its ports.
```

```
Please enter the user that the Adobe Flash Media Server service will run  
as Default user [nobody]: admin
```

8. Enter a group for the FMS service user; Media Flow Controller requires this to be **admin**.

```
Please enter a valid user group for the "admin" user: Default group  
[admin]: admin
```

9. Decline Apache installation.

Do you want to install apache? (y/n) Default [y]: **n**

10. Decline FMS running as a daemon.

Do you want the Adobe Flash Media Server service to run as a daemon? (y/n)
Default [y]: **n**

11. Decline starting FMS after this installation.

Do you want to start the Adobe Flash Media Server after the installation is done? (y/n) Default [y]: **n**

```
----- Install Action Summary -----
WARNING: You have chosen to overwrite a previous installation.
Installation directory           = /nkn/adobe/fms
Flash Media Server Port          = 1935
Flash Media Admin Server Port    = 1111

Administrative username          = admin
Administrative password          = (suppressed)

service owner                    = admin
service user                     = admin
service group                    = admin
```

12. Finish the installation.

Proceed with the installation? (y/n/q): **y**

Modifying and Restarting the FMS Service (CLI)

You must manually configure two FMS files. If any configuration changes are made to FMS, including installation, the FMS services need to be restarted.

1. Modify the FMS **server** and **adminserver** files using **vi**. To do this, first log in to the FMS shell from the CLI **config** mode in Media Flow Controller:

```
application fms shell
```

2. Locate the **server** and **adminserver** files and edit them as indicated; for this line:

```
if [ "$USERID" != "Xroot" ]; then
```

Change to:

```
if [ "$USERID" != "Xadmin" ]; then
```

Example:

```
MFC # enable
MFC # configure terminal
MFC (config) # application fms shell
BusyBox v1.00 (2008.12.21-08:07+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

# ls
fms lib bin
# cd fms
# ls
documentation          tcSrvMsg                fmsadmin.pid
applications            uninstallFMS            modules
auto_fms_start_enabled fmsmgr                  tools
shmr                    logo.gif                dh1024.pem
```


| | | |
|----------------------------|-----------------------|----------------------------|
| <code>fmsmaster.pid</code> | <code>logs</code> | <code>scriptlib</code> |
| <code>licenses</code> | <code>conf</code> | <code>fmscore</code> |
| <code>fmsmaster</code> | <code>fmsadmin</code> | <code>libasneu.so.1</code> |
| <code>server</code> | <code>webroot</code> | <code>readme.htm</code> |
| <code>License.htm</code> | <code>samples</code> | <code>dh512.pem</code> |
| <code>touch</code> | <code>tmp</code> | <code>adminserver</code> |
| <code>License.txt</code> | <code>fmsedge</code> | |

- Exit the FMS shell by typing **exit**; the window closes and you must open a new session to Media Flow Controller.

- In the new Media Flow Controller session, enter **config** mode and restart FMS:

```
service restart mod-rtmp-admin
service restart mod-rtmp-fms
```

Configuring the FMS Admin Console—First Time (CLI)

Use the FMS Admin Console to monitor and manage FMS activity, including the current FMS user sessions, streams, and performance. The Media Flow Controller Web-based Management Console lets you monitor only the delivery of the files configured under VOD.



NOTE: Do not use your Media Flow Controller as a Web server for the FMS Admin Console.

To use the FMS Admin Console to monitor FMS activity, you need to copy three files under the FMS installation directory to the doc root of your origin-server; We recommend placing these files in a directory separate from your FMS video files directory.

- To obtain the three files, download the FMS Development Server to your origin-server and unpack it.
- Locate the three files you need in the **fms/webroot** directory of the unpacked FMS Development Server tarball:

```
/fms_adminConsole.htm
/resources/AC_RunActiveContent.js
/swfs/fms_adminConsole.swf
```

Copy these three files to your doc root (`/var/www/html`) in a directory you create; for example **fmsAdmin**.

- Open a browser to your Media Flow Controller using this URL (example uses **fmsAdmin**):

```
http://<IP_address_of_origin-server>/fmsAdmin/fms_adminConsole.htm
```

- Log in with the FMS Admin Console credentials you configured in step 6 of [“Installing FMS on Media Flow Controller \(CLI\)” on page 110](#) and the IP address of your Media Flow Controller for **Server Address**.
- Complete FMS setup on Media Flow Controller by following the instructions given in [“Configuring FMS on Media Flow Controller for Video On Demand \(CLI\).”](#)

The advantage to installing the FMS Admin Console before FMS configuration is complete is that, when FMS configuration on Media Flow Controller is complete, you can immediately check the installation via the FMS Admin Console. Otherwise, you must send appropriately defined traffic and verify FMS processing through the Media Flow Controller. You can do this

either by looking at the logs and graphs in the Media Flow Controller Web-based Management Console or by using **show counters** at the CLI to observe FMS traffic.

Configuring FMS on Media Flow Controller for Video On Demand (CLI)

To configure FMS for video on demand (VOD), log in to the FMS shell from the CLI **config** mode in Media Flow Controller:

```
application fms shell
```

This CLI command takes you to the FMS installation directory and from there, all FMS configuration changes can be done. The configuration entails creating a Media Flow Controller FMS directory and namespace or namespaces.

When an RTMP request comes to FMS, it scans the configuration file in the application directory **/fms/applications** to find the requested file. You configure Media Flow Controller **namespace** to correspond with the configuration of VOD_DIR.



NOTE: The configuration in FMS must match your FMS Media Flow Controller **namespace** configuration.

- [Configuring FMS VOD \(CLI\)](#)
- [Configuring Namespace for FMS VOD \(CLI\)](#)
- [Using Video Directories for FMS](#)

Configuring FMS VOD (CLI)

Assuming we are using the default FMS VOD application:

1. Locate the **Application.xml** file in the **/fms/applications/vod** directory and replace **/\${VOD_DIR}** with **/nkn/mnt/fuse/fms** in the configuration for **VirtualDirectory**. The **VirtualDirectory** configuration part of **Application.xml** looks like this (before editing):

```
<VirtualDirectory>
<!-- Specifies application specific virtual directory mapping for recorded
<Streams>/;${VOD_COMMON_DIR}</Streams>
<Streams>/;${VOD_DIR}</Streams>
</VirtualDirectory>
```

Edit this as indicated. For this line:

```
<Streams>/;${VOD_DIR}</Streams>
```

Change to:

```
<Streams>/;/nkn/mnt/fuse/fms</Streams>
```

Example:

```
MFC (config) # application fms shell
BusyBox v1.00 (2008.12.21-08:07+0000) Built-in shell (ash)
Enter 'help' for a list of built-in commands.

# ls
lib          fms          downloads   bin
# cd fms
# ls
auto_fms_start_enabled  logs          readme.htm
fmsmgr                  License.txt   tcSrvMsg
```

```

server                dh1024.pem           fmsadmin
fmsmaster.pid        shmrtd               tmp
uninstallFMS        modules              logo.gif
webroot              fmsedge              applications
libasneu.so.1        tools                 samples
licenses             fmscore              documentation
conf                 License.htm          adminserver
dh512.pem            scriptlib
fmsmaster            fmsadmin.pid
# cd applications
# ls
vod    live
# cd vod
# ls
main.far                media                readme.txt
allowedHTMLdomains.txt Application.xml        allowedSWFdomains.txt
#

```

2. Exit the FMS shell by typing **exit**; the window closes and you must open a new session to Media Flow Controller.

The directory configuration is **/nkn/mnt/fuse** *PLUS* the URI path **/fms** from the **namespace** configuration. If the file requested by the client player for application VOD is **sample.flv**, then the request that comes to Media Flow Controller is **/fms/sample.flv**. Media Flow Controller matches the incoming request **/fms/sample.flv** to namespace **fms_vod** via the configured **match uri** of **/fms**.

Configuring Namespace for FMS VOD (CLI)

The **namespace** and **match uri** must match the application configuration in FMS. For example, if you used **/nkn/mnt/fuse/fms/vod** instead of **/nkn/mnt/fuse/fms**, then your corresponding **namespace** configuration would have to have a **match uri** of **/fms/vod**. Example assumes a configuration of **/nkn/mnt/fuse/fms**.

The video files and associated files such as SWFs, JavaScript, and HTML are stored in an HTTP or NFS origin server and the configured Media Flow Controller **namespace** points to that origin server.

The files also can be pre-ingested using FTP instead of fetching from origin on cache-miss.

It's also possible to cache only the video file in Media Flow Controller and for all other associated files, the player directly goes to the origin server.

Example:

```

MFC (config) # namespace fms_vod
MFC (config namespace fms_vod) # domain any
MFC (config namespace fms_vod) # match uri /fms
MFC (config namespace fms_vod) # origin-server http fms-origin.test.com
MFC (config namespace fms_vod) # status active
MFC (config namespace fms_vod) # exit

```

Configuring Multiple Namespaces and URIs for VOD—Method 1

Have multiple applications such as **vod**, **vod1**, and **vod2**, in FMS and let them map to their corresponding namespaces as given:

vod:

Application.xml: **<Streams>/;nkn/mnt/fuse/firstpath</Streams>**

namespace match uri: **/firstpath**

Player1 requests file: **application vod** and file **sample.flv**

vod1:

Application.xml: **<Streams>/;nkn/mnt/fuse/secondpath</Streams>**

name space match uri: **/secondpath**

Player2 requests file: **application vod1** and file **sample.flv**

vod2:

Application.xml: **<Streams>/;nkn/mnt/fuse/thirdpath</Streams>**

name space match uri: **/thirdpath**

Player3 requests file: **application vod2** and file **sample.flv**

Configuring Multiple Namespaces and URIs for VOD—Method 2

Have only one application and use the client player to provide different file paths that map to different namespaces.

vod:

Application.xml: **<Streams>/;nkn/mnt/fuse</Streams>**

player1 requests file: **firstpath/sample.flv**

player2 requests file: **secondpath/sample.flv**

player3 requests file: **thirdpath/sample.flv**

Using Video Directories for FMS

Put your videos in the Apache webroot (`var/www/html`) in a directory named **fms** (or whatever you used for the URI path configuration in the Application.xml file) of the origin server configured for your streaming namespace. Provide access to them on your Website. You can then open a browser and request a video via RTMP and observe FMS on Media Flow Controller either by looking at the logs and graphs in the Media Flow Controller Web-based Management Console or by using **show counters** at the CLI.

Applying the Adobe Full-Function FMS Server License (CLI)

If you installed FMS on Media Flow Controller using the free FMS Development Server, and decide to upgrade to the full-function Adobe FMS Server, you can obtain a full-function FMS Server license, **juniper-mfc-fms**, from Juniper Networks. Without the full-function FMS Server license, the functionality is limited to 10 simultaneous streams. To obtain the FMS Server license, see [“Requesting Technical Support” on page 29](#).

To install the full-function FMS Server license obtained from Juniper Networks:

1. Install the license:

```
license install <license_key>
```

2. Verify that the license is installed correctly:

```
show license
```

3. Restart the FMS service:

```
service restart mod-rtmp-fms
```

Related Topics

- [“Before You Configure Media Flow Controller” on page 82.](#)

Administering Media Flow Controller Overview (CLI)

You can perform standard Media Flow Controller administrative tasks using the CLI.

- [Checking Media Flow Controller Version and Status](#)
- [Saving and Applying Configurations, Resetting Factory Defaults \(CLI\)](#)
- [Rebooting Media Flow Controller \(CLI\)](#)
- [Upgrading Media Flow Controller \(CLI\)](#)
- [Configuring the Web Interface \(CLI\)](#)
- [Configuring the Web Interface Proxy \(CLI\)](#)

Checking Media Flow Controller Version and Status

You can check the release version and system status and other conditions with the show version command. Example:

```
MFC (config) # show version
Copyright (c) 2008-2010 by Juniper Networks, Inc

Product name:      mfc
Product release:   mfc-2.1.0-dv
Build ID:          302_12730_216
Build date:        2010-10-19 10:31:36
Target arch:       x86_64
Built by:          build@build03

Uptime:           6m 11s

Product model:     standard
Host ID:           D0F4D91
System memory:     921 MB used / 1088 MB free / 2009 MB total
Swap:              380 MB used / 648 MB free / 1028 MB total
Number of CPUs:    2
CPU load averages: 1.86 / 1.33 / 0.59
```

Saving and Applying Configurations, Resetting Factory Defaults (CLI)

You can save a binary file with all current configuration data, that can be used to restore the system configuration. You can also reset custom configurations to their factory defaults, upload a saved configuration, and import a configuration from another Media Flow Controller.

To save and apply a configuration using the CLI:

1. Save a configuration to a file; use **no-switch** to leave the current configuration active. Use **show configuration files** to see the saved file name.
`write memory to <file_name> no-switch`
2. Use SCP to send the just-saved configuration file to a server (must have SCP installed); See [“Terminology” on page 31](#) for the **scp** URL format).
`configuration upload <file_name> <URL>`
3. When you are ready, fetch the saved configuration file.
`configuration fetch <URL>/<file_name>`
4. Verify that you have the saved configuration file.
`show configuration files`
5. Switch to the saved configuration.
`configuration switch-to <file_name>`

Example:

```
MFC (config) # write memory to 04_01_09 no-switch
MFC (config) # show configuration files
04_01_09
initial (active)
initial.bak

MFC (config) # configuration upload 04_01_09 scp://joe@example.com/home/
joe
Password: *****
MFC (config) # configuration delete 04_01_09
MFC (config) # show configuration files
initial (active)
initial.bak

MFC (config) # configuration fetch scp://joe@example.com/home/joe/
04_01_09
Password: *****

MFC (config) # show configuration files
04_01_09
initial (active)
initial.bak

MFC (config) # configuration switch-to 04_01_09
MFC (config) # show configuration files
04_01_09 (active)
initial
initial.bak
```

6. Merge the common settings from a given configuration file to the active configuration file.
`configuration merge <file_name>`
7. Revert the active configuration to either the factory defaults or the last saved configuration. Use **keep-basic** to preserve licenses and SSH host keys, use **keep-**

connect to preserve anything necessary to maintain network connectivity to the system: interfaces, routes, and ARP; either or both may be used.

```
configuration revert {factory | saved} [keep-basic] [keep-connect]
```

Rebooting Media Flow Controller (CLI)

You can either reboot or shutdown Media Flow Controller; if you use **shutdown**, the system does not reboot until it is power-cycled.

To reboot using the CLI:

1. Reboot or shut down the system.
`reload`
2. Set boot parameters; optionally specify a default location from which the image boots; there are only two locations to choose from so the options are **1** and **2** for location ID. If **next** is used, set the boot location to be the next one after the one currently booted from.
`boot {location <location_ID> | next}`
3. View boot parameters.
`boot ?`
4. View current settings.
`show boot`

Upgrading Media Flow Controller (CLI)

When upgrades are available, Juniper Networks will broadcast the upgrade URL to use in this procedure. Upgrade preserves the current, saved, configurations; however, you may still want to save the current configuration to a file on another system by following the previous procedure.



NOTE: If disk partitions are changing then a re-manufacture is required; follow [“Saving and Applying Configurations, Resetting Factory Defaults \(CLI\)” on page 118](#) steps along with the appropriate installation instructions described in the *Media Flow Controller Installation Guide*.

To upgrade using the CLI:

1. Fetch the upgrade image file with the supplied URL.
`image fetch <URL>/<filename>`
2. Install the image.
`image install <filename>`
3. Verify which boot image contains the upgrade.
`show images`
4. Switch to the boot partition containing the upgrade image, if needed.
`image boot next`
5. Reboot to that partition.
`reload`
6. Verify that the new image is booted.
`show version`

Configuring the Web Interface (CLI)

Before you configure the Media Flow Controller Web interface, see [“Before You Configure Media Flow Controller” on page 82](#).

Configure Web interface settings. The **web** command lets you specify time-outs, ports, and protocols for access to the Media Flow Controller Web interface, also referred to as the Management Console. See **web** for CLI details. We highly recommend setting the Web interface port to something other than 80 as that port is used for service traffic; port 8080 is the default, and recommended for Web interface access.

To configure the Web interface:

1. The Web interface (Management Console) is **enabled** by default. If you need to disable it or re-enable it.

```
web no enable
web enable
```

2. Set the automatic logout when the Web interface is idle; default is **15** minutes.

```
web auto-logout <number_of_minutes>
```

3. Enable HTTP, HTTPD, and HTTPS and set a port or listen interface (HTTPD only); all are **enabled** by default. Default HTTP port is **8080**, default HTTPD listen interface is **eth0** (used for management), default HTTPS port is **443**.

```
web http enable
web http port <port_number>
web httpd listen enable
web httpd listen interface <interface_name>
web https enable
web https port <port_number>
```

4. Configure Web session cookie options; **renewal** is the length of time before Web session cookies are automatically regenerated, default is **30 minutes**; **timeout** is the time after which a session expires, default is **900 seconds** or **15 minutes**.

```
web session renewal <number_of_minutes>
web session timeout <number_of_minutes>
```

Example:

```
MFC (config) # web auto-logout 9000
MFC (config) # web http port 8080
MFC (config) # web httpd listen interface eth0
MFC (config) # web https port 443
MFC (config) # web session renewal 60
MFC (config) # web session timeout 9000
MFC (config) #
```

Configuring the Web Interface Proxy (CLI)

Configure Web interface proxy settings if your Web server uses a proxy. The **web proxy** command lets you specify authentication for access to the Media Flow Controller Web interface, and a host for the proxy.

The Web **proxy auth** options do not take effect without a configured Web proxy host. See **web** for CLI details.

1. Set Web proxy authentication options; **basic** is HTTP basic authentication. Default is **none**. If you set the **auth** type to **basic**, then use **auth basic** to set the **password** and **username** to be authenticated.

```
web proxy auth authtype {none | basic}
web proxy auth basic password <plaintext_password>
web proxy auth basic username <username>
```

2. Set the Web proxy **host** and, optionally, **port** (default is 1080). Setting the Web proxy host enables the Web proxy. If set, this proxy accepts HTTP and FTP downloads (FTP default port is 21).

```
web proxy host <IP_address> [port <TCP_port>]
```

Example:

```
MFC (config) # web proxy auth authtype basic
MFC (config) # web proxy auth basic password 123
MFC (config) # web proxy auth basic username admin
MFC (config) # web proxy host 123.45.10.9 port 8080
MFC (config) #
```

Related Topics

- [“Before You Configure Media Flow Controller” on page 82.](#)

Configuring Caching All Contents for a Website (CLI)

Configure Media Flow Controller to cache all contents of the Website **www.example.com**. Change variables, like **domain** and **origin-server**, as needed. Before you configure Media Flow Controller to cache all contents of a Website, see [“Before You Configure Media Flow Controller” on page 82](#) and [“Media Flow Controller System Configuration Overview \(CLI\)” on page 88](#). This example does not include Media Flow Controller system set up.

To configure caching of all contents for a Website:

1. Set up DNS records or Hosts entries so incoming requests are routed to Media Flow Controller. Be sure to set the Media Flow Controller DNS server so it is not the same as your Web server's DNS server (to prevent looping).
2. Configure a **namespace** with **domain www.example.com** and **match uri-prefix /** (slash), and an **origin-server**; make the namespace **active**:

```
MFC (config) # namespace example
MFC (config namespace example) # domain www.example.com
MFC (config namespace example) # match uri /
MFC (config namespace example) # origin-server http www.exampleOS.com
MFC (config namespace example) # status active
MFC (config namespace example) # exit
```

The **match uri-prefix** of just a **/** (slash) tells Media Flow Controller to use these namespace rules for all incoming requests to **domain www.example.com**, since all incoming requests have a **/** (slash) in them.

Tip! If unsure what port your **origin-server** is using, use standard Linux shell commands (for example, **netstat -nl**) to figure out the port, and then configure it along with the **origin-server**, if not the default. If you need to change the **origin-server**, or any **namespace** setting, simply enter the new setting.

There are many **namespace** options, including **cache-inherit**, **delivery protocol**, **origin-**

fetch, and so forth. See [Chapter 6, “Configuring Namespaces \(CLI\)”](#) for more information and **namespace** for CLI details.

- Repeat the configuration twice, if needed, to cover requests coming in to “example.com” (without the “www” prefix) and the IP address for example.com (if requests may come in that way).

- Verify the configuration.

```
show namespace example
```

```
Namespace: example
  Active: yes
  Precedence: 0
  Domain Name: www.example.com
  Proxy Mode: reverse
  Match Type:
    URI-Prefix - /
  Origin-Server: http://www.exampleOS.com:80
  Delivery Protocol: HTTP           Status Enabled: yes
  Origin Fetch Configuration:
    Cache-Age (Default): 28800 (seconds)
    Cache Age Threshold: 60 (seconds)
    Cache-Directive: follow
    Object Size Threshold: NONE (Always Cached)
    Modify Date Header: deny
  Origin Request Configuration:
    Cache-Revalidate: permit
    Use 'Date' Header when Last-Modified is not present: no
    Convert HEAD to GET: permit
    Host-header Inherit: deny
    Set X-Forwarded-For Header : yes
  Client-Request Configuration:
    Allow objects with a query-string to be cached: no
  Client-Response Configuration :
  Delivery Protocol: RTSP           Status Enabled: no
  Origin Fetch Configuration:
    Cache-Age (Default): 28800 (seconds)
    Cache Age Threshold: 60 (seconds)
    Cache-Directive: follow
    Object Size Threshold: NONE (Always Cached)
  Virtual Player:
  Live Pub-Point Details:
  Pre-stage FTP Configuration:
    User: example_ftpuser
```

- From the browser, initiate a connection to Media Flow Controller for **www.example.com**. Check the accesslog to see that Media Flow Controller processed it. You can view the errorlog and the accesslog via the Media Flow Controller Web interface; use a browser to open the Media Flow Controller IP address on port 8080 and login (default login: **admin** / no password) and open the **Service Logs** tab.

CHAPTER 5

Configuring Virtual Players, Media Fetch and Pre-Staging (CLI)

- [Creating and Configuring Virtual Players \(CLI\)](#)
- [Configuring YouTube Video Caching \(CLI\)](#)
- [Configuring SmoothStream Video Caching \(CLI\)](#)
- [Configuring FlashStream Video Caching \(CLI\)](#)
- [Configuring NFS Fetch for Images \(CLI\)](#)
- [Configuring HTTP Fetch for Videos \(CLI\)](#)
- [Configuring RTSP Fetch for Videos \(CLI\)](#)
- [Pre-Staging Content with FTP \(CLI\)](#)

Creating and Configuring Virtual Players (CLI)

Create virtual players to customize how videos are delivered; after created, they are assigned to a namespace. This is optional; if a namespace does not have a defined virtual player assigned to it, it uses the **network connection** settings. You may want to create a virtual player for each type of video you deliver; for example, if you deliver YouTube® videos, create a **type youtube** virtual-player for use in the corresponding **namespace**. Media Flow Controller virtual players support and complement client-side video players. Whereas namespaces allow you to define what gets fetched from where and how, virtual players let you fine-tune video delivery.

There are six types of virtual players: Type **generic** virtual player offers all generic virtual player options; Type **qss-streamlet** virtual players allow multiple settings of assured flow via a rate-map, Type **yahoo** players are for YouTube videos and include a health-probe option and a special authentication option. Type **youtube** players are for YouTube media. Type, **smoothstream-pub** supports Smoothstream (Microsoft IIS). Type **flashstream-pub** supports Flashstream videos.

There are six types of virtual players, described in this section:

- **Virtual Player Type generic** has a super-set of delivery options including hash verification of data.
- **Virtual Player Type break** has a sub-set of virtual player type **generic** delivery options.
- **Virtual Player Type qss-streamlet** lets you create an assured flow rate map to apply different delivery rates per URI.
- **Virtual Player Type yahoo** provides a subset of delivery options plus special health and hash verification options.

- **Virtual Player Type smoothflow** provides functionality for Media Flow Controller adaptive bit-rate delivery.
- **Virtual Player Type youtube** provides a subset of delivery options and YouTube-specific options for caching YouTube videos and to provide trick-play functions.
- **Virtual Player Type smoothstream-pub** supports publishing Smooth Streaming format ISMV files to fragmented format for media delivery to Silverlight player without using an IIS Windows Server.



NOTE: The Break **virtual-player** is deprecated in Release 2.1. All Break virtual player features are supported in the Generic type **virtual-player**.

NOTE: The Smoothflow **virtual-player** is deprecated in Release 2.1; all Smoothflow functionality is deprecated.

See also the following sections before beginning your virtual player configuration:

- [Using assured-flow](#)
- [Using query-string-param](#)
- [Using hash-verify](#)
- [Using virtual-player type qss-streamlet rate-map](#)
- [Example: Configuring generic Virtual Players \(CLI\)](#)

Additionally, these sections provide detailed, deployment-specific virtual player configurations:

- [“Configuring YouTube Video Caching \(CLI\)” on page 131](#)
- [“Configuring SmoothStream Video Caching \(CLI\)” on page 134](#)
- [“Configuring FlashStream Video Caching \(CLI\)” on page 135](#)



NOTE: In Media Flow Controller Release 2.0.7, the **show options ?** (question mark), lists all virtual player options no matter what virtual player type you are configuring; however, if you try to set an option that does not apply to that player type, an error is displayed.

Virtual Player Type generic

The **type generic** (formerly type 0) virtual player provides options for nearly all virtual player functions.

- **assured-flow**—Assure that content is delivered at the specified rate (AFR), but no more than the configured **connection max-bandwidth**. AFR is defined in a kilobits per second **rate**, or signaled with a query param.
- **connection max-bandwidth**—Maximum allowable bandwidth for a session. The actual session bandwidth used does not exceed this value, even if there is available bandwidth in the link. When it is a full download, Media Flow Controller tries to allocate this value to the session.
- **fast-start**—Deliver the 1st set of kilobytes at either the configured maximum session speed or the detected available bandwidth; or by a value determined by one of the **fast-start** options: either a static **size** value, or a **query-string-param** name.

- **full-download**—Allow the delivery to download at the fastest possible speed, up to the configured **connection max-bandwidth** and possibly exceeding the configured **assured-flow rate**.
- **hash-verify**—Verify the authorization hash value specified in the URL query string.
- **seek**—Implement seek (for FLV and MP4 media files) based on the value of **query-string-parm**. This function allows the client player to seek a specific part of the media content; for example, to jump ahead a few minutes or go back a few minutes in the video.

Virtual Player Type break

The **type break** (formerly type 1) virtual player offers a subset of Media Flow Controller delivery options: **assured-flow**, **connection max-bandwidth**, **fast-start**, **hash-verify**, and **seek** options (see [Virtual Player Type generic](#) for details).

Virtual Player Type qss-streamlet

The **type qss-streamlet** (formerly type 2) virtual player includes **connection max-bandwidth** (see [Virtual Player Type generic](#) for details) and AssuredFlow via a special **rate-map** attribute. The **rate-map rate** argument extracts the value from the URL to calculate the assured flow rate needed for each HTTP request. By default, the **match** string (length must be 2 bytes) is extracted by going to the end of the URL and skipping 12 bytes from the end. The value in that location is mapped to the configured rate in kbps. Example:

```
http://video.example.com/public/BBB87026/xy_750_1938344/  
AC60E15B2A7C4A45AC4C1472E2AC0816_030000003F.flv
```

In the URL, the value **03** (12 bytes from the end of the URL) is extracted, and the corresponding assured flow rate (1000 Kbps) is applied.

Virtual Player Type yahoo

The **type yahoo** (formerly type 3) virtual player includes **assured-flow**, **connection max-bandwidth**, and **seek** options (see [Virtual Player Type generic](#) for details) as well as these special options:

- **health-probe**—Configure an external server to do health checks by making Media Flow Controller fetch data from origin and play it to the server initiating the health check. The signal that a given HTTP request is for a health probe is the **health-probe query-string-parm name**. If that **name** value matches the following **<string>** value, the GET request is treated as a health probe. When servicing health probes, Media Flow Controller does not cache the data into disk or in buffer. Use **virtual player <name> type 3 no health-probe** to disable.
- **req-auth**—Compute MD-5 hash of query string parameters representing **stream-id**, **auth-id**, a configured **shared-secret**, and **time-interval**; and match the computed value with the specified **match query-string-parm <string>**. The HTTP GET proceeds if the computed MD-5 hash matches; if there is no match, the session is rejected. Use **virtual player <name> type 3 no req-auth** to disable.

Virtual Player Type smoothflow

The **type smoothflow** (formerly type 4) virtual player configures SmoothFlow. It is a requirement of SmoothFlow that this virtual player be configured and used through a configured namespace. The Type 4 virtual player includes **connection max-bandwidth**,

hash-verify, and **seek** options (see [Virtual Player Type generic](#) for details) as well as these special options:

- **control-point**—Specify either **server** or **player** for smooth flow signaling. If **server**, then Media Flow Controller detects the bandwidth variations at the client side and adjusts the bit-rate of the video accordingly. If **player**, then the player at the client side explicitly signals the bandwidth changes and Media Flow Controller adjusts the bit-rate of the video accordingly.
- **signals**—Configure triggers for delivery functions; use **virtual player <name> type 4 no signals** to disable.
 - **session-id query-string-param**—Specify a **query-string-param** name to set the **session ID**; default is **sid**. Session ID is the way you bind the control session that signals the bandwidth change with the data channel.
 - **state query-string-param**—Specify a **query-string-param** name to set SmoothFlow **state**; default is **sf**. The values this query param take signal various function calls to SmoothFlow.
 - **profile query-string-param**—Specify a query param name to set the media bit-rate profile; default is **pf**.

Virtual Player Type youtube

The **type youtube** (formerly type 5) virtual player is designed for YouTube and offers a subset of Media Flow Controller delivery options: **assured-flow**, **connection max-bandwidth**, **fast-start**, **seek** (see [Virtual Player Type generic](#) for details), and YouTube-specific options for identifying the requested video, **video-id**, and its format, **format-tag**.

Virtual Player Type smoothstream-pub

Smooth Streaming is an HTTP-based adaptive streaming technology implemented by Microsoft. The media format defined by Microsoft for smooth streaming supports both storage and on-the-wire delivery, and is based on the ISO/IEC 14496-12 ISO Base Media File Format specification. Smooth Streaming technology requires content to be encoded at multiple bit-rates that is then delivered to clients (for example, Microsoft Silverlight Player) as a series of small chunks or fragments. This allows the client player to dynamically switch between fragments of different bit-rates/qualities, depending on network bandwidth and CPU state, allowing viewers to have the best possible experience they possibly can.

The **type smoothstream-pub** virtual player supports publishing Smooth Streaming format ISMV files to fragmented format for the Silverlight player similar to IIS Windows Server.

Using assured-flow

Assured Flow ensures that Media Flow Controller provides the required bandwidth for a connection so that media encoded at different bit-rates are delivered at approximately the encoded bit-rate rather than the fastest possible. This helps optimize use of the available bandwidth per session along with contributing to the viewing experience of the end user. Using this feature requires that Media Flow Controller manages all physical interface's bandwidth:

1. All namespaces must include a virtual-player configured with AFR.
2. No **resource-pool** configuration (**resource-pool** is a way to reserve bandwidth).
3. Best-case deployments:

- a. Other binary or tunnel-code path traffic use very little bandwidth (good example: 100% cache hit in reverse proxy deployment).
- b. There are no idle or slow client connections.

Using query-string-parm

The **query-string-parm** argument, used extensively in virtual player configurations, allows you to use query parameters (also known as query params). Query parameters, a string with an associated value, are a way of passing information through a URL. The query parameter part of the URL is designated with a question mark (?) followed by defined query parameters. The query parameter is a name that is associated with a pre-defined value. Additional queries in the URL are separated by ampersand signs (&). Query parameters are composed of a name and value pair. For example, a request for a query parameter for assured-flow-rate could be shown in a URL like this:

```
http://xyz.com/test.flv?afr=100
```

In the example, the **query-string-parm <string>** is **afr** and its value is **100**. The namespace for this connection tells Media Flow Controller that when it finds **afr** in the query params part of the URL it is to use the value following it for that function. So, if the URL has **?afr=100** Media Flow Controller knows (through the URL's defined namespace and associated virtual player or configured network connection properties) to use 100 Kbps for the assured-flow rate.

In the Media Flow Controller CLI, you can only specify the query parameter **<string>** and should know the units of the value for that query parameter as query parameters can be defined to mean many different things and are used to signal the start or value of **assured flow**, **fast-start**, **full-download**, **seek**, and the **match** value for **hash-verify** and **rate-map**.



CAUTION: The virtual player **query-string-parm** values you configure in your Media Flow Controller origin must match the corresponding **query-string-parm** values configured in your Media Flow Controller edge.

Using hash-verify

Media Flow Controller computes an **md-5** hash of an incoming URL by combining a part of the URL, specified by the **url-type** and including the **expiry-time-verify** value if used, along with a configured **shared-secret** that is **appended** or **prefixed** (as configured) to it. The computed hash digest value is then compared with the hash value provided in the incoming URL via a configured **match query-string-parm**. If a match between the computed and provided hash values is unsuccessful, the request is denied.

Example URL showing **expiry-time-verify query-string-parm e**, and **match query-string-parm h**:

```
http://www.example.com/media/foe.flv?e=3312665958&h=<128-bit-md-5-hash>
```

If Media Flow Controller encounters this URL, and **url-type** is set to **absolute-url**, it takes the entire URL up to the configured **match query-string-parm** (underlined in the example). If **url-type** is set to **relative-uri**, it takes the part of the URL after the access method and domain plus the query string up to the configured **match query-string-parm** (`/media/foe.flv?e=3312665958` in the example). If **url-type** is set to **object-name**, it takes the part of the URL after the last slash plus the query string up to the configured **match query-string-parm** (`foe.flv?e=3312665958` in the example).

The hash value is then computed by either appending or prefixing to the URL (or URL part if **url-type** is set to **relative-uri** or **object-name**) the configured **shared-secret**, and comparing the computed value with the hash value provided via the **match query-string-parm** (shown above in blue).

Example if **shared-secret** is appended and **url-type** is set to **absolute-uri**:

```
Computed hash value = MD5(http://video.example.com/public/2010/
  qwerty.flv?fs=5000&ri=300&rs=1234567 + shared-secret)
```

Example if **shared-secret** is prefixed and **url-type** is set to **absolute-uri**:

```
Computed hash value = MD5(shared-secret + http://video.example.com/public/
  2010/qwerty.flv?fs=5000&ri=300&rs=1234567)
```

Example **hash-verify expiry-time-verify** use case if **expiry-time-verify** is set to a non-zero value:

1. User goes to customer website to play a video.
2. The customer website generates a URL and provides it to the client player. This URL has an expiry time that is secured by hash authentication.
3. Client player uses this URL to fetch content from a video server.
4. Server receives the request, the URL has not expired, server delivers the video.
5. If anyone uses the same URL after some time and sends a request to the server, the server rejects it because the expiry time in the URL will not match the hash expiry time.

The **shared-secret** key is defined on Media Flow Controller and on the client player or browser that generates the video request. The content owner defines a secret key that the client player uses to generate hashed URL's. The content owner configures the same secret key, the **shared-secret**, on Media Flow Controller so that Media Flow Controller can validate those requests.



NOTE: Type **generic** virtual-players have a default **match query-string-parm** of **h**; this can be changed in the CLI.

Using virtual-player type qss-streamlet rate-map

Virtual player type qss-streamlet allows you to configure a **rate-map** to ensure a specified delivery rate is applied to certain requests. The configuration calls for you to specify a **match <string>** to a **rate <kbps>**; when the request arrives the **match <string>** is extracted from the URL and its corresponding **rate** (in kbps) is used for the delivery rate. The **match** string (length must be 2 bytes; for example, **01**) is extracted by going to the end of the URL and skipping 12 bytes from the end; the value in that location is mapped to the configured **rate** in kbps. Example:

```
http://video.example.com/public/BBB87026/xy_750_1938344/
  AC60E15B2A7C4A45AC4C1472E2AC0816_030000003F.flv
```

In the URL, the value **03** (12 bytes from the end of the URL) is extracted, and the corresponding assured flow rate (1000 Kbps) is applied.

The configured CLI looks like this:

```
virtual-player my_virtual_player type qss-streamlet
  rate-map match 01 rate 300
  rate-map match 02 rate 500
```



```
rate-map match 03 rate 1000
```

In this way, URLs containing **01** in the correct place (12 bytes from the end) map to an assured flow rate of **300**; with **02** the assured flow rate is **500**; and so on.

Defaults are:

```
Match: 00 Rate: 150 kbps
Match: 01 Rate: 180 kbps
Match: 02 Rate: 270 kbps
Match: 03 Rate: 330 kbps
Match: 04 Rate: 420 kbps
Match: 05 Rate: 470 kbps
Match: 06 Rate: 520 kbps
Match: 07 Rate: 575 kbps
Match: 08 Rate: 700 kbps
Match: 09 Rate: 800 kbps
Match: 0A Rate: 900 kbps
Match: 0B Rate: 1300 kbps
Match: 0C Rate: 1750 kbps
Match: 0D Rate: 1920 kbps
```



CAUTION: Media Flow Controller checks for an underscore (`_`) before byte 14; if the underscore is missing, the URI does not map correctly.

Related Topics

- [“Before You Configure Media Flow Controller” on page 82.](#)
- [“Configuring YouTube Video Caching \(CLI\)” on page 131](#)
- [“Configuring SmoothStream Video Caching \(CLI\)” on page 134](#)
- [“Configuring FlashStream Video Caching \(CLI\)” on page 135](#)

Example: Configuring generic Virtual Players (CLI)

The generic virtual player can be used to cache most Web content; however, if you want Juniper Networks' adaptive bit-rate functionality, you must configure a SmoothStream, or FlashStream virtual-player. For more information, [“Configuring SmoothStream Video Caching \(CLI\)” on page 134](#), or [“Configuring FlashStream Video Caching \(CLI\)” on page 135](#).

See **virtual-player** for CLI details. See [“Media Flow Controller Virtual Players” on page 51](#) for background information. Before you configure Media Flow Controller **virtual-player** options, see [“Before You Configure Media Flow Controller” on page 82](#).

To configure the **type generic** (formerly Type 0) virtual player:

1. Configure a virtual player with a **name** and **type generic** (puts you in virtual-player configuration mode). Use **no virtual-player <name>** to delete; use **virtual-player <name> no <option>** to make changes to configurations (either reset default or remove setting).
`virtual-player <name> type generic`

2. Configure hash verification options. In Release 2.0.7, only **md-5** digest is supported. Configure a **shared secret** value to be appended or prefixed to the URL as specified, for matching against the hash value provided in the URL and indicated by the **match query-string-param** you configure. Optionally, configure **expiry-time-verify** and **url-type** to help prevent bandwidth stealing (see ["Using hash-verify" on page 127](#) for details).


```
hash-verify digest md-5 shared-secret <string> {append | prefix} match
  query-string-param <string> expiry-time-verify query-string-param
  <string> url-type <type>
```
3. Configure download parameters for delivering files at the fastest possible speed. If you set **always**, file downloads are always delivered at the fastest possible speed; otherwise, you must have either a query param or a header name that initiate a full download.


```
full-download {always | match <string> {query-string-param <string> |
  header <header_name>}}
```
4. Optionally, configure **assured-flow** delivery optimization. The **auto** option is not supported in Release 2.0.7 (and not shown in the example). A query param can be used or define a static **rate** value in kbps, a value of **0** (zero) means no throughput at all. After a value is entered, this parameter is enabled. See ["Using assured-flow" on page 126](#) for configuration information and ["Media Flow Controller AssuredFlow" on page 47](#) for background.


```
assured-flow {query-string-param <string> | rate <kbps>}
```
5. Optionally, configure **connection max-bandwidth** delivery optimization. Default is **0** (unbounded) with the Media Flow Controller license, **200** kbps without it; you must have the license to change the unlicensed default. Use **no connection** to reset default.


```
connection max-bandwidth {0 | <kbps>}
```
6. Optionally, configure **fast-start** delivery optimization.


```
fast-start {query-string-param <string> | size <KB>}
```
7. Optionally, configure **seek** delivery optimization specifying a query param for when seek should start and how long it should last.


```
seek-config query-string-param <string> [seek-flv-type {byte-offset | time-
  secs | time-msec}] [seek-length query-string-param <string>] [seek-mp4-
  type {time-secs | time-msec}] [tunnel-mode {disable | enable}]
```
8. Type **exit** to leave virtual-player configuration mode. Example:


```
MFC (config) # virtual-player test type generic
MFC (config virtual-player test) # hash-verify digest md-5 match query-
  string-param h shared-secret zpzp prefix
MFC (config virtual-player test) # assured-flow query-string-param afr
MFC (config virtual-player test) # full-download always
MFC (config virtual-player test) # connection max-bandwidth 0
MFC (config virtual-player test) # fast-start time 90
MFC (config virtual-player test) # seek-config query-string-param sk
MFC (config virtual-player test) # smooth-flow query-string-param sf
MFC (config virtual-player test) # exit
MFC (config) #
```
9. Verify configurations with **show virtual-player <name>**.

Related Topics

- ["Before You Configure Media Flow Controller" on page 82.](#)
- ["Configuring YouTube Video Caching \(CLI\)" on page 131](#)

- [“Configuring SmoothStream Video Caching \(CLI\)” on page 134](#)
- [“Configuring FlashStream Video Caching \(CLI\)” on page 135](#)

Configuring YouTube Video Caching (CLI)

Configure a **youtube** type virtual player if you want to cache and optimize the delivery of YouTube videos.

- [Using Virtual Player Type YouTube](#)
- [Tunnel YouTube Seeks](#)
- [Configure YouTube Video Caching.](#)

Using Virtual Player Type YouTube

YouTube encodes media content using industry standard video and audio compression schemes such as H.264/AVC for video and AAC for audio. It stores the encoded bit streams using either FLV, MP4, or 3GP containers, depending on the spatial resolution of the video. Currently, YouTube supports the following different formats as outlined in [Table 11](#).

Associations to one of these formats is signaled through a request originating from the player via a query parameter typically of the type **fmt** or **itag**.

Table 11 YouTube Formats

| Media Types | Standard | Medium | High | 720p | 1080P | Mobile |
|-------------------------------|--------------------|--------------------|-----------|-----------|-----------|---------------|
| Format/Tag Values (fmt, itag) | 34 | 18 | 35 | 22 | 37 | 17 |
| Container Types | FLV | MP4 | FLV | MP4 | MP4 | 3GPP |
| Video Codec | H.264/AVC | H.264/AVC | H.264/AVC | H.264/AVC | H.264/AVC | MPEG-4 Part 2 |
| Audio Codec | AAC | AAC | AAC | AAC | AAC | AAC |
| Spatial Resolution | 320x240 640x480 | 480x360 480x270 | 854x480 | 1280x720 | 1920x1080 | 176x144 |

Requests originating from a YouTube player for a video asset have been observed to typically come in the following two forms (underlining highlights important details):

(a)GET "http://www.youtube.com/get_video?video_id=fBE7y6Uba5M&t=vjVQa1PpcFPfHDFKYQ1s_RIHTM-GxADM8vFGLxxc_rs=&el=detailpage&ps=&fmt=34&asv=2&noflv=1"

(b)GET "http://v8.nonxt7.c.YouTube.com/videoplayback?ip=0.0.0.0&sparams=id%2Cexpire%2Cip%2Cipbits%2Citag%2Calgorithm%2Cburst%2Cfactor&fexp=904405&algorithm=throttle-factor&itag=34&ipbits=0&burst=40&sver=3&expire=1266310800&key=yt1&signature=66222E9350B9BB5AC68297F12AC1DCB4C53AAFDE.55B33FFFA04EBF001AF39A4F316E657FC318E0E5&factor=1.25&id=efa3a0434887fdc0&redirect_counter=1"

It is observed that these two request formats do not have an explicit association or reference to the media object, and the URI themselves are not cache friendly. The association to the media object is provided using a combination of an **id** and **format** tag.

For case (a) this association is provided by the **video_id** and **fmt** query parameters.

For case (b) this association is provided by the **id** and **itag** query parameters.

Media Flow Controller uses a combination of these query parameters to generate an internal cache name for the media object. YouTube videos in Media Flow Controller are cached with a cache name format as:

```
yt_video_id_efa3a0434887fdc0_fmt_34
```

Media Flow Controller also supports random access via seek/scrub for YouTube videos. YouTube signals a seek point via a query parameter, **begin**, with units of milliseconds. Media Flow Controller translates this seek point to the correct position in the video file for both the FLV or MP4 container formats and data that is delivered to the player is from the seek point onwards to the end of the file.

Tunnel YouTube Seeks

To tunnel YouTube seeks, use this **virtual-player** configuration:

```
virtual-player youtube_player type youtube
  cache-name video-id query-string-param "id" format-tag query-string-
  param "itag"
  exit
```

Configure YouTube Video Caching

To configure YouTube video caching:

1. Configure a virtual player of **type youtube**:
 - a. The **cache name** and **seek** configuration are required.
 - b. For **seek** configuration, the URI query is **begin** (seek length is not required).
 - c. The cache name configuration is **video_id** and **fmt** (case **a**) OR **id** and **itag** (case **b**); see [“Using Virtual Player Type YouTube” on page 131](#) for descriptions.

```
virtual-player youtube_player type youtube
  cache-name video-id query-string-param "video_id" format-tag query-
  string-param "fmt"
  exit
```

OR

```
virtual-player youtube_player type youtube
  cache-name video-id query-string-param "id" format-tag query-string-
  param "itag"
  exit
```

2. Create a **namespace**, and set **match**, **origin-server**, and **domain** name values to filter YouTube video requests. The **<client_traffic_NIC>** for the **origin-server** must first be set with the **delivery protocol** command if this is for a transparent proxy deployment. See [“Transparent Proxy Example Configuration—YouTube” on page 71](#) for details.

```
namespace <name>
  match uri /videoplayback OR match uri /get_video
  origin-server http follow header host use-client-ip bind-to <client_traffic_NIC>
  domain regex "^.*\c\.youtube\.com|^.*\.googlevideo\.com"
```

3. Override the default cache age allowed for YouTube assets using **namespace origin-fetch** options to enable longer cache intervals; from **namespace** prefix mode set:

```

delivery protocol http origin-fetch content-store media cache-age-
threshold 300
delivery protocol http origin-fetch cache-directive no-cache override
delivery protocol http origin-request host-header inherit incoming-req
  permit
delivery protocol http origin-request x-forwarded-for disable

```

4. Allow Media Flow Controller to stop downloading an object after the client stops viewing it. If an object is partially cached, then on a second subscriber request, the remainder object is downloaded via a byte-range request. If the origin doesn't support byte-range requests, it sends the whole object and Media Flow Controller discards the part that has already been stored. From **namespace** prefix mode set:

```

delivery protocol http origin-fetch cache-fill client-driven

```

5. Add the **virtual-player** you configured; activate the **namespace**, exit, and save your configuration. From **namespace** prefix mode set:

```

virtual-player youtube_player
status active
write memory

```

Example:

```

MFC (config) # virtual-player ytplayerA type youtube
MFC (config virtual-player ytplayerA) # cache-name video-id query-string-
  parm id format-tag query-string-parm itag
MFC (config virtual-player ytplayerA) # seek-config query-string-parm begin
MFC (config virtual-player ytplayerA) # exit
MFC (config) # delivery protocol http interface <client_traffic_NIC>
MFC (config) # delivery protocol http transparent <client_traffic_NIC> enable
MFC (config) # namespace youtube
MFC (config namespace youtube) # delivery protocol http origin-fetch
  cache-fill client-driven
MFC (config namespace youtube) # delivery protocol http origin-fetch
  content-store media cache-age-threshold 300
MFC (config namespace youtube) # domain regex
  "^.*\.c\.youtube\.com|^.*\.googlevideo\.com"
MFC (config namespace youtube) # match uri /videoplayback precedence 5
MFC (config namespace youtube) # origin-server http follow header host
  use-client-ip bind-to <client_traffic_NIC>
MFC (config namespace youtube) # virtual-player youtube_player
MFC (config namespace youtube) # status active
MFC (config namespace youtube) # exit
MFC (config) # write memory

```



NOTE: For transparent proxy deployments, see [“Transparent Proxy Example Configuration—YouTube” on page 71](#) and [“Example: Virtual Player Tuning” on page 77](#).

Configuring SmoothStream Video Caching (CLI)

- [About SmoothStreaming](#)
- [SmoothStreaming Multi-Bit-Rate Assets](#)
- [Example: SmoothStreaming Workflow](#)
- [Configure SmoothStreaming Caching and Delivery \(CLI\)](#).

About SmoothStreaming

Smooth Streaming is an HTTP-based adaptive streaming technology implemented by Microsoft. The media format defined by Microsoft for smooth streaming supports both storage and on-the-wire delivery, and is based on the ISO/IEC 14496-12 ISO Base Media File Format specification. Smooth Streaming technology requires content to be encoded at multiple bit-rates which is then delivered to clients (i.e., Microsoft Silverlight Player) as a series of small chunks or fragments. This allows the client player to dynamically switch between fragments of different bit-rates/qualities depending on network bandwidth, CPU state, and so forth, allowing viewers to have the best possible experience they possibly can.

SmoothStreaming Multi-Bit-Rate Assets

A typical multi-bit-rate media asset encoded according to the Smooth Streaming format comprises of following files:

- Media files containing video and/or audio (*.ismv or *.isma): Denoted by the **ismv** and **isma** extensions. The format supports storage of either a single bit-rate per file or can package all bit-rates into a single file.
- Server Manifest File (*.ism): Denoted by the ***ism** extension. They are XML files which describe to the streaming server the relationship between media tracks, their bit-rates, and files on disk, in the smooth streaming package.
- Client Manifest File (*.ismc): Denoted by the ***ismc** extension. It is typically the first file requested by a client and describes to the client the streams, codecs, encoded bit-rates, and video resolutions that are available in this package.

Example: SmoothStreaming Workflow

A typical workflow between a player/client and the server is as follows:

1. The Client requests from the server a client manifest (*.ismc) file. The manifest describes to the client which bit-rates and resolutions are available, and a list of all the available chunks and either their start times or durations.
2. Thereafter the client requests fragments in the form of specific RESTful URLs, for example:

```
http://test.media.com/bunny.ism/QualityLevels(400000)/
  Fragments(video=134345672)

http://test.media.com/bunny.ism/QualityLevels(64000)/
  Fragments(audio=123452356)
```

The values passed in the URL represent bit-rate (**400000**) and the fragment start offset (**134345672**).

Media Flow Controller with the **virtual-player type smoothstream-pub** configured, supports the delivery of Microsoft's Smooth Streaming media assets. This virtual player supports on-demand, file-based media only; it does not support live streaming. It is capable of understanding the RESTful URL's and can fragment in real time by looking at the specified quality level/fragment offset in the URL. Internally it reads the server manifest (*.ism) and locates the physical *.ismv or *.isma file. It can then parse the .ismv/.isma files as per the Smooth Streaming spec and finds the fragment box ('moof' + 'mdat') that corresponds to the requested start time offset. Media Flow Controller can extract this fragment box and send over the wire to the client as a standalone file with the correct content type (video/mp4, audio/mp4, text/xml, and so forth).

The fragmentation functionality can be deployed in multiple scenarios; for example, a single Media Flow Controller at the origin to perform the fragmentation and multiple Media Flow Controller edge caches caching and delivering the fragments. Or, the fragmentation functionality can be deployed at each edge location.

Configure SmoothStreaming Caching and Delivery (CLI)

To configure SmoothStreaming caching and delivery:

1. Create the SmoothStream virtual player:
`virtual player <name> type smoothstream-pub`
2. In prefix mode, define the **quality-tag** identifier to describe to Media Flow Controller the bit rate of the requested media segment. Default is **QualityLevels** (case sensitive).
`quality-tag <string>`
3. In prefix mode, define the **fragment-tag** identifier whose value describes to Media Flow Controller the timestamp of the requested media segment. Default is **Fragments** (case sensitive).
`fragment-tag <string>`

Example:

In the above CLI, the **quality-tag** identifier and **fragment-tag** identifier would be the parameters specified in the URL below in parenthesis:

```
http://test.media.com/bunny.ism/QualityLevels(400000)/  
Fragments(video=134345672)
```

Here the **quality-tag** is configured as **QualityLevels**, and the **fragment-tag** as **Fragments**.

Configuring FlashStream Video Caching (CLI)

- [About FlashStreaming](#)
- [Example: FlashStreaming Workflow](#)
- [Configure FlashStreaming Caching and Delivery \(CLI\)](#)

About FlashStreaming

HTTP adaptive streaming support to the Flash Platform commonly referred to as HDS (HTTP Dynamic Streaming) or Zeri was introduced by Adobe recently. The technology requires media assets to be packaged/pre-published into fragments which Flash player clients can access without the need to download an entire file.

To support HDS in Flash, several components in the publishing and delivery workflow need to be addressed. Media assets need to be packaged in the Zeri format. Delivery servers/caches need to understand the HDS fragment requests sent by a player. The servers/caches also need to perform in-line fragmentation to deliver the right fragment when configured as origin-servers/caches.



NOTE: To publish assets according to the Zeri format, Adobe provides a utility called the File Packager, which translates multiple bit-rate media assets into segments as F4F files along with the manifest file (F4M) and an index file (F4X). The File Packager is available from adobe.com and is installed with Adobe® Flash® Media Server in the `rootinstall/tools/f4packager` folder.

Delivery servers, particularly those that front-end origin stores/libraries hosting Zeri formatted media need to be capable of performing on-demand fragmentation of the assets into the respective Zeri fragments (F4F). In Media Flow Controller, a new **virtual-player type**, **flashstream-pub**, is capable of serving on-demand files created by the File Packager (Adobe). This virtual player supports on-demand, file-based media only. It does not support live streaming.

The **flashstream-pub** virtual player is capable of processing Flash Fragment File Format (F4F), Flash Manifest File Format (F4M), and Flash Index File Format (F4X) requests.

Example: FlashStreaming Workflow

The example presumes a sample player that supports HDS is the OSMF player, built using the Open Source Media Framework (OSMF) which runs in Flash Player 10.1 (osmf.org):

1. The player first requests for a manifest file (F4M file).
`http://<server-name>/<path-to-asset>/<manifest-file-name>.f4m`
2. The player receives the manifest file, parses the bootstrap information and then creates a request by mapping a video time-code to a Segment#-Fragment# combination.
3. The player sends this request to Media Flow Controller requesting for a specific fragment from an F4F file.
`http://<server-name>/<path-to-asset>/foo1000Seg1-Frag4`
4. Media Flow Controller receives this request and the virtual player `flashstream-pub` parses the request to infer the video asset name (`foo1000`), segment number (`1`), and fragment number (`4`).
5. Media Flow Controller fetches the corresponding index file (F4X: `foo1000Seg1.f4x`) for the asset segment (`foo1000Seg1`) and finds the offset to the fragment number `4` in the segment file (F4F: `foo1000Seg1.f4f`).
6. Media Flow Controller then serves only the required bits from the obtained byte offset which corresponds to the fragment number `4` for this asset to the player.
7. The Flash player receives this fragment and starts playback.
8. Once the asset is fragmented, the delivery/caching systems further down the pipeline do not need a virtual player as the data is now pre-chunked.

Configure FlashStreaming Caching and Delivery (CLI)

To configure FlashStreaming caching and delivery:

1. Create the FlashStream virtual player:

```
virtual player <name> type flashstream-pub
```
2. Define the **segment-tag** identifier that describes to Media Flow Controller the string to search while parsing a Zeri HDS request. The number immediately following this tag in the request denotes the segment number.

```
segment-tag <string>
```
3. Define the **fragment-tag** identifier that describes to Media Flow Controller the string to search while parsing a Zeri HDS request. The number immediately following this tag in the request denotes the fragment number.

```
fragment-tag <string>
```
4. Define the **seg-frag-delimiter** identifier that describes to Media Flow Controller the separator to search while parsing a Zeri HDS request which acts as a delimiter to the segment and fragment tags.

```
seg-frag-delimiter <string>
```

Example; a typical HDS request for a fragment is of the form:

```
http://<server-name>/<path-to-asset>/foo1000Seg1-Frag4
```

Here the **segment-tag** is configured as **Seg**, the **fragment-tag** as **Frag**, and the **seg-frag-delimiter** as - (dash).

Configuring NFS Fetch for Images (CLI)

Configuring Media Flow Controller to fetch all JPEG image files located in a particular directory on an NFS origin server requires another **namespace** configuration.

In this example, all requests for files at **www.example.com/jpgImages/** that incur a cache miss (necessitating an origin fetch) are fetched from the specified origin using NFS; the configuration could look like this:

```
MFC (config) # namespace exampleJpegs
MFC (config namespace exampleJpegs) # domain www.example.com
MFC (config namespace exampleJpegs) # match uri /jpgImages/
MFC (config namespace exampleJpegs) # origin-server nfs exampleOS.com: /
    home/jpgImages
MFC (config namespace exampleJpegs) # status active
MFC (config namespace exampleJpegs) # exit
MFC (config) #
```

The **match uri-prefix** of **/jpgImages** tells Media Flow Controller to use the defined namespace rules for incoming requests containing **/jpgImages** in the URL: NFS to fetch the requested file from the specified origin-server on the default port (default NFS port is 2049).

Just as with the initial namespace configuration, repeat the namespace configurations with the **domain** specified without the “www” prefix, and again specified with the IP address, if requests may come in those ways. Test the configuration with **ping**.

Tip! If unsure what port you are using for **origin-server**, use standard Linux shell commands to figure out the port, and then configure it along with the origin-server, if not the default.

Configuring HTTP Fetch for Videos (CLI)

Configuring Media Flow Controller to fetch all video files, located in a particular directory on the origin server, via HTTP, requires another namespace configuration. In this example, all requests for videos at **www.example.com/videos/** that incur a cache miss (necessitating an origin fetch) are fetched from the specified origin using HTTP.

For our example company, **www.example.com**, the configuration could look like this:

```
MFC (config) # namespace exampleVideos
MFC (config namespace exampleVideos) # domain www.example.com
MFC (config namespace exampleVideos) # match uri /videos/
MFC (config namespace exampleVideos) # origin-server http
    www.exampleOS.com 80
MFC (config namespace exampleVideos) # status active
MFC (config namespace exampleVideos) # exit
MFC (config) #
```

The **match uri-prefix** of **/videos** tells Media Flow Controller to use these namespace rules for incoming requests containing **/videos** in the URL: use HTTP to fetch the requested file from the specified origin-server on port 80 (HTTP default). Just as with the initial namespace configuration, repeat the namespace configurations with the **domain** specified without the “www” prefix, and again specified with the IP address, if requests may come in those ways.

Tip! If unsure what port you are using for **origin-server**, use standard Linux shell commands to figure out the port, and then configure it along with the origin-server, if not the default.

Configuring RTSP Fetch for Videos (CLI)

Configuring Media Flow Controller to fetch all video files, located in a particular directory on the origin server, via RTSP, requires another namespace configuration. In this example, all requests for videos at **www.example.com/videos/** that incur a cache miss (necessitating an origin fetch) are fetched from the specified origin using RTSP.

For our example company, **www.example.com**, the configuration could look like this:

```
MFC (config) # namespace exampleVideos
MFC (config namespace exampleVideos) # domain www.example.com
MFC (config namespace exampleVideos) # match uri /videos/
MFC (config namespace exampleVideos) # origin-server rtsp
    www.exampleOS.com 554
MFC (config namespace exampleVideos) # status active
MFC (config namespace exampleVideos) # exit
MFC (config) #
```

The **match uri-prefix** of **/videos** tells Media Flow Controller to use these namespace rules for incoming requests containing **/videos** in the URL: use RTSP to fetch the requested file from the specified origin-server on port 554 (RTSP default).

Just as with the initial namespace configuration, repeat the namespace configurations with the **domain** specified without the “www” prefix, and again specified with the IP address, if requests may come in those ways.

Tip! If unsure what port you are using for **origin-server**, use standard Linux shell commands to figure out the port, and then configure it along with the origin-server, if not the default.

Pre-Staging Content with FTP (CLI)

You pre-stage content to a Media Flow Controller edge cache or origin so it is there when the first request arrives. Typically, content is pre-staged to origin; however, you can pre-stage content to the edge, but always pre-stage such content to origin as well because edge caches function as volatile storage and evict content rapidly.

When you create a namespace, an FTP user is automatically created with the name **<namespace>_ftpuser**, *without a password*. You need to use the CLI **username** or **namespace** commands to give the user a password; then you can log in as that namespace's FTP user and issue the FTP commands to push the content to Media Flow Controller (default FTP port is 21). In this way, each FTP session transfers content only for one namespace at a time and users cannot view the content of another FTP user.

If the pre-staging is to an attached library (2 machine arrangement), then you can see the listing of the files using your FTP client. If the pre-staging is to Media Flow Controller directly, then you can use the **namespace object list all** command to see the listing of transferred files.



NOTE: FTP in Media Flow Controller has no checksum support.

CHAPTER 6

Configuring Namespaces (CLI)

- [Creating a Namespace and Setting Namespace Options \(CLI\)](#)
- [Using namespace cache-inherit](#)
- [Using namespace Cookie-Based Authentication and Filtering](#)
- [Using namespace delivery protocol {http | rtsp} origin-fetch cache-age](#)
- [Using namespace domain regex](#)
- [Using namespace domain <FQDN:Port>](#)
- [Using namespace match <criteria> precedence](#)
- [Using namespace match uri regex](#)
- [Using namespace match virtual-host](#)
- [Using namespace object delete | list](#)
- [Using Namespace Forced Tunneled-Transaction Override](#)
- [Using namespace for Pre-Staging Content via FTP](#)
- [Using namespace for Live Streaming Delivery Without Caching](#)
- [Using namespace for Live Streaming Delivery With Caching](#)
- [Using namespace for Proxy Configurations](#)
- [Using namespace for Cluster Configurations](#)
- [Example: Configuring Media Flow Controller Namespaces \(CLI\)](#)
- [Using Namespace for Dynamic URI Remapping](#)
- [Dynamic URI Websites' Namespace Configuration Examples](#)

Creating a Namespace and Setting Namespace Options (CLI)

Create namespaces to define fine-grained delivery policies, including optionally adding a custom virtual player. You must create a namespace for each origin server and delivery criteria scheme you use. You can create up to 256 namespaces in one Media Flow Controller. See [namespace](#) for CLI details. Before you configure Media Flow Controller namespaces, see [“Before You Configure Media Flow Controller” on page 82](#).

If your namespace will use a server map, see [Chapter 8, “Configuring Media Flow Controller Server Maps”](#) for information about creating server maps.

Related Topics

- [Chapter 3, “Media Flow Controller Deployment Guidelines.”](#) for reverse proxy and transparent proxy configuration examples.
- [Chapter 8, “Configuring Media Flow Controller Server Maps.”](#) for server map details.
- [Chapter 10, “Troubleshooting Media Flow Controller.”](#) for additional namespace tips.

Using namespace cache-inherit

Use the namespace **cache-inherit** option to add an existing namespace's cache and UUID to a new one; the contents are not duplicated, but the new namespace uses the inherited cache rather than creating a new one. When a namespace is created, the system assigns it a Unique ID (UUID). There is no option in the CLI to configure the UUID; but it can be set indirectly using the **cache-inherit** subcommand that sets a new namespace to inherit the cache of an existing namespace. This is useful under the following situations:

- You add a new namespace and want it to share the UUID with an existing namespace; sharing the UUID allows the two namespaces to have a common cache.
- You delete an existing namespace, rename it, and want to use the data cached under it.
- You delete a namespace by accident and want to re-create it and you do not want it given a new UUID. In this case, you dump the namespace and its associated UUID, and force the UUID of an existing namespace for the one you are creating. You would do this by performing the following steps:
 1. Issue **show namespace list** to gather the list of "Currently defined" and "Deleted/non-existing" (but whose cache content still exists) namespaces and their associated UUIDs.
 2. Issue **namespace <name> cache-inherit <existing or non-existing namespace integer>**.

Example: **show namespace list** (namespace **test2** inherited namespace **test** cache):

```
show namespace list
Currently defined namespaces :
    Name : UUID
        new_ns : /new_ns:b3ad64a8 (inactive)
        test2 : /test:1250bcac (active)
        testIE : /testIE:b911b24 (inactive)
-----
List of unmapped/deleted namespace UUIDs (if any)
    non-exsiting1: /test3:954ef8aa
```

Using namespace Cookie-Based Authentication and Filtering

Media Flow Controller has configurations that allow you to cache or not cache request URIs with cookies, and to specify responses to have a cookie set that causes the response to be cached.

You define when an incoming object with a cookie can be cached using the **namespace <name> delivery protocol http client-request cookie-action** options (**cache** and **no-cache**). Setting this option to **no-cache** means “tunnel the response.”

You can configure Media Flow Controller to set a cookie on certain headers to cache or not cache objects when media is delivered from Media Flow Controller using the **namespace <name> delivery protocol http origin-fetch header set-cookie** option.

Other unknown strings including “Cache-Control: public,” “Cache-Control: no-cache=set-cookie” and so forth, Media Flow Controller does not honor but just ignores them, forwarding them as-is to the client.

Using namespace delivery protocol {http | rtsp} origin-fetch cache-age

The **namespace <name> delivery protocol http origin-fetch cache-age** argument allows you to set granular cache aging policies based on content type, as well as a default cache age. The **cache-age** for each **content-type** must be specified separately with positive integers. Examples:

- cache-age content-type-any 28800**
Irrespective of content type, override your configured **cache-age-default** or, if **cache-age-default** is unconfigured, set the Max-Age for any content type to **28800** seconds. If the content request does not specify a Max-Age, set it to Max-Age **28800**.
- cache-age content-type application/flv 2880**
cache-age-default 900
When **content-type** is **application/flv**, set Max-Age to **2880** seconds. For all other content-types use **default** configuration (**900** seconds in the example). If the received Max-Age is set, use that value.
- cache-age content-type application/flv 28800**
cache-age content-type application/mov 2880
cache-age content-type application/3gp 288
cache-age content-type application/f4v 28
cache-age-default 900
When **content-type** is **application/flv**, set Max-Age to **28800** seconds; for **application/mov**, set Max-Age to **2880** seconds; for **application/3gp**, set Max-Age to **288**; and for **application/f4v**, set Max-Age to **28**. For all other content-types, use **default** configuration (**900** seconds in the example). If the received Max-Age is set, use that value.
- cache-age content-type application/qmx 60**
cache-age content-type application/qss 288000
When **content-type** is **application/qmx**, set **cache-age** to **60** seconds; for **application/qss**, set **cache-age** to **288000** seconds. For all other content-types, use **default** configuration: if Max-Age is not set in the data coming from origin, set it to the configured **default** value (**28800** if unspecified). If the received Max-Age is set, use that value.

Using namespace delivery protocol http origin-fetch cache-fill

When the origin server is known to not support byte-range requests, do not configure the namespace **origin-fetch cache-fill** policy to be **client-driven**. Instead, use the **cache-fill aggressive** option.

If Media Flow Controller is configured with a namespace **origin-fetch cache-fill client-driven** policy, and a partial file is present in the cache, if the client makes GET request for the same object, Media Flow Controller delivers to the client the partial file available from cache and makes a byte-range request to the origin server for the remaining bytes. If the origin server does not support byte-range requests, it responds with “200 OK” for the byte-range request, the partial file is not deleted from the cache, and the video delivery is stopped.

Using namespace domain regex

This section provides some examples of **namespace domain regex** use. Change specifics accordingly. See [Table 12](#).



NOTE: Regex entries do not contain spaces; also, enclose all **regex** entries in double quotation marks (as shown in examples).

Table 12 Example **namespace domain regex** Entries

| Regex | Example Matches |
|--|---|
| "www.example.com example.com" | www.example.com example.com |
| .*example.com | example.com |
| "^[a-f,0-9]{8}\.(origin\. cdn\.)?cms\.example\.com:80\$" | abcdef02.origin.cms.example.com:80 abcdef23.cdn.cms.example.com:80 abcdef09.cms.example.com:80 |
| "^cms[0-9]{3}\.(dc2 qcg7)\.example\.com:80\$" | cms123.dc2.example.com:80 cms079.qcg7.example.com:80 |
| "^orig.(sv1 qcg1 qcg5)\.example\.com:80\$" | orig.sv1.example.com:80 orig.qcg1.example.com:80 orig.qcg5.example.com:80 |
| "^(cms[0-9]{3}).*(qcg[0-9]+ sv1 ch1 dc2 af1)\.example\.com:80\$" | cms123.x.y.qcg0.example.com:80 cms257.x.y.qcg01.example.com:80 cms379.x.y.sv1.example.com:80 cms222.x.y.ch1.example.com:80 cms876.x.y.dc2.example.com:80 cms343.x.y.af1.example.com:80 |

Using namespace domain <FQDN:Port>

Media Flow Controller can listen for requests on multiple TCP ports, up to 64; default is port 80 for HTTP, and port 554 for RTSP. In order to map incoming requests to the correct namespace, especially on a non-default port, the **namespace domain** must be set properly with port number included. Media Flow Controller maps an incoming URL to a namespace by extracting the value against the HOST header and matching it to the value configured for the **namespace domain**.

When non-default port numbers are used you must ensure that the HOST header has the port number coded correctly in the incoming URL and also in the **namespace domain**.

Example: If Media Flow Controller listens on port 80, 8080, and 4040 for incoming HTTP requests; and the requests on port 80 must go to namespace **ns80**, those coming in on port

4040 must go to namespace **ns4040**, and those on port 5050 must go to namespace **ns5050**, then the configuration would be as follows:

```
namespace ns80
  domain video.example.com

namespace ns8080
  domain video.example.com:4040

namespace ns4040
  domain video.example.com:5050
```

For requests to match **ns80**, **domain/HOST**: header must be **video.example.com**.

For requests to match **ns4040**, **domain/HOST**: header must be **video.example.com:4040**.

For requests to match **ns5050**, **domain/HOST**: header must be **video.example.com:5050**.

Using namespace match <criteria> precedence

Use **namespace <name> match <criteria> precedence** to set unambiguous mapping of incoming GET requests in the case of **match <criteria>** overlap; **precedence** can be set on all **match <criteria>**. The lower the number, the higher the preference for that namespace; values **0** (highest precedence) through **10** (lowest precedence) can be used. All namespaces have a default precedence of **0**. For example, consider three URLs and namespaces as follows:

1. http://a.com/abc/def/file1.flv
2. http://a.com/abc/file2.flv
3. http://a.com/pqr/file3.flv

```
namespace ns1
  domain a.com
  match uri /abc/def precedence 1
  origin-server http o1.com
  status active

namespace ns2
  domain a.com
  match uri /abc precedence 2
  origin-server http o2.com
  status active

namespace ns3
  domain a.com
  match uri / precedence 3
  origin-server http o3.com
  status active
```

All three URLs match namespace **ns3** with **domain (a.com)** and **match uri /** (slash). In order to ensure that **match uri #1 (/abc/def)** maps to **ns1** and not **ns3**, set the **precedence** value. Likewise, with **match uri #2 (/abc)**, to map to **ns2**, a **precedence** value is needed. Only **match uri 3 (/)** should be mapped to **ns3** with the **precedence** value configured as shown.



NOTE: Excessive use of **precedence** has performance impact as **precedence** allows for longest prefix matching. If possible, namespaces should be configured in such a way that they have no overlaps in the **domain** and **match <criteria>** combination, which are used for mapping the incoming HTTP GET to a namespace.

Using namespace match uri regex

This section provides some examples of **namespace match uri regex** use.



NOTE: Regex entries do not contain spaces; also, enclose all **regex** entries in double quotation marks (as shown in examples). See [Table 13](#).

For **namespace match uri <uri-prefix>**, the regex is written against the absolute path portion of the URL. For example, given the following URL: **http://abc.com:8080/index.html**, **/index.html** would be the absolute path portion of the URI.

Table 13 Example **namespace match uri regex** Entries

| Regex | Example Matches |
|------------------------|--|
| "\A\b\c\d\index\.html" | /A/b/c/d/index.html |
| "\A\b\c\d\index\..*" | /A/b/c/d/index.html /A/b/c/d/index.html |
| "\A\b\.*\d\index\..*" | /A/b/x/d/index.html /A/b/xx/d/index.html /A/b/abc/d/index.html |

Using namespace match virtual-host

You can specify a virtual host for the **namespace match** criteria; see [Virtual host](#) for definition. To do this, enter the IP address of the virtual host and, optionally, a port number; you can also set a precedence, if needed. In this configuration, the incoming session's destination IP is used to find the match to the namespace. In order for the match to be based only on destination IP you should set **domain** to **any**. In that way, the incoming session's destination IP and destination port (if given) are used to match to the namespace. Examples, from namespace prefix mode:

- Match all sessions whose destination IP is 10.1.1.1 to this namespace:
`match virtual-host 10.1.1.1`
- Match all incoming requests with destination IP 10.1.1.1 and port 8080 to this namespace:
`match virtual-host 10.1.1.1:8080`
- Match all incoming requests on destination port 8080 to this namespace:
`match virtual-host 0.0.0.0:8080`

Using namespace object delete | list

List or delete contents in a namespace. The command takes in the name of a **namespace** and applies a **list** or **delete** operation to the objects matching the given pattern.

```
namespace <name> object {list | delete} {all | <URI> | pattern}
```

For example, with this namespace and a URL of **http://example.com/abc/def/file.flv**:

```
namespace ns1
domain example.com
match uri /abc
```

- To list an object and get its characteristics:
`namespace ns1 object list /abc/def/file.flv`

- To delete an object with the same URL:
`namespace ns1 object delete /abc/def/file.flv`
- To delete all the objects in that namespace's disk cache with the same URL:
`namespace ns1 object delete all`
- To list the first 50 objects in that disk cache and create a file named with the UUID of the namespace listing all cached objects for that namespace. In the example, if the namespace had a UUID of **80213A2C**, the file containing the list is **80213A2C.lst**. Only the first 50 cached objects are listed; if there are more than 50, use the **upload** command. See [“Terminology” on page 31](#) for the **scp** URL format).
`namespace ns1 object list all`
`upload object list <namespace> <SCP>`
- To list and delete objects based on patterns. For example; you can specify ***.flv** as a pattern. Media Flow Controller does not support a full regular expression for deleting or listing. The **namespace ns1 object list all** command is equivalent to **namespace ns1 object list /abc/def/***.

Using Namespace Forced Tunneled-Transaction Override

Media Flow Controller provides a mechanism for caching content as well as a mechanism for controlling how and when the cache can be bypassed. This bypass mechanism is called “tunneling.” The main benefit of cache tunneling is that by overriding certain tunneled transactions, the cache hit ratio is improved for the remaining transactions.

By default, Media Flow Controller will not cache objects under the following conditions:

- The request is sent with an auth header, cache-control header, or a cookie.
- The response contains a cache-control.

The **tunnel-override** command allows Media Flow Controller to override a few tunneled transactions using reason codes.

All **tunnel-override** options are **disabled** by default.

The CLI commands are:

```
namespace <name> delivery protocol http client-request tunnel-override
  auth-header
  cache-control
  cookie
```

```
namespace <name> delivery protocol http origin-fetch tunnel-override
  cache-control-no-transform
  object-expired
```

Notes:

- **client-request tunnel-override**—Configure cache-index parameters.
 - **auth-header**—Cache requests containing the “Auth” header.
 - **cache-control**—Cache requests containing either “Cache-Control: No-Cache,” or “PRAGMA: No-Cache” headers.
 - **cookie**—Cache requests containing the “Cookie” header.
- **origin-fetch tunnel-override**—Configure the tunnel reason code for responses that Media Flow Controller should cache.

- **cache-control no-transform**—Cache responses with the “Cache Control: No-Transform” header.
- **object-expired**—Cache responses that indicate an object has expired, and set a new expiry date as the current date plus the configured **cache_age**. This is discussed in more detail in the *Media Flow Controller Administration Guide*, in the section “Using namespace delivery protocol <protocol> origin-fetch cache-age.”

Using namespace for Pre-Staging Content via FTP

When creating a namespace for pre-staging the content using FTP, be sure to configure the **namespace <> delivery protocol http origin-fetch cache-age-default <seconds>** to a large, non-zero value; for example **28800**.

Using namespace for Live Streaming Delivery Without Caching

An example **namespace** configuration to deliver live streaming objects without caching is given; issuing the **delivery protocol** or **live-pub-point** command enters you to prefix mode.

```
namespace <name>
  match uri <uri-prefix>
  origin-server rtsp <IP_address | hostname> [port]
  status active
  delivery protocol rtsp
  exit
  live-pub-point <pp_name>
  receive-mode on-demand
  status active
  exit
  exit
```

Using namespace for Live Streaming Delivery With Caching

An example **namespace** configuration to deliver live streaming objects with caching is given; issuing the **delivery protocol** or **live-pub-point** command enters you to prefix mode.

```
namespace <name>
  match uri <uri-prefix>
  origin-server rtsp <IP_address | hostname> [port]
  status active
  delivery protocol rtsp
  exit
  live-pub-point <pp_name>
  receive-mode on-demand
  status active
  caching enable
  exit
  exit
```

Using namespace for Proxy Configurations

You can use **namespace** settings to configure Media Flow Controller to operate as a proxy in various ways.

- **Reverse proxy**—Caches and delivers content for a set of domains; client requests are routed to a configured IP address. Setting **namespace origin-server** to **<FQDN>** or

server-map implies a reverse proxy configuration. Media Flow Controller as an edge cache is effectively a **reverse** proxy that reduces network and CPU load on an origin server by serving previously retrieved content, and enhances user experience by decreasing latency. See also [“Reverse Proxy Deployments” on page 53](#).

- **Mid-tier proxy**—Caches and delivers content for a set of domains; client requests are routed to a configured IP address. Setting **namespace origin-server** to **absolute-url** implies a mid-tier proxy configuration. As a **mid-tier** proxy, Media Flow Controller must be explicitly configured in the browser to intercept all requests. After Media Flow Controller receives traffic from the client, it separates the traffic; cacheable requests are sent via Media Flow Controller for performance-enhanced delivery. Non-cacheable requests are tunneled. See also [“Mid-Tier Proxy Deployments” on page 79](#).
- **Transparent proxy**—Caches popular content and optimizes the backhaul network utilization; the cache is made to look transparent by spoofing the origin server IP address in the response to the client and spoofing the client IP address in the request to the origin. A transparent proxy is one that requires no browser configuration and is not readily visible to end users. As a **transparent proxy** where origin-server access is derived from the HOST header, the X-NKN header, or the destination IP address given in the incoming request, explicit **origin-server** configuration is disallowed. Use this as an alternative to providing a single origin server address. Be sure that **delivery protocol http allow-req** is set to **all** (default). See also [“Transparent Proxy Deployments” on page 66](#).

Table 14 gives details about the configurations and defaults per proxy deployment. See also [Chapter 3, “Media Flow Controller Deployment Guidelines.”](#) for details about configuring proxies.

Table 14 Namespace **origin-server** and **origin-request** Dependencies per Proxy Mode

| Proxy Mode | origin-server setting | For origin use the... | Source IP for Cache Miss | Destination IP for Cache Miss |
|--|--|-----------------------------|----------------------------------|---|
| | default for origin-request host header inherit incoming-req | | | |
| Reverse | http <FQDN> | Specified FQDN | Media Flow Controller IP address | DNS resolved IP address of FQDN |
| | deny | | | |
| Reverse | http server-map | Server-map HTTP mount point | Media Flow Controller IP address | DNS resolved IP address of server-map origin-server |
| | deny | | | |
| Reverse | nfs <FQDN:path> | NFS mount point | None (NFS mounted) | None (NFS mounted) |
| | permit | | | |
| Reverse | nfs server-map | Server-map NFS mount point | None (NFS mounted) | None (NFS mounted) |
| | permit | | | |
| Mid-Tier | http absolute-url | Client request absolute URL | Media Flow Controller IP address | DNS resolved IP address of origin-server chosen from the client request |
| | permit | | | |
| Transparent (origin based on HOST header) | http follow header HOST | HOST header value | Media Flow Controller IP address | DNS resolved IP address of the origin-server from the HOST header |
| | permit | | | |

Table 14 Namespace **origin-server** and **origin-request** Dependencies per Proxy Mode (Continued)

| Proxy Mode | origin-server setting | For origin use the... | Source IP for Cache Miss | Destination IP for Cache Miss |
|---|--|-------------------------------|----------------------------------|--|
| | default for origin-request host header inherit incoming-req | | | |
| Transparent (origin based on X-NKN or custom header) | <code>http follow header <name> use-client-ip</code> | Specified header | Client IP address | DNS resolved IP address of the origin-server from the given header |
| | <code>permit</code> | | | |
| Transparent (origin based on client destination IP) | <code>http follow dest-ip</code> | Client request destination IP | Media Flow Controller IP address | No DNS resolution. Origin IP is client destination IP |
| | <code>permit</code> | | | |
| Transparent | <code>http follow dest-ip use-client-ip</code> | Client request destination IP | Client IP address | No DNS resolution. Origin IP is client destination IP |
| | <code>permit</code> | | | |



CAUTION: If a non-default **origin-request host header inherit incoming-req** value is configured against an **origin-server** setting, the behavior is undefined and no error or warning is issued. For example, if **origin-server http absolute-url** is set (Mid-Tier proxy), and you set **origin-request host-header inherit incoming-req deny**, the behavior is undefined.

Using namespace for Cluster Configurations

You configure clustering via namespace origin server-facing origin clusters.

Configure origin clusters by adding up to three cluster-type (**cluster-map** or **origin-escalation-map**) server maps to a **namespace** via the **origin-server server-map** option.

To create origin cluster namespaces:

1. Create cluster-type maps (**cluster-map** and **origin-escalation-map**). For more information, see [“Creating the cluster-map XML File” on page 163](#) and [“Creating the origin-escalation-map XML File” on page 165](#).
2. Assign up to three cluster-type maps to a namespace:

```
namespace origin-server http <server-map_1>
namespace origin-server http <server-map_2>
namespace origin-server http <server-map_3>
```

Example: Configuring Media Flow Controller Namespaces (CLI)

Configure namespaces to set fine-grained delivery policies; every Media Flow Controller deployment must have at least one namespace, and usually several. To configure an Origin Cluster namespace, see [“Using namespace for Cluster Configurations” on page 150](#). See [namespace](#) for CLI details.

To configure a namespace:

1. Configure a namespace with a **name** (puts you in namespace configuration mode; use **exit** when finished); optionally inherit another namespace's cache or UUID. Use **show namespace list** to find namespace UUIDs.

```
namespace <name> [cache-inherit <namespace:UUID>]
```

2. Configure **domain** settings (default is **any**). The domain you enter should match whatever you have configured as HOST header, unless using **regex**; you may append a port number as well if needed (and used in HOST header). See ["Using namespace domain regex" on page 144](#) and ["Using namespace domain <FQDN:Port>" on page 144](#) for details.

```
domain {any | regex <regex> | <FQDN>}
```

3. Configure **origin-server** settings (example uses **http**); multiple origin servers can be configured with the **server-map** option; **port** specification is optional. See [Chapter 8, "Configuring Media Flow Controller Server Maps,"](#) for more information. See [\(namespace\) origin-server](#) for CLI details.

```
origin-server
```

```
http {absolute-url | follow {header <header> [use-client-ip] | dest-ip
    [use-client-ip]} | server-map <map_name> | <FQDN/path> [<port>]}
```

```
nfs {<FQDN:export_path> [<port>] | server-map <name>}
```

```
rtsp {<FQDN> [<port#>]} [alternate <string> [<port#>]]
```

Tip! If unsure what port your **origin-server** is using, use standard Linux shell commands (for example, **netstat -nl**) to figure out the port, and then configure it along with the **origin-server**, [if not the default](#). If you need to change the **origin-server**, or any **namespace** setting, simply enter the new setting.

4. Configure **match** criteria options (determines the URI to cache). All **match** options may utilize the **precedence** argument to break ties when namespaces are defined with the same **match** criteria. See ["Using namespace match <criteria> precedence" on page 145](#), for details. See ["uri-prefix" on page 34](#) for **uri-prefix** definition and usage details.

```
match
```

```
header {<header> | regex <regex>} [precedence <number>]
```

```
query-string {<name> | regex <regex>} [precedence <number>]
```

```
uri <uri-prefix> | regex <regex>} [precedence <number>]
```

```
virtual-host <IP_address> [<port>] [precedence <number>]
```

5. Configure **delivery protocol** options. Default **delivery protocol** is **http**. To enable **delivery protocol rtsp**, press Enter after **rtsp**; then set RTSP options.
 - a. Configure **delivery protocol http client-request** options; manipulate incoming client requests. See [\(namespace\) delivery protocol http client-request](#) for CLI details.

```
client-request
    cache-control max-age <seconds> action serve-from-origin
    cache-hit action revalidate-always
    cookie action {cache | no-cache}
    query-string action {cache [exclude-query-string] | no-cache}
```
 - b. Configure **delivery protocol http client-response** options. Delete headers in, or add headers to, outgoing responses to client requests. Up to 16 **headers**, including the custom header X-Accel-Cache-Control, can be configured with an **action** value (either **add** or **delete**). If you enter only a header **<name>**, the only action allowed is **delete**; if you enter a header **<name>** and **<value>**, the only action allowed is **add**.

- `client-response header <name> [<value>] action {add | delete}`
- c. Configure **delivery protocol origin-fetch** options; most are for **http**. Only **cache-age-default** is available for **rtsp origin-fetch**. See [\(namespace\) delivery protocol http origin-fetch](#) for CLI details.


```
origin-fetch
  cache-age {content-type<string><secs> | content-type-any <secs>}
  cache-age-default <seconds>
  cache-fill {aggressive | client-driven}
  content-store media [cache-age-threshold<secs>] [object-size<bytes>]
```
 - d. Configure **delivery protocol http origin-request** parameters for data requested from origin. See [\(namespace\) delivery protocol http origin-request](#) for CLI details. Set **origin-request host-header inherit incoming-req** in accordance with the **origin-server** setting; see [Table 14, "Namespace origin-server and origin-request Dependencies per Proxy Mode"](#) for CLI details. Use **x-forwarded-for** to allow (with **enable**) or disallow (with **disable**) setting the X-Forwarded-For header to the client IP address; default is **enable**.


```
origin-request
  cache-revalidation
  connect
  header
  host header
  read
  x-forwarded-for
```

Example:

```
MFC (config) # namespace test
MFC (config namespace test) # domain any
MFC (config namespace test) # origin-server http example.com/video
MFC (config namespace test) # match uri / precedence 3
MFC (config namespace test) # delivery protocol http
MFC (config namespace test delivery protocol http) # client-request cookie
  action no-cache
MFC (config namespace test delivery protocol http) # client-response header
  Location action delete
MFC (config namespace test delivery protocol http) # origin-fetch cache-age-
  default 0
MFC (config namespace test delivery protocol http) # origin-request x-
  forwarded-for enable
MFC (config namespace test delivery protocol http) # exit
MFC (config namespace test) # exit
```

6. Optionally, make **live-pub-point** settings if needed for live streaming.
 - **caching**—Enable caching for this service (default is disabled).
 - **receive-mode**—Set a method for receiving live streaming:
 - **on-demand**—When a request is received.
 - **sdp-name <URL>**—Use a service delivery protocol (SDP) file to set the live publishing point. The URL can be **scp://...** or **http://...** only. After Media Flow Controller encounters this, it pulls in the file from the specified location, and saves it in the file system (not disk cache) so it is available for RTP/RTSP. Optionally, choose **immediate** to start as soon as the file is retrieved or enter a **start-time**

and, optionally, an **end-time**. See [“Terminology” on page 31](#) for the **scp** URL format).

- **status**—Make active or inactive the live-pub-point.
7. Set parameters for pre-staging content from origin; authentication schemes must be pre-configured to be used. See [namespace](#) for CLI details. The **ftp user** is auto-generated as **<namespace>_ftpuser**, *without a password* (login disallowed). Set the password here; this entry overrides a **user <namespace>_ftpuser** password setting. When configuring a namespace for FTP pre-staging, also set **namespace <> delivery protocol http origin-fetch cache-age-default <seconds>** to a large, non-zero value; for example **28800**. Verify with **show usernames**. Remove set password with **no pre-stage ftp user <user_name>**.

```
pre-stage ftp user <name> password {RADIUS | TACACS | <password>
    [encrypt]}
namespace <> delivery protocol http origin-fetch cache-age-default 28800
```

Example:

```
MFC (config namespace test) # domain any
MFC (config namespace test) # origin-server http example.com/video
MFC (config namespace test) # pre-stage ftp user test_ftpuser password 678
MFC (config namespace test) # delivery protocol http origin-fetch cache-
    age-default 28800
```

```
MFC (config namespace test) # show usernames
```

| USERNAME | FULL NAME | CAPABILITY | ACCOUNT STATUS |
|--------------|----------------------|------------|--------------------------------|
| admin | System Administrator | admin | No password required for login |
| cmcrendv | CMC Rendezvous User | cmcrendv | Local password login disabled |
| monitor | System Monitor | monitor | No password required for login |
| test_ftpuser | | ftpuser | Password set |

```
MFC (config namespace test) # no pre-stage ftp user test_ftpuser
```

```
MFC (config namespace test) # show usernames
```

| USERNAME | FULL NAME | CAPABILITY | ACCOUNT STATUS |
|--------------|----------------------|------------|--------------------------------|
| admin | System Administrator | admin | No password required for login |
| cmcrendv | CMC Rendezvous User | cmcrendv | Local password login disabled |
| monitor | System Monitor | monitor | No password required for login |
| test_ftpuser | | ftpuser | Local password login disabled |

8. Also optional, add an existing **virtual-player** to the new namespace.
- ```
virtual-player <name>
```
9. Activate the namespace. Verify configurations with **show namespace <name>**.
- ```
status active
```

10. Type **exit** to leave namespace configuration mode. Example:

```
MFC (config namespace test) # virtual-player test
MFC (config namespace test) # status active
MFC (config namespace test) # exit
```



NOTE: Configuration changes, including a **namespace** deletion, may not be updated for up to 30 seconds. This is due to a deferred update scheme that requires an HTTP request. An internal probe ensures that such a request occurs at least every 30 seconds.

Related Topics

- [“Before You Configure Media Flow Controller” on page 82.](#)
- [“Media Flow Controller Namespaces” on page 50.](#)
- [Chapter 3. “Media Flow Controller Deployment Guidelines.](#)
- [Configuring Namespace for FMS VOD \(CLI\)](#)

Using Namespace for Dynamic URI Remapping

The commands for dynamic URI mapping specify that an incoming client request having a URI matching the configured regex value (**client-request cache-index url-match <regex>**) is matched to a cache index string (**map-to <map_string>**). The optional **no-match-tunnel** option specifies that if the match fails, the request is tunneled; use this option if you want HTTP to send the request on the normal path with no **cache-index** mapping, if the match fails. With **revalidate-always** configured, if the origin server returns “Object Is Modified,” the transaction is tunneled; the object is deleted from the cache, if its time-to-live (TTL) is expired; and the new object is fetched into the cache.

See [“Dynamic URI Remapping” on page 45](#) for background; see [“\(namespace\) delivery protocol http client-request” on page 413](#) for CLI details.



NOTE: You must configure **revalidate-always** in order for dynamic URI mapping to work.

Some **url-regex <regex>** examples:

1. Regex example with no substring address; only **\$0** returned:

```
/opt/nkn/bin/nknregex -m -e '/videoplayback\?.*\&id=[^\&]+.*' -d
'/videoplayback?xxx&id=1xxuuu'
match[0]: /videoplayback?xxx&id=1xxuuu
```

2. Regex example with one substring denoted (note parenthesis); **\$0** and **\$1** returned.

```
/opt/nkn/bin/nknregex -m -e '(/videoplayback\?.*)\&id=[^\&]+.*' -d
'/videoplayback?xxx&id=1xxuuu'
match[0]: /videoplayback?xxx&id=1xxuuu
match[1]: /videoplayback?xxx
```

3. Regex example with two substrings denoted; **\$0**, **\$1**, and **\$2** returned:

```
/opt/nkn/bin/nknregex -m -e '(/videoplayback\?.*)\&id=([^\&]+).*' -d
'/videoplayback?xxx&id=1xxuuu'
match[0]: /videoplayback?xxx&id=1xxuuu
match[1]: /videoplayback?xxx
match[2]: 1xxuuu
```

Some **map-to <map_string>** examples.

- Using regex example 3:
`/abc/$1/$2 => /abc//videoplayback?xxx/1xxuuu`
- Using regex example 3:
`/XXX$0/$1/$2 => /XXX/videoplayback?xxx&id=1xxuuu//videoplayback?xxx/1xxuuu`
- Using regex example 3:
`$$$1/$2 => $/videoplayback?xxx/1xxuuu`
- Using regex example 3:
`$$$1$$ => $/videoplayback?xxx$`

Configuring Dynamic URI Mapping

For example configurations for the subset of websites that Media Flow Controller supports for dynamic URIs, see [“Transparent Proxy Deployments” on page 66](#) and [Dynamic URI Websites’ Namespace Configuration Examples](#). For other URIs, Media Flow Controller treats the dynamic URIs as new requests, even though they are referencing an already cached object.

To configure dynamic URI mapping:

1. For your namespace **delivery protocol http client-request cache-index** parameter, configure a **url-match** regular expression (regex) to match the dynamic URL to a pattern. You can use the dynamic URI regex examples given in these Release Notes; or you can analyze your accesslogs to find the needed regex. Juniper Support can help; see [“Requesting Technical Support” on page 29](#) for details.
2. Configure the cache index, **map-to**, which indicates a part of the dynamic URL. Media Flow Controller uses this parameter to find an already-stored cache index, thereby mapping the dynamic URI to the same object in the cache.
3. If there is no match for the regex, and **no-match-tunnel** is configured, the object is tunneled. If **no-match-tunnel** is not configured, the object follows the caching path with the original URI.
4. Configure the **revalidate-always** option to ensure delivering the right content to the client: if the origin server returns “Object Is Modified,” the transaction is tunneled. If the request goes to the origin manager with dynamic URI configured, the original URI is always used.



NOTE: Dynamic URI mapping can be achieved with the **virtual-player** function for some websites. Currently, we recommend using the dynamic URI mapping **namespace** commands only for megaupload, hotfile, and rapidshare videos. Configuration examples are given in [Dynamic URI Websites’ Namespace Configuration Examples](#).

Media Flow Controller Enhancements for Dynamic URI Mapping

This section describes some Media Flow Controller enhancements for dynamic URI mapping.

- Accesslog enhancement.
`"%{X-NKN-Remapped-Uri}i"`: display internal cache index after dynamic URI remapping.

- Domain name exclusion from Media Flow Controller cache index.
To support GSLB deployment, same content could be delivered by different domains. For this kind of traffic, we exclude all or part of domain from the cache index. Thereby we could have a higher cache hit ratio. The CLI command to enable this feature is:

```
namespace <name> delivery protocol http cache-index domain-name exclude
```

Dynamic URI Websites' Namespace Configuration Examples

These example configurations for dynamic URI mapping do not use a **virtual-player**.



CAUTION: These examples are not guaranteed nor approved by the target content owners, but are believed to be correct at this time.

- [Example Configuration: Megaupload Namespace](#)
- [Example Configuration: Hotfile Namespace](#)
- [Example Configuration: Rapidshare Namespace](#)
- [Example Configuration: Filesonic Namespace](#)
- [Example Configuration: Putlocker Namespace](#)
- [Example Configuration: Mediafire Namespace](#)

Example Configuration: Megaupload Namespace

Use this **namespace** configuration for dynamic URI mapping for Megaupload videos:

```
namespace <megaupload>
  delivery protocol http cache-index domain-name exclude
  delivery protocol http client-request cache-hit action revalidate-
    always
  delivery protocol http client-request cache-index url-match /files/[^/
    ]+/(.*) map-to /$1 no-match-tunnel
  delivery protocol http origin-request x-forwarded-for disable
  delivery protocol http origin-fetch cache-directive no-cache override
  delivery protocol http origin-fetch content-store media object-size 0
  delivery protocol http origin-fetch cache-age-default 28800
  delivery protocol http origin-request host-header inherit incoming-req
    deny
  delivery protocol http origin-request cache-revalidation permit method
    head validate-header Etag
  domain regex "^www.*\.megaupload\.com"
  match uri /files
  origin-server http follow header "Host"
  status active
  exit
```

Example Configuration: Hotfile Namespace

Use this **namespace** configuration for dynamic URI mapping for Hotfile videos:

```
namespace <hotfile>
  delivery protocol http cache-index domain-name exclude
```

```

delivery protocol http client-request cache-index url-match /get/[^/
  ]+/[^/]+/[^/]+/(.*) map-to /$1 no-match-tunnel
delivery protocol http origin-request x-forwarded-for disable
delivery protocol http origin-fetch cache-age-default 28800
delivery protocol http origin-request host-header inherit incoming-req
  deny
delivery protocol http origin-fetch cache-directive no-cache override
delivery protocol http origin-fetch tunnel-override object-expired
domain regex "^.*\hotfile\.com"
match uri /get
origin-server http follow header "Host"
status active
exit

```

Example Configuration: Rapidshare Namespace

Use this **namespace** configuration for dynamic URI mapping for Rapidshare videos:

```

namespace <rapidshare>
  delivery protocol http cache-index domain-name exclude
  delivery protocol http client-request cache-index url-match /cgi-bin/
    [^/]+download\&[^&]+\&[^&]+\&([^&]+)\&([^&]+)\&.* map-to /$1/
    $2 no-match-tunnel
  delivery protocol http client-request query-string action cache
    exclude-query-string
  delivery protocol http origin-request x-forwarded-for disable
  delivery protocol http origin-fetch cache-age-default 28800
  delivery protocol http origin-request host-header inherit incoming-req
    deny
  delivery protocol http origin-fetch content-store media object-size 0
domain regex "^.*\rapidshare\.com"
match uri /cgi-bin
origin-server http follow header "Host"
status active
exit

```

Example Configuration: Filesonic Namespace

Use this **namespace** configuration for dynamic URI mapping for Filesonic videos:

```

namespace <filesonic>
  delivery protocol http cache-index domain-name exclude
  delivery protocol http client-request cache-index url-match /download/
    ([^/]+)/.* map-to /$1 no-match-tunnel
  delivery protocol http origin-request x-forwarded-for disable
  delivery protocol http origin-fetch tunnel-override object-expired
  delivery protocol http origin-fetch cache-age-default 28800
  delivery protocol http origin-request host-header inherit incoming-req
    deny
  delivery protocol http origin-fetch cache-directive no-cache override
domain regex "^s.*\filesonic\.com"
match uri /download
origin-server http follow header "Host"
status active
exit

```

Example Configuration: Putlocker Namespace

Use this **namespace** configuration for dynamic URI mapping for Putlocker videos:

```
namespace <putlocker>
  delivery protocol http cache-index domain-name exclude
  delivery protocol http client-request cache-index url-match /download/
    ([^/]+)/([^/]+) map-to /$1/$2 no-match-tunnel
  delivery protocol http origin-request x-forwarded-for disable
  delivery protocol http origin-fetch tunnel-override object-expired
  delivery protocol http origin-fetch cache-age-default 28800
  delivery protocol http origin-request host-header inherit incoming-req
    deny
  delivery protocol http origin-fetch cache-directive no-cache override
  domain regex "^media-.*\.putlocker\.com"
  match uri /download
  origin-server http follow header "Host"
  status active
  exit
```

Example Configuration: Mediafire Namespace

Use this **namespace** configuration for dynamic URI mapping for Mediafire videos:

```
namespace <mediafire>
  delivery protocol http cache-index domain-name exclude
  delivery protocol http client-request cache-index url-match /[a-z0-
    9]{12}/([a-z0-9]{11}/.*) map-to /$1 no-match-tunnel
  delivery protocol http origin-request x-forwarded-for disable
  delivery protocol http origin-fetch tunnel-override object-expired
  delivery protocol http origin-fetch cache-age-default 28800
  delivery protocol http origin-request host-header inherit incoming-req
    deny
  delivery protocol http origin-fetch cache-directive no-cache override
  domain regex "^download.*\.mediafire\.com"
  match uri /
  origin-server http follow header "Host"
  status active
  exit
```

CHAPTER 7

Configuring Media Flow Controller Load Balancing

- [Media Flow Controller Load Balancing Overview](#)
- [Load Balancing with Direct Server Return](#)

Media Flow Controller Load Balancing Overview

Load balancing allows you to distribute incoming requests across two or more Media Flow Controllers, providing scalability and high availability of the service. Load balancers may operate in Layer 4 (L4) or Layer 7 (L7) mode, when distributing load across Media Flow Controllers.

- Layer 4 load balancer: Requests are routed to the Media Flow Controller by a load balancer or an L4 switch or router. When Media Flow Controller serves user requests, the responses go through the L4 load balancer, so there may be some performance impact in this deployment mode.
- Layer 4 load balancer + Direct Server Return (DSR): This is one of the most popular modes as it allows deployment with an inexpensive load balancer. DSR allows the return data (response) to go from Media Flow Controller to the client directly. This allows scaling from to 10Gbps with a relatively inexpensive load balancer. The load balancer itself need not scale to 10Gbps; as long as it has enough bandwidth to handle the rate of incoming requests, this provides a good solution.
- Layer 7 load balancer: DSR does not work in this case. The load balancer must match the sum of the capacity of all the Media Flow Controllers to which the load balancer is load balancing. However, rich Layer 7 policies based on URI, header, and so forth, can be built on the load balancer, and traffic steering can be done in a more flexible way.

Media Flow Controller can be configured for Direct Server Return (DSR), when load balancers distribute requests across caches in L4 mode. Load balancing with Direct Server Return (DSR) reduces traffic load on the load balancer because the response from the target server bypasses the load balancer and goes directly to the client. This chapter describes how Media Flow Controllers can be configured to work in L4/DSR mode.

Load Balancing with Direct Server Return

Load balancing with Direct Server Return (DSR) reduces traffic load on the load balancer because the response from the target server bypasses the load balancer and goes directly to

the client. In order to implement DSR in Media Flow Controller, the following configurations are required.

1. The server load balancer (SLB) and Media Flow Controller must be Layer 2 adjacent.
2. Media Flow Controller must have the destination load balancer virtual IP address (VIP) configured on a loopback or a network interface that will not broadcast that IP address on the network.
3. Media Flow Controller must not GARP (gratuitous address resolution protocol) the VIP address with its own MAC (media access control) address. You can disable ARP on a Media Flow Controller interface with **interface <interface_name> arp disable**.
4. The return response from Media Flow Controller must bypass the SLB.
5. Media Flow Controller responses are routed to a host *not* Layer 2 adjacent, via your configured route or gateway.

Examples:

- Layer 4 load balancer + DSR: This is one of the most popular modes as it allows deployment with an inexpensive load balancer. DSR allows the return data (response) to go from Media Flow Controller to the client directly. This allows scaling from to 10Gbps with a relatively inexpensive load balancer. The load balancer itself need not scale to 10Gbps; as long as it has enough bandwidth to handle the rate of incoming requests, this provides a good solution.
- Layer 7 load balancer: DSR does not work in this case. The load balancer must match the sum of the capacity of all the Media Flow Controllers to which the load balancer is load-balancing. However, rich Layer 7 policies based on URI, header, and so forth, can be built on the load balancer, and traffic steering can be done in a more flexible way.

See [Figure 8, "Direct Server Return,"](#) for illustration.

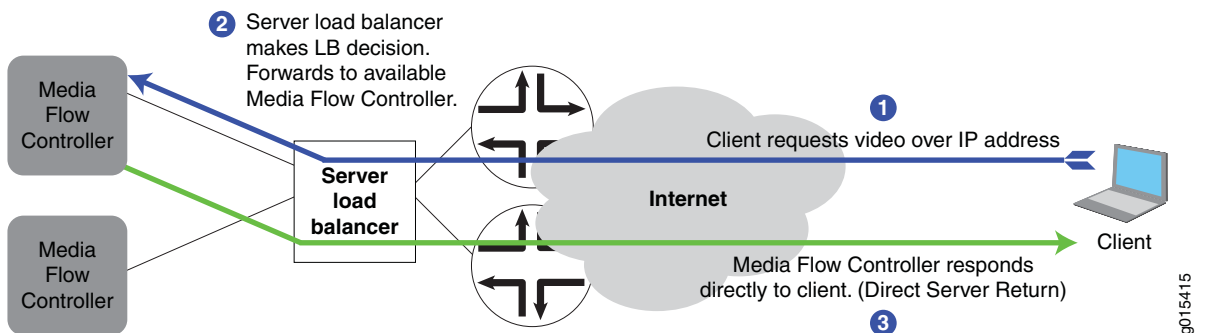


Figure 8 Direct Server Return

g015415

CHAPTER 8

Configuring Media Flow Controller Server Maps

- [Media Flow Controller Server Map Overview](#)
- [Origin Clustering Using Media Flow Controllers](#)
- [Creating the cluster-map XML File](#)
- [Origin Server Load Distribution and Failover](#)
- [Creating the origin-escalation-map XML File](#)
- [Creating the host-origin-map XML File](#)
- [Creating the nfs-map XML File](#)
- [Configuring Server Maps \(CLI\)](#)
- [Example: HTTP host-origin-map Configuration](#)
- [Example cluster-map Configuration](#)
- [Example: origin-escalation-map Configuration](#)

Media Flow Controller Server Map Overview

Media Flow Controller can use an XML file to specify various origin fetch schemes; the XML file can reside in an external server. The format types and syntax for the XML files that the **server-map** command provides access to are described in this chapter. After you define the XML file, then go to the Media Flow Controller CLI for **namespace** and configure the **origin-server <protocol> server-map** option with the **format-type** and file URL of the XML mapping file that you defined.

Server Map Format Types

Media Flow Controller provides four different **server-map format-type** options:

- **host-origin-map**—Use this server map type to configure multiple HTTP origin servers. The hostname (Host header) in the incoming request is matched to the Host entry you define in the XML file. The matched XML **Host** entry denotes the target origin server hostname and port via the associated XML **Origin** and **Port** entries. You cannot use another **server-map** with this **format-type**.
- **cluster-map**—Use this server map type to distribute incoming requests across a cluster of origin servers using consistent hashing to bind objects to nodes, or to create a Cluster Layer 7 redirect configuration. You can use this **format-type** with the **origin-escalation-map** to create a server map hierarchy, or you can use this map alone.
- **origin-escalation-map**—Use this server map type to configure multiple redundant HTTP origin servers for failover protection. Requests are sequentially initiated to specific

origin servers based on a configured weight, until the request is satisfied or all available origin servers have been tried. You can use this **format-type** with **cluster-map** to create a server map hierarchy, or you can use this map alone.

- **nfs-map**—Use this server map type to configure multiple NFS publishing points for origin. You cannot use another **server-map** with this **format-type**.



NOTE: For **cluster-map** and **origin-escalation-map** only, you can assign up to three, in any combination (together or separately), as needed.

Origin Clustering Using Media Flow Controllers

A cluster is a group of Media Flow Controller appliances, or nodes. You can deploy Media Flow Controllers as origin clusters within a point of presence (POP) to increase the cache storage capacity. To do this, you use a consistent hashing scheme to bind objects to origin servers (also referred to as a cluster) using an XML file server map. Additionally, you can create an origin server node map for origin escalation: if the target origin server fails, another configured origin server is automatically chosen. Both of these configurations are achieved through the creation of a server map (**format-type cluster-map** and **format-type origin-escalation-map**) that is then associated with a **namespace**.



NOTE: The hash scheme and origin server data distribution must be pre-configured.

A consistent hash scheme is used by the Media Flow Controller with the **cluster-map** node definitions (and, optionally, **namespace cluster-hash** configuration) to map to the target origins. In the case where no origins exist due to network connectivity issues, an alternative set of origin servers can be consulted via an **origin-escalation-map**, or another **cluster-map**, to resolve the request. See [“Origin Server Load Distribution and Failover” on page 164](#) for information on origin escalation.

Consistent hash-based cluster properties are as follows:

- No inter-cache communication required.
- Ability to identify the target cache server for redirection via strict computation.
- Uniform distribution amongst the caches.
- Object stickiness after cluster re-configuration.
 - After a node deletion, existing entries map to the same node and objects associated with the deleted node are uniformly distributed amongst the remaining nodes.
 - After an addition of a preexisting node, entries map to the same nodes as observed prior to the node deletion.
 - After an addition of a new node, an equal portion of the address space is remapped to the new node resulting in a uniform distribution amongst the nodes. Existing entries may be moved to the new node.

You can use the two server-map types together to create a server hierarchy of server-maps consisting of multiple instances of **cluster-map** and **origin-escalation**; up to three server

maps of either type can be assigned to a single namespace. The order in which the maps are added to the **namespace** denotes the order in which they are read.

Creating the cluster-map XML File

Use the **server-map format-type cluster-map** to distribute incoming requests across a cluster of origin servers. This server-map type calls for specific definition of a consistent hashing scheme used to bind objects to nodes. For background information on consistent hashing, see the following: <http://www8.org/w8-papers/2a-webserver/caching/paper2.html>.

See **server-map** for CLI details.

Consistent hash cluster requirements:

- A cluster is one Level 1 node and \geq one Level 2 nodes.
- Level 1 node must be a Media Flow Controller operating as a reverse proxy.

Sample cluster-map XML File for Origin-Based Clusters

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE ClusterMap SYSTEM "ClusterMap.dtd">
<ClusterMap>
  <Header>
    <Version>1.0</Version>
    <Application>MapXML</Application>
  </Header>
  <ClusterMapEntry>
    <Node>NodeName1</Node>
    <IP>172.19.172.52</IP>
    <Port>80</Port>
    <Options>heartbeatpath=/root</Options>
  </ClusterMapEntry>
  <ClusterMapEntry>
    <Node>NodeName2</Node>
    <IP>172.19.172.53</IP>
    <Port>81</Port>
    <Options>heartbeatpath=/root</Options>
  </ClusterMapEntry>
</ClusterMap>
```

To create the **cluster-map** XML file:

1. Open a text editor and copy and paste the above example into the file. Change all the tag variables and add entries as needed; the following are the tags you can use:
 - **ClusterMap**—Parent tag for **Header** and **ClusterMapEntry** tags.
 - **Header**—Parent tag for **Version** and **Application** tags.
 - **Version**—MapXML utility version: **1.0**.
 - **Application**—Type of Media Flow Controller application: **MapXML**.

- **ClusterMapEntry**—Parent tag for **Node**, **IP**, **Port**, and **Options** tags.
 - **Node**—Name to uniquely identify cluster nodes within the XML file. Use and scope is only within the XML file.
 - **IP**—IP address of the node (hostnames not allowed).
 - **Port**—TCP port of the node.
 - **Options**—**heartbeatpath** = Relative URI to use to heart beat the nodes; for Origin-Based Clusters, this can be left at **/root**. For Cluster Layer 7 Redirect, the heartbeatpath must be **:8080/cmm-node-status.html**

*Do not change the **Version** from **1.0** or the **Application** from **MapXML** without explicit, new instructions. These values must be present or the XML file will be rejected.*

2. Validate your XML syntax against the DTD by running the following under linux:

```
xmllint --noout --dtdvalid <DTD filename> <XML filename>
```

Example:

```
xmllint --noout --dtdvalid ./dtd/ClusterMap.dtd ClusterMap.xml
```

Ignore the following xmllint warning:

```
ClusterMap.xml:2: warning: failed to load external entity "ClusterMap.dtd"
```

Server Map Example: cluster-map DTD

Document Type Definition (DTD) for **cluster-map**. "PCDATA" indicates data that the XML parser fills in after reading your file. You can use this to validate your XML as described. For an example host-origin-map.xml file, see [“Creating the cluster-map XML File” on page 163](#).

```
<!-- ClusterMap 1.0 DTD, Copyright (c) 2010 by Juniper Networks, Inc -->
<!ELEMENT ClusterMap (Header*, ClusterMapEntry*)>
  <!ELEMENT Header (Version, Application)>
    <!ELEMENT Version (#PCDATA)>
    <!ELEMENT Application (#PCDATA)>
  <!ELEMENT ClusterMapEntry (Node, IP, Port, Options?)>
    <!ELEMENT Node (#PCDATA)>
    <!ELEMENT IP (#PCDATA)>
    <!ELEMENT Port (#PCDATA)>
    <!ELEMENT Options (#PCDATA)>
```

Origin Server Load Distribution and Failover

Media Flow Controller lets you to create an origin server node map for origin escalation: if the target origin server fails or returns a configured HTTP code requiring escalation, another configured origin server is automatically chosen. This is achieved through the creation of a server map (**format-type origin-escalation-map**) that is then associated with a **namespace**.

In the cache-miss case, the **origin-escalation-map** server map is consulted.

Origin escalation is a configuration consisting of <N> origin servers which are logically viewed as one, where requests are sequentially initiated to specific origin servers (based on a configured **weight**), until the request is satisfied or all known available origin servers at

request initiation time have been tried. An origin server request is re-initiated to the next configured origin server (escalation) when network connectivity errors are received or when a specific, configured, origin server response code is received (for example, HTTP 404). The origin servers are tried in the order in which they are listed in the XML origin escalation map file, and the order of in which the maps are added to the **namespace** denotes the order in which they are read. You can add up to three origin-escalation maps to a namespace.

Additionally, you can create a simple **host-origin-map** server map for load distribution, but not failover.

Creating the origin-escalation-map XML File

The **origin-escalation-map** functionality determines which defined origins are online; if an origin is not online, it automatically moves to the next defined origin, based on defined **weight**. Under origin escalation, the specified **weight** of each origin defined in the map denotes the order in which requests are tried until a success is received or all nodes have been tried. Escalation occurs when a network connectivity error is encountered, or when an origin returns an HTTP return code specified in the XML file configuration.

See [server-map](#) for CLI details.

Origin escalation requirements:

- All origin servers are viewed as a single entity where any origin server is capable of resolving a miss or handling a validate request.
- All origin servers are monitored via a periodic heartbeat (HTTP request) insuring that escalation only occurs to currently online members.
- Origin server HTTP response codes resulting in escalation are configurable on a per origin server basis.
- Applicable only to a Media Flow Controller reverse proxy configuration.

Sample **origin-escalation-map** XML file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE OriginEscalationMap SYSTEM "OriginEscalationMap.dtd">
<OriginEscalationMap>
  <Header>
    <Version>1.0</Version>
    <Application>MapXML</Application>
  </Header>
  <OriginEscalationMapEntry>
    <Origin>dest.xxx.com</Origin>
    <Port>80</Port>
    <Options>heartbeatpath=/hello.html,weight=1,
      http_response_failure_codes=404;500</Options>
  </OriginEscalationMapEntry>
  <OriginEscalationMapEntry>
    <Origin>dest.yyy.com</Origin>
    <Port>81</Port>
    <Options>heartbeatpath=/hello2.html,weight=2,
```

```

        http_response_failure_codes=404;500</Options>
    </OriginEscalationMapEntry>

    <OriginEscalationMapEntry>
        <Origin>dest.zzz.com</Origin>
        <Port>82</Port>
        <Options>heartbeatpath=/hello3.html,weight=3,
            http_response_failure_codes=404;500;503</Options>
    </OriginEscalationMapEntry>
</OriginEscalationMap>

```

To create the **origin-escalation-map** XML file:

1. Open a text editor and copy and paste the above example into the file. You can define up to 64 nodes in a single **origin-escalation-map**; if you have more than 64 nodes you want to use for origin escalation, you can define additional maps (up to 3 server maps can be added to a namespace). Change all the tag variables and add entries as needed; the following are the tags you can use:

- **OriginEscalationMap**—Parent tag for **Header** and **OriginEscalationMapEntry** tags.
 - **Header**—Parent tag for **Version** and **Application** tags.
 - **Version**—MapXML utility version: **1.0**.
 - **Application**—Type of Media Flow Controller application: **MapXML**.
 - **OriginEscalationMapEntry**—Parent tag for **Origin**, **Port**, and **Options** tags.
 - **Origin**—Domain name or IP address of origin server.
 - **Port**—TCP port of origin server.
 - **Options**
 - **heartbeatpath** = Relative URI to use to heart beat node.
 - **http_response_failure_codes** = Codes that trigger escalation.
 - **weight=<integer>** = Order of escalation; highest priority is the lowest value.

Do not change the **Version** from **1.0** or the **Application** from **MapXML** without explicit, new instructions. These values must be present or the XML file will be rejected.

2. Validate your XML against the DTD by running the following under linux:

```
xmllint --noout --dtdvalid <DTD filename> <XML filename>
```

Example:

```
xmllint --noout --dtdvalid ./dtd/OriginEscalationMap.dtd
    OriginEscalationMap.xml
```

Ignore the following xmllint warning:

```
OriginEscalationMap.xml:2: warning: failed to load external entity
    "OriginEscalationMap.dtd"
```

Server Map Example origin-escalation-map DTD

Document Type Definition (DTD) for **origin-escalation-map**. “PCDATA” indicates data that the XML parser fills in after reading your file. You can use this to validate your XML as

described. For an example host-origin-map.xml file, see [“Creating the origin-escalation-map XML File” on page 165](#).

```
<!-- OriginEscalationMap 1.0 DTD, Copyright (c) 2010 by Juniper Networks, Inc
-->
<!ELEMENT OriginEscalationMap (Header*, OriginEscalationMapEntry*)>
  <!ELEMENT Header (Version, Application)>
    <!ELEMENT Version (#PCDATA)>
    <!ELEMENT Application (#PCDATA)>
  <!ELEMENT OriginEscalationMapEntry (Origin, Port, Options?)>
    <!ELEMENT Origin (#PCDATA)>
    <!ELEMENT Port (#PCDATA)>
    <!ELEMENT Options (#PCDATA)>
```

Creating the host-origin-map XML File

Use the **server-map format-type host-origin-map** XML file to map the incoming (target origin) HOST header value to a specified origin server and (optionally) port. No check is made to determine if an origin is online when requested. There is no maximum on how many origins you configure in this file. The XML file must provide the following tags.

Sample **host-origin-map** XML file:

```
<?xml version="1.0" encoding="ISO-8859-1"?>
<!DOCTYPE HostOriginMap SYSTEM "HostOriginMap.dtd">
<HostOriginMap>
  <Header>
    <Version>1.0</Version>
    <Application>MapXML</Application>
  </Header>
  <HostOriginEntry>
    <Host>host1.xxx.yyy.com</Host>
    <Origin>host1.com</Origin>
    <Port>80</Port>
  </HostOriginEntry>
  <HostOriginEntry>
    <Host>host2.aaa.bbb.com</Host>
    <Origin>host2.com</Origin>
    <Port>81</Port>
  </HostOriginEntry>
  <HostOriginEntry>
    <Host>host3.ccc.ddd.com</Host>
    <Origin>host3.com</Origin>
    <Port>82</Port>
  </HostOriginEntry>
</HostOriginMap>
```

To create the **host-origin-map** XML file:

1. Open a text editor and copy and paste the above example into the file. Change all the tag variables and add entries as needed; these are the tags you can use:

- **HostOriginMap**—Parent tag for **Header** and **HostOriginEntry** tags.
 - **Header**—Parent tag for **Version** and **Application** tags.
 - **Version**—MapXML utility version:**1.0**.
 - **Application**—Type of Media Flow Controller application: **MapXML**.
 - **HostOriginEntry**—Parent tag for **Host**, **Origin**, **Port**, and **Options** tags.
 - **Host**—The incoming request HOST header (target origin).
 - **Origin**—The origin-server to use to resolve cache misses.
 - **Port**—The origin server port to use to resolve cache misses.
 - **Options**—Not supported in Release 2.0.7.

Do not change the **Version** from **1.0** or the **Application** from **MapXML** without explicit, new instructions. These values must be present or the XML file will be rejected.

2. Validate your XML syntax against the DTD by running the following under linux:

```
xmllint --noout --dtdvalid <DTD filename> <XML filename>
```

Example:

```
xmllint --noout --dtdvalid ./dtd/HostOriginMap.dtd HostOriginMap.xml
```

Ignore the following xmllint warning:

```
HostOriginMap.xml:2: warning: failed to load external entity
"HostOriginMap.dtd"
```

Server Map Example: host-origin-map DTD

Document Type Definition (DTD) for **host-origin-map**. “PCDATA” indicates data that the XML parser fills in after reading your file. You can use this to validate your XML as described. For an example host-origin-map.xml file, see [“Creating the host-origin-map XML File” on page 167](#).

```
<!-- HostOriginMap 1.0 DTD, Copyright (c) 2010 by Juniper Networks, Inc. -->
<!ELEMENT HostOriginMap (Header*, HostOriginEntry*)>
  <!ELEMENT Header (Version, Application)>
    <!ELEMENT Version (#PCDATA)>
    <!ELEMENT Application (#PCDATA)>
  <!ELEMENT HostOriginEntry (Host, Origin, Port)>
    <!ELEMENT Host (#PCDATA)>
    <!ELEMENT Origin (#PCDATA)>
    <!ELEMENT Port (#PCDATA)>
    <!ELEMENT Options (#PCDATA)>
```


Creating the nfs-map XML File

Use the **server-map format-type nfs-map** XML file to provide a list of NFS mount-points; mainly, hostname and publishing point, and some other information. The XML file must provide the following tags.

Sample **nfs-map** XML file:

```
<response scode="0" scode_description="success!">
<PUBLISHINGPOINTS INTERVAL-SEC="3600">
<MISSINGFILE PATH="C:\path1\path2\path3\FileNotFound.wma"/>
<INVALIDPUBLISHINGPOINT PATH="C:\path1\path2\path3\StorageOffline.wma"/>
<PUBLISHINGPOINT NAME="name1" PATH="\\<name>.<name>.<name>.com\name1"/>
<PUBLISHINGPOINT NAME="name2" PATH="\\<name>.<name>.<name>.com\name2"/>
<PUBLISHINGPOINT NAME="name3" PATH="\\<name>.<name>.<name>.com\name3"/>
<PUBLISHINGPOINT NAME="name4" PATH="\\<name>.<name>.<name>.com\name4"/>
</PUBLISHINGPOINTS>
</response>
```



NOTE: Media Flow Controller NFS origin server map uses a proprietary parser to extract data from the XML file; the file must use the tags and format described. There is no DTD available to use to validate this type map.

To create the **nfs-map** XML file:

1. Open a text editor and copy and paste the above example into the file. Change all the tag variables and add entries as needed; the following are the tags you can use:
 - **Response Scode**—The standard response code when indicating success in calling the NFS service; you can modify the description.
 - **PublishingPoints Interval-Sec**—The polling interval; you can modify the number of seconds. This can also be set in the Media Flow Controller CLI with the **server-map** command; the CLI setting overrides the XML file setting.
 - **MissingFile Path**—A file to use in the case of a missing file path; only needed if you will be handling streaming of Windows Media content.
 - **InvalidPublishingPoint Path**—A file to use in the case of an invalid publishing point; only needed if you will be handling streaming of Windows Media content.
 - **PublishingPoint Name**—The NFS mount point for that file path. Example:

```
<PUBLISHINGPOINT NAME="<name>"
PATH="\\<name>.<name>.<name>.com\<mount_point>" />
```



CAUTION: The defined Publishing Point Names are expected to be the first component in the incoming URI request. For example, if you define a Publishing Point Name “**p12s34**” (<PUBLISHINGPOINT NAME=“**p12s34**”>), then incoming URI requests must include that name in the first part of the URI for the mapping of that publishing point to be successful; for example, **http://p12s34.example.com/path/filename**.

Configuring Server Maps (CLI)

In the Media Flow Controller CLI, after you have created the necessary XML file, you define the server map with the **server-map** command, and then assign it to a defined **namespace**. To define a server map:

1. First, create the server-map with a name:

```
server-map <name>
```

You are entered to prefix mode.

2. Next, set the **format-type**.

- If this server-map will be assigned to a namespace with **origin-server nfs**, and you want multiple NFS origin servers, set **format-type** to **nfs-map**.
- If this server-map will be assigned to a namespace with **origin-server http**, and you want multiple HTTP origin servers (without consistent hashing), set the **format-type** to **host-origin-map**.
- If this server-map will be assigned to a namespace with **origin-server http**, and you want to distribute requests across a cluster of origin servers using consistent hashing to bind objects to nodes, set the **format-type** to **cluster-map**.
- If this server-map will be assigned to a namespace with **origin-server http**, and you want to configure origin escalation, set the **format-type** to **origin-escalation-map**.

```
format-type <type>
```

The prefix prompt is re-displayed.

3. If you are using a **cluster-map** or **origin-escalation-map**, optionally set additional **allowed-fails** and **node-monitoring** options:

- **allowed-fails**—Specify how many request failures are allowed before the node is declared down. Default is **3**, minimum allowed value is **0** (zero); maximum value is **32**.
- **node-monitoring**—Set node monitoring options:
 - **connect-timeout**—Specify the allowable time in milliseconds for the socket connect to complete.
 - **heartbeat-interval**—Specify the time in milliseconds for nodes to wait before sending a “heartbeat” signal to the other nodes indicating availability status.
 - **read-timeout**—Specify the allowable time in milliseconds for the socket read to complete after the connection is established.

```
allowed-fails <integer> node-monitoring <option> <value>...
```

The prefix prompt is re-displayed.

4. Set a **file-url** for the server-map; this tells Media Flow Controller where to go to fetch the XML mapping file. Also, set how often Media Flow Controller refreshes the XML file; the default **refresh-interval** is **0** (zero), which means no refresh. Refresh interval can also be set in the XML mapping file; the CLI setting overrides the XML file setting.

```
file-url <URL> refresh-interval <time>
```

Media Flow Controller immediately issues a request to the **file-url** you configured to verify the XML file. If there is a problem; for example, Media Flow Controller cannot find the file, an error message is displayed. Otherwise, a confirmation of the file is displayed.

5. The final step is to assign the **server-map** to a **namespace**, and set options as applicable. You can assign up to three server-maps of the **origin-escalation** or **cluster-map** type and; you can use the same server map in multiple namespaces.

```
namespace <name> match uri <uri-prefix> origin-server <protocol> server-  
map <name>
```

Example cluster-map Configuration

```
MFC (config) # server-map clusterSF  
MFC (config server-map clusterSF) # format-type cluster-map  
MFC (config server-map clusterSF) # file-url http://example.com/nfs/path/  
cluster.xml refresh-interval 60  
MFC (config server-map clusterSF) # exit  
MFC (config) # show server-map clusterSF  
Server-map : clusterSF  
    Format-type : cluster-map  
    Map File : http://example.com/nfs/path/cluster.xml  
    Refresh Interval : 60  
MFC (config) # namespace newCM  
MFC (config namespace newCM) # match uri /  
MFC (config namespace newCM) # origin-server http server-map clusterSF  
MFC (config namespace newCM) # exit  
MFC (config) #
```

Example: origin-escalation-map Configuration

```
MFC (config) # server-map OE1  
MFC (config server-map OE1) # format-type origin-escalation-map  
MFC (config server-map OE1) # file-url http://example.com/nfs/path/  
originEscalation.xml refresh-interval 60  
MFC (config server-map OE1) # exit  
MFC (config) # show server-map OE1  
Server-map : OE1  
    Format-type : origin-escalation-map  
    Map File : http://example.com/nfs/path/originEscalation.xml  
    Refresh Interval : 60  
MFC (config) # namespace newOE  
MFC (config namespace newOE) # match uri /  
MFC (config namespace newOE) # origin-server http server-map OE1  
MFC (config namespace newOE) # exit  
MFC (config) #
```

Example: HTTP host-origin-map Configuration

```
MFC (config) # server-map newMap  
MFC (config server-map newMap) # format-type host-origin-map  
MFC (config server-map newMap) # file-url http://example.com/vod/  
newHostOriginMap.xml refresh-interval 9000
```

```
MFC (config server-map newMap) # exit
MFC (config) # show server-map newMap
Server-map : newMap
  Format-type : host-origin-map
  Map File : http://example.com/vod/newHostOriginMap.xml
  Refresh Interval : 9000
MFC (config) # namespace newTest
MFC (config namespace newTest) # match uri /
MFC (config namespace newTest) # origin-server http server-map newMap
MFC (config namespace newTest) # exit
MFC (config) #
```

CHAPTER 9

Configuring and Using Media Flow Controller Logs and Alarms

- [Media Flow Controller Logging Overview](#)
- [Media Flow Controller Log Codes and Sub-Codes](#)
- [Configuring Media Flow Controller Service Logs Overview](#)
- [Configuring Media Flow Controller Service Logs \(CLI\)](#)
- [Reading Media Flow Controller Service Logs Overview](#)
- [Configuring Media Flow Controller System Log](#)
- [Reading the Media Flow Controller System Log](#)
- [Reading the Media Flow Controller Tech-Support Log](#)
- [Configuring Media Flow Controller Log Statistics Thresholds \(CLI\)](#)
- [Configuring Media Flow Controller Stats Alarms](#)
- [Configuring Media Flow Controller Fault Notifications \(CLI\)](#)

Media Flow Controller Logging Overview

Use the Media Flow Controller system log (also referred to as syslog) to discover system-level information such as who has logged in and what commands they issued. Use the Media Flow Controller service logs, `accesslog`, `cachelog`, `errorlog`, `fmsaccesslog`, `fmsedgelog`, `fuselog`, `publishlog`, and `streamlog`, to gather information on various aspects of the Media Flow Controller service. Use the tech-support log to send Customer Support performance information. Use the `tracelog` to follow the path of a particular request.

System Baseline and Health

Logging can provide more information than any one person can process during any given moment; finding the required information can be a daunting task. This problem can be reduced if you take the time with each installed system to perform a baseline audit. A baseline audit identifies normal activity for your system, normal system log entries, normal network traffic for the system, and how the system reacts to certain conditions.

A baseline audit helps you get the most value from syslog data. In particular, a baseline audit helps identify what the system was like before you became suspicious of its behavior and helps you determine what has changed. On a day to day basis the system logs can be used to determine network health and how well the network and computer systems are running. This provides the ability to proactively solve issues before the user is aware.

Media Flow Controller Log Codes and Sub-Codes

These status codes and sub-codes may appear in the accesslog, errorlog, streamlog or other messages.

Status and Error Codes

[Table 15](#) describes the standard status codes that might be returned in Media Flow Controller logs. See [Table 16](#), for status sub-codes and [Table 17](#) for Media Flow Publisher error codes.

Table 15 Logging Status (%s) HTTP Codes

| Code | Description |
|------|--|
| 100 | Continue |
| 200 | OK (Success) |
| 201 | Created |
| 206 | Partial Content downloaded |
| 250 | Low on Storage Space |
| 300 | Multiple Choices |
| 301 | Moved Permanently |
| 302 | Moved Temporarily |
| 303 | See Other |
| 304 | Not Modified |
| 305 | Use Proxy |
| 400 | (52xxx sub-codes, except as given below) Any HTTP request parser error |
| 401 | Unauthorized |
| 402 | Payment Required |
| 403 | (50007, 52004 sub-codes) Namespace Lookup Failure |
| 404 | Not Found (90001, 90002 sub-codes) When origin server is NFS and file does not exist (52016 sub-code) Any unknown, unexpected socket closure without any sub-status code |
| 405 | Method Not Allowed |
| 406 | Not Acceptable |
| 407 | Proxy Authentication Required |
| 408 | Request Time-out |
| 410 | Gone |
| 411 | Length Required |
| 412 | Precondition Failed |
| 413 | Request Entity Too Large |
| 414 | Request-URI Too Large |
| 415 | Unsupported Media Type |
| 416 | (52005 - 52010 sub-codes) Bad request. Requested range not satisfiable (90003 - 90006 sub-codes) When origin server is NFS and other error code |
| 451 | Parameter Not Understood |
| 452 | Conference Not Found |
| 453 | Not Enough Bandwidth |

Table 15 Logging Status (%s) HTTP Codes (Continued)

| Code | Description |
|------|--|
| 454 | Session Not Found |
| 455 | Method Not Valid in This State |
| 456 | Header Field Not Valid for Resource |
| 457 | Invalid Range |
| 458 | Parameter Is Read-Only |
| 459 | Aggregate operation not allowed |
| 460 | Only aggregate operation allowed |
| 461 | Unsupported transport |
| 462 | Destination unreachable |
| 500 | Internal Server Error (90008 sub-code) When origin server is NFS and NFS server is not mounted |
| 502 | Bad Gateway |
| 503 | (10003 sub-code) Service Unavailable: Media Flow Controller internal server is busy (for example, no CPU, malloc failure, and so forth) (52026 sub-code) Limited by resource-pool configuration |
| 504 | (50001, 50002, 50003 sub-codes) Origin Server is not reachable or not available |
| 505 | RTSP Version not supported |
| 551 | Option not supported |

Status and Error Sub-Codes

The Media Flow Controller status sub-codes are shown in [Table 16](#), see [Table 15](#) for status codes. Successful transactions are indicated by a **0** (zero) for the status sub-code.

Table 16 Additional Logging Status Sub-Codes

| Sub-Code | Module Name HTTP Response Line (if any) | Comments Counters (if any) |
|---|--|--|
| Object Attribute and Server System Error Codes | | |
| 10001 | Internal storage overflow. Media Flow Controller stores all cacheable origin server response headers in a per-object fixed-size name/value pair table; in this case, the table size has been exceeded. | |
| 10002 | Internal storage overflow; in this case, either the name or value length of a name/value entry exceeds the system maximum. | |
| 10003 | NKN_SERVER_BUSY 404 Not Found | Session Admission Control due to internal resource |
| Buffer Manager Error Codes | | |
| 20001 | Internal error. An incorrect parameter (i.e. length, offset, etc.) was specified in a call to the buffer manager. | |
| 20002 | Internal error. Buffer manager failed to get the contents from a provider. | |
| 20003 | NKN_BUF_INVALID_OFFSET 416 Requested range not satisfiable | Either Start or Stop offset in byte range is beyond total content length. glob_cm_inval_off |
| 20004 | NKN_BUF_ORIGIN_EXPIRED | Got expired data from origin. |
| 20005 | NKN_BUF_VERSION_EXPIRED | Version referred to has expired. |
| 20006 | NKN_BUF_WAIT_FAILED | Failed to do timed wait. |
| 20007 | NKN_BUF_WAIT_LOOP | Got repeat timed wait from provider. |

Table 16 Additional Logging Status Sub-Codes (Continued)

| Sub-Code | Module Name HTTP Response Line (if any) | Comments Counters (if any) |
|--|---|--|
| 20008 | NKN_BUF_NOT_AVAILABLE | Requested buffer not available locally. |
| Cache Object Descriptor (COD) Manager Error Codes | | |
| 21001 | Invalid COD entry specified. | |
| 21002 | Entry has unknown version. | |
| 21003 | The request for a given object could not be satisfied because that version is no longer available. Other than a defect, this can occur if Media Flow Controller cached (and served) part of an object and then discovered that the origin modified the object when Media Flow Controller tried to retrieve the remaining portion. Subsequent requests work correctly because Media Flow Controller discards the "stale" portion in the cache. | |
| 21004 | Entry is expired. | |
| Server-Side Player Error Codes | | |
| 30001 | NKN_SSP_BAD_URL | Malformed URI. Virtual player cannot decipher the URL and/or its query parameters |
| 30003 | NKN_SSP_AFR_INSUFF | SSP could not enforce AFR due to unavailable bandwidth and closed the connection |
| 30005 | Connection close directive from SSP. | |
| 30011 | NKN_SSP_HASHCHECK_FAIL | SSP was unable to authenticate the request and closed the connection. |
| 30021 | NKN_SSP_INSUFF_QUERY_PARAM | Incoming request did not match any query parameters. |
| 30023 | NKN_SSP_MISSING_QUERY_PARAM | A required query parameter was absent in the incoming request. |
| 30035 | NKN_SSP_CHUNK_METAFILE_NOT_SUPP | Chunked meta file reads are not supported by SSP from cache. |
| 30037 | NKN_SSP_CONTAINER_FRMT_NOT_SUPP | Unsupported video container format requested for server side processing and/or delivery. |
| 30041 | NKN_SSP_BAD_QUERY_OFFSET | Invalid or unsupported value for a query parameter. |
| 30043 | NKN_SSP_PYTHON_PLUGIN_FAILURE | Call to the Python plug-in was unsuccessful. |
| 30045 | NKN_SSP_CLI_PARAM_UNSPECIFIED | CLI param is not specified as part of configuration |
| 30050 | NKN_SSP_VIDEOID_DISABLED | Video ID field for Type-5 virtual player has not been enabled; cannot cache youtube videos. |
| 30051 | NKN_SSP_BAD_REMAPPED_URL | SSP (server side player) remapped the request URL to an internal URL and received a 404 response from the origin server when attempting a fetch. |
| SmoothStream-Pub Virtual Player Error Codes | | |
| 30404 | NKN_SSP_SS_NULL_BUF_INT_FETCH | An internal fetch for manifest returned an empty buffer. |
| 30408 | NKN_SSP_SS_MFRO_OFF_FAIL | The MFRO (media fragment random offset) was not found in the ISMV / ISMA files. |
| 30400 | NKN_SSP_SS_TRACK_NOT_FOUND | Unable to find a media track in the ISMV package corresponding to the bit rate and track type specified by the manifest. |
| 30402 | NKN_SSP_SS_SVR_MANIFEST_ISM_NOT_FOUND | Server Manifest file was not found as part of the SmoothStreaming package. |

Table 16 Additional Logging Status Sub-Codes (Continued)

| Sub-Code | Module Name HTTP Response Line (if any) | Comments Counters (if any) |
|---|---|--|
| 30406 | NKN_SSP_SS_ISMV_ISMA_NOT_FOUND | The ISMV / ISMA file was not found. |
| 30410 | NKN_SSP_SS_MEM_ALLOC_HDR_FAIL | Memory allocation error. |
| Flashstream-Pub Virtual Player Error Codes | | |
| 30450 | NKN_SSP_FS_URI_FORMAT_ERR | URI does not conform to Flashstream the recommended format. |
| 30452 | NKN_SSP_FS_SEGFRAG_TAG_NOT_FOUND | Mandatory Seg, Frag and or Delimiter tag is missing in request. |
| 30454 | NKN_SSP_FS_F4X_NOT_FOUND | The Flash F4X index file was not found. |
| 30456 | NKN_SSP_FS_MEM_ALLOC_HDR_FAIL | Memory allocation error. |
| 30458 | NKN_SSP_FS_F4X_CONTEXT_FAILURE | Failed to create an F4X context. |
| 30460 | NKN_SSP_FS_F4F_OFFSET_PARSE_ERROR | Failed to parse the fragment offset in the F4F file. |
| Initialization and Disk Management Errors | | |
| 40001 | An unknown disk cache name was given. | |
| 40002 | An issued command is incorrect given the state of the disk. | |
| 40003 | During a status active command, the system could not find the requested device. | |
| 40004 | A media-cache disk <disk_name> deactivate command failed (most likely an un-mount failed). | |
| 40005 | NKN_DM2_DISK_ADMIN_ACTION | A previous command has been given to cause the disk to be in the "cache enabled" state. |
| 40006 | NKN_DM2_BITMAP_NOT_FOUND | A startup error. |
| 40007 | NKN_DM2_EMPTY_CACHE_TIER | A request was sent to a disk tier which has no enabled disks. |
| 40008 | A media-cache disk <disk_name> format command failed. | |
| 40009 | A media-cache disk <disk_name> status active command failed. | |
| 40010 | NKN_DM2_EVICTION_SKIPPED | An eviction request was made but the request had some type of error in it. |
| 40011 | NKN_DM2_OBJ_NOT_YET_WRITTEN | A startup error. |
| 40013 | NKN_DM2_WRONG_CACHE_VERSION | Attempting to enable a cache which has the wrong version number. This should never happen because older versions should be converted to the current version. Media Flow Controller doesn't downgrade cache versions. |
| 40014 | Format conversion failed during a status active or format command. | |
| 41001 | NKN_DM2_DISK_CACHE_NOT_FOUND | An unknown disk cache name was given to the CLI, and Media Flow Controller returns this error. |
| 41002 | NKN_DM2_INVALID_MGMT_REQUEST | The command attempted was not valid for the given state of the disk. This is a generic error. |
| 41003 | NKN_DM2_DISK_DEVICE_NOT_FOUND | During a 'status active' command, the system could not find the requested device. |
| 41004 | NKN_DM2_DISK_DEACTIVATE_FAILED | A media-cache disk <disk> status inactive command was issued and failed. |
| 41005 | NKN_DM2_FORMAT_FAILED | A media-cache disk <disk> format command was issued and failed. |

Table 16 Additional Logging Status Sub-Codes (Continued)

| Sub-Code | Module Name HTTP Response Line (if any) | Comments Counters (if any) |
|-------------------------|--|--|
| 41006 | NKN_DM2_DISK_ACTIVATE_FAILED | A media-cache disk <disk> status active command was issued and failed. |
| 41007 | NKN_DM2_CONVERT_FAILED | When Media Flow Controller is starting or a drive is made 'status active', a possible format conversion is done. If that conversion fails, Media Flow Controller returns this error. |
| 41008 | NKN_DM2_MUST_CACHE_DISABLE | If a drive is in the 'cache enabled' state and a 'status inactive' command is given, we return this error. |
| 41009 | NKN_DM2_MUST_CLEAR_ERROR | If a drive is in an error state of any kind and a 'cache enable' command is given, Media Flow Controller returns this error. |
| 41010 | NKN_DM2_MUST_STATUS_ACTIVE | If a drive is in the 'status inactive' state and a 'cache enable' command is given, we return this error. |
| 50001 | Premature close on an origin server connection due to an inactivity timeout or an unexpected condition. | |
| 50002 | Internal resource allocation error. | |
| 50003 | Target origin server is unavailable. | |
| 50005 | Connection error on attempt to establish a connection to the target origin server. | |
| 50006 | Unexpected condition when building the HTTP request for the target origin server. | |
| 50007 | Unable to determine the namespace associated with the HTTP request when attempting to resolve a cache miss via the origin server provider. | |
| 50008 | Internal system inconsistency when attempting to resolve a cache miss via the origin server provider. | |
| 50009 | Internal error. | |
| 50011 | Parse error processing origin server response headers. | |
| 50012 | Parse error processing Content-Range header. | |
| 50013 | Parsed Content-Range data inconsistent with internal state. | |
| 50015 | Attempt to store query string data failed. | |
| 50016 | Attempt to save object validation specific data failed. | |
| 50018 | Unexpected internal error when attempting to initiate an HTTP GET to the origin server. | |
| 50019 | Internal error between HTTP and OM. | |
| 50020 | Conversion of origin server response data to attribute data failed. | |
| 50021 | System inconsistency detected. Data retrieved by OM on for client is inconsistent with existing data in cache. | |
| 50022 | Attempt to store origin server response data for chunk encoded object failed. | |
| 50023 | Origin manager received an HTTP 404 (Not Found) response from the origin server for a namespace with SSP configured. | |
| 50024 | Attempt to save object validation specific data for chunk encoded object failed. | |
| HTTP Error Codes | | |
| 52002 | Internal error (low system memory). | |
| 52003 | Client sent malformed GET request. | |
| 52004 | NKN_HTTP_NAMESPACE 404 Not Found | Failed to match any namespace in configuration glob_http_namespace_err_52004 |
| 52005 | NKN_HTTP_REQ_RANGE_1 416 Requested Range Not Satisfiable | Either Start or Stop byte range in request is negative. |

Table 16 Additional Logging Status Sub-Codes (Continued)

| Sub-Code | Module Name HTTP Response Line (if any) | Comments Counters (if any) |
|----------|---|--|
| 52006 | NKN_HTTP_REQ_RANGE_2 416 Requested Range Not Satisfiable | Stop offset is less than start offset in byte range. |
| 52007 | Client request byte range inaccurate. | |
| 52010 | Origin server response byte range inaccurate. | |
| 52011 | Requested content type does not match content. | |
| 52012 | NKN_HTTP_URI 400 Bad Request | URI does not exist in the HTTP request. |
| 52014 | URI length must be between 1 and 256 bytes. "URI_MORE means the requested URI exceeded 256 bytes; URI_LESS means it was less than 1 byte. | |
| 52015 | Internal error (problem with seek function). | |
| 52016 | NKN_HTTP_CLOSE_CONN 404 Not Found | Failed to tunnel the request because origin server is NFS or server map. glob_http_close_conn_err_52016 |
| 52018 | NKN_HTTP_NO_HOST_HTTP11 400 Bad Request | HTTP/1.1 request without Host field. |
| 52019 | NKN_HTTP_PARSER 400 Bad Request | Malformed HTTP request, Media Flow Controller cannot parse the request. glob_http_parse_err_nulls glob_http_parse_err_req_rate glob_http_parse_err_uri_tolong glob_http_parse_err_req_toshort glob_http_parse_err_normalize_uri glob_http_parse_err_badreq_1 glob_http_parse_err_badreq_2 glob_http_parse_err_badreq_3 glob_http_parse_err_longcookie glob_http_parse_err_badreq_4 glob_http_parse_err_badreq_5 |
| 52020 | NKN_HTTP_ERR_CHUNKED_REQ 400 Bad Request | Chunked encoding request. |
| 52021 | NKN_HTTP_BAD_HOST_HEADER 400 Bad Request | Domain in HTTP request is not an FQDN domain. |
| 52022 | NKN_HTTP_BAD_URL_HOST_HEADER 400 Bad Request | Domain in HTTP absolute URI is not an FQDN domain. |
| 52023 | NKN_HTTP_UNSUPPORTED_VERSION 505 HTTP Version Not Supported | Request HTTP version is neither 1.0 or 1.1. |

Media Manager Error Codes

| | | |
|-------|-------------------------|--|
| 70001 | NKN_MM_GET_URI_ERR | Error in GET request |
| 70002 | NKN_MM_STAT_URI_ERR | Error in STAT request |
| 70003 | NKN_MM_PROMOTE_PUT_ERR | Error in cache write to disk |
| 70004 | NKN_MM_PROMOTE_GET_ERR | Error in fetching data for ingest or promote |
| 70005 | NKN_MM_PROMOTE_ATTR_ERR | Error in fetching attributes for ingest or promotion |
| 70006 | NKN_MM_PROMOTE_GEN_ERR | Generic error during preparation for ingest or promotion |

Table 16 Additional Logging Status Sub-Codes (Continued)

| Sub-Code | Module Name HTTP Response Line (if any) | Comments Counters (if any) |
|---------------------------------------|--|---|
| 70007 | NKN_MM_COD_ERR | COD open failure during a GET request |
| 70008 | NKN_MM_COD_ERR | COD open failure during a STAT request |
| 70009 | NKN_MM_OBJ_NOT_CACHE_WORTHY | Object not worthy to be ingested into cache |
| 70010 | NKN_MM_TOKEN_ERR | Namespace error during ingest or promotion |
| 70020 | NKN_MM_CONF_RETRY_REQUEST | Retry request from origin manager |
| 70021 | NKN_MM_UNCOND_RETRY_REQUEST | Unconditional retry request from origin manager |
| 70022 | NKN_MM_OBJ_NOT_FOUND_IN_CACHE | Object not found in cache |
| 71002 | Internal error: memory allocation error. | |
| 71003 | Internal network API error. | |
| 71005 | Error in the network socket. | |
| 71006 | Error in network connection. | |
| NFS Origin Manager Error Codes | | |
| 90001 | NKN_NFS_STAT_URI_ERR | Error during a STAT request to NFS origin |
| 90002 | NKN_NFS_GET_URI_ERR | Error during a GET request to NFS origin |
| 90003 | NKN_NFS_STAT_INV_OFFSET | Invalid offset in a STAT or GET request to origin |
| 90005 | NKN_NFS_GET_SERVER_BUSY | Too many GET requests queued for the NFS origin |
| 90007 | NKN_NFS_VERSION_MISMATCH | Object version mismatch happened during a GET request |
| 90008 | NKN_NFS_MOUNT_ERR | Error during NFS mount |

Table 17 Media Flow Publisher Publish Log, Error Codes

| Error Code | Module | Description |
|--|------------------------------|---|
| Generic Media Flow Publisher Errors | | |
| -1 | E_MFP_NO_MEM | Run out of memory |
| -2 | E_MFP_INVALID_FILE | Invalid input/source files |
| -3 | E_MFP_INVALID_SESS | Invalid session ID/ session ID collision |
| -4 | E_MFP_INVALID_CONVERSION_REQ | Unsupported conversion request (FILE conversion only) |
| -5 | E_MFP SOCK_CREATE | Unable to create a socket fd |
| -6 | E_MFP_BIND_FAIL | Unable to bind to port specified in PMF |
| -7 | E_MFP_NO_SPACE | Unable to bind to port specified in PMF |
| PMF Parsing Errors | | |
| -1000 | E_MFP_PMF_INVALID_FILE | Invalid PMF file |
| -1001 | E_MFP_PMF_NO_MEM | Run out of memory |
| -1002 | E_MFP_PMF_INVALID_TAG | Invalid Tag |

Table 17 Media Flow Publisher Publish Log, Error Codes (Continued)

| Error Code | Module | Description |
|-----------------------------------|------------------------------------|--|
| -1003 | E_MFP_PMF_INVALID_ARG | UNUSED |
| -1004 | E_MFP_PMF_UNSUPPORTED_TYPE | Source Type is neither FILE nor LIVE |
| -1005 | E_MFP_PMF_INCOMPLETE_TAG | Missing child tag |
| -1006 | E_MFP_PMF_DOC_PTR_FAILED | libxml unable to parse the file |
| -1007 | E_MFP_PMF_XSD_MISMATCH | XSD compliance fails |
| -1008 | E_MFP_PMF_XSD_INT_ERROR | XSD parser fails |
| -1009 | E_MFP_PMF_INSUFF_MEDIA_TYPE | Missing parameters in the pub scheme |
| -1010 | E_MFP_PMF_INCOMPATIBLE_MEDIA_TYPE | Unsupported Media type tag |
| TS Formatter Errors | | |
| -3001 | E_MFP_FRUIT_INIT | Initialization of APPLE formatter fails |
| -3002 | E_MFP_SL_INIT | Initialization of silverlight fails |
| File Conversion Errors | | |
| -105 | E_VPE_FILE_CONVERT_CONTINUE | Continue file conversion |
| -106 | E_VPE_FILE_CONVERT_STOP | Stop file conversion |
| -107 | E_VPE_FILE_CONVERT_SKIP | Skip file conversion |
| -108 | E_VPE_FILE_CONVERT_CLEANUP_PENDING | Cleanup pending for file conversion |
| -109 | E_VPE_INVALID_FILE_FORMAT | Invalid file format |
| Generic Parser Error Codes | | |
| -201 | E_VPE_PARSER_NO_MEM | Out of memory |
| -202 | E_VPE_PARSER_INVALID_FILE | unable to open file descriptor |
| -203 | E_VPE_PARSER_INVALID_OUTPUT_PATH | invalid output path (unable to write file) |
| -204 | E_VPE_PARSER_FATAL_ERR | Fatal error, invalid state change |
| -205 | E_VPE_PARSER_INVALID_CTX | Invalid context/ state |
| -206 | E_VPE_MAX_PROFILE_ERR | The num:of: profiles exceed the MAXIMUM assigned |
| MP4 Parsing Error Codes | | |
| -301 | E_VPE_MP4_MOOV_ERR | error in parsing MOOV box |
| -302 | E_VPE_MP4_NO_STSS_TAB | No STSS box |
| -303 | E_VPE_NO_AVCC_BOX | No AVCC box |
| -304 | E_VPE_MP4_MISALIGNMENT | Syncpoints across MP\$ profiles are misaligned |
| -305 | E_VPE_MP4_ESDS_ERR | Error in parsing ESDS box |
| -306 | E_VPE_MP4_AV_SYNC_ERROR | Audio and video sync point are not equal |

Table 17 Media Flow Publisher Publish Log, Error Codes (Continued)

| Error Code | Module | Description |
|---|---------------------------------|--|
| -307 | E_VPE_PROFILE_LEVEL_ERROR | Profiles other than baseline 3.0 will throw this error |
| -308 | E_VPE_VIDEO_CODEEC_ERR | Error in parsing video codec data |
| -309 | E_VPE_DIFF_MOOFS_ACROSS_PROFILE | The number of moofs is different across the profiles |
| ISMx Parsing Related Error Codes | | |
| -401 | E_ISMV_INVALID_CTX | Invalid context |
| -402 | E_ISMV_NOMEM | Out of memory |
| -403 | E_ISMV_INVALID_TIMESCALE | Invalid track timescale |
| -404 | E_ISMV_NO_MORE_FRAG | No more fragments to parse |
| -405 | E_ISMC_PARSE_ERR | Required tag missing in ISMC file |
| -406 | E_ISM_PARSE_ERR | Required tag missing in ISM file |
| -407 | E_ISMV_PARSE_ERR | Required tag missing in ISMV file |
| -408 | E_ISMV_MFRA_MISSING | MFRA missing |
| HTTP Sync Reader Error Codes | | |
| -450 | E_VPE_HTTP_SYNC_VER_ERR | Invalid HTTP version number |
| -451 | E_VPE_HTTP_SYNC_PARSE_ERR | HTTP parser error |
| F4V Parser Error Codes | | |
| -501 | E_VPE_F4V_INVALID_F4M | Invalid F4m |
| -502 | E_VPE_F4V_INVALID_FRAG | Invalid Frag |
| -700 | E_VPE_VID_ONLY_NOT_SUPPORTED | Only video is present |
| -701 | E_VPE_AUD_ONLY_NOT_SUPPORTED | Only audio is present |
| TS Creation Error Codes | | |
| -800 | E_VPE_TS_CREATION_FAILED | Failed to create TS |
| -801 | E_VPE_NO_VIDEO_PKT | No video pkts |

Configuring Media Flow Controller Service Logs Overview

- [“About Log Rotation” on page 183](#)
- [“Using Accesslog Copy With SFTP” on page 183](#)
- [“Access Log Format Options” on page 183](#)
- [“Stream Log Format Options” on page 185](#)
- [“Error Log Module Options” on page 187](#)
- [“Error Log Options” on page 187](#)
- [“Configuring Media Flow Controller Service Logs \(CLI\)” on page 188](#)

About Log Rotation

All Media Flow Controller service logs offer rotation options. It is important to remember that rotation parameters are checked *only when activity is written to the log*. If no activity, or very widely spaced activity, is written to the log, the rotation setting may appear to not be honored. For example, if **cachelog on-the-hour enable** is set, and data is written to the cachelog at 1pm and no data is written to it again until 6:30 p.m., the rotation happens when the data is written to it: at 6:30 p.m. If logging continues, rotation happens again at 7 p.m. and so on.



NOTE: The log options **on-the-hour enable** and **rotate time-interval 1**, both set log rotation to every hour; the **on-the-hour** setting takes precedence, if you set both. You cannot set **time-interval** to less than **1** (hour). Additionally, if you set **on-the-hour enable** and **filesize**, the **filesize** setting takes precedence; should this happen between hours, the log is also rotated on the hour (if active).

Using Accesslog Copy With SFTP

Media Flow Controller can auto-upload log files after they reach a certain threshold size using **accesslog copy** with **scp** or **sftp**.

To configure **accesslog copy** with **sftp**:

```
(config) # accesslog copy sftp://user@host:path
```

To verify, use the **upload** command to manually transfer an accesslog file to a target host:

```
(config) # upload accesslog current sftp://root@192.168.1.10:/tmp
sftp> cd /tmp
sftp> put access2.log access2.log.tmp
```

By default, **scp** and **sftp** use password-based authentication which requires user input. To further automate log file auto-uploading, **scp** and **sftp** can be configured to use password-less authentication based on SSH public keys described in [“Using SSH in Automated Scripts \(CLI\)” on page 85](#).

Access Log Format Options

We recommend that you do not change the default format as it conforms to NCSA standard, followed by APACHE, SQUID, and so forth; however, you may do so if you wish. Media Flow Controller is capable of tracking all the fields described. Options are given in [Table 18](#) (use any combination). The default format is:

```
%h %V %u %t "%r" %s %b "%{Referer}i" "%{User-Agent}i" %y
```

Table 18 Accesslog Format Options

| Field | Description | Examples |
|--|---|------------------------------|
| %b (Bytes Out No Header) (default) | The size of the object returned to the client, not including the response headers. If no content was returned to the client, this value is "-" (dash). | 8388608 |
| %c (cache_hit) | Shows from where the object is delivered; for example, Buffer (from buffer cache), Origin (from origin), Tunnel (from tunnel path), SSD (from SSD disk), SAS (from SAS disk), SATA (from STAT disk), NFS (from NFS manager), TFM (from temporary file manager). | Buffer |
| %f (filename) | The requested filename. Not supported in Release 2.0.7. | |
| %h (c-ip client source IP address or remote_host) | The IP address of the client (remote host) that made the request. The IP address reported here is not necessarily the address of the machine at which the user is sitting. If a proxy server exists between the user and the server, this address is the address of the proxy, rather than the originating machine. | 10.15.4.211 |
| %i (header) | Contents of header lines in the request sent to the server. | |
| %m (request_method) | The method the incoming request used. | HTTP |
| %o (response_header) | Contents of the header lines in the response. | |
| %q (query_string) | The query string; prepended with a question mark (?) if a query string exists, otherwise an empty string. | |
| "%r" (request_line) (default) | The request line from the client, including the method, path, query-string, and protocol; this is equivalent to %m %U %H . | "GET /user/8M HTTP/1.0" |
| %s (status) (default) | The request status code the server sends back to the client: <ul style="list-style-type: none"> • Successful response (codes begin at 2), • Redirection (codes begin at 3), • Error (at client, codes begin at 4), at Server (codes begin at 5). See "Status and Error Codes" on page 174 for more details. | 200 |
| %t (timestamp) (default) | The time that the server finished processing the request. The format is day/month/year:hour:minute:second zone: <ul style="list-style-type: none"> • day = 2*digit • month = 3*letter • year = 4*digit • hour = 2*digit • minute = 2*digit • second = 2*digit • zone = ('+' '-') 4*digit | [15/Nov/2010:23:52:14 +0000] |
| %u (remote_user) (default) | Not Supported in Release 2.1. | - |
| %v (server_name) | The canonical ServerName of the server serving the request. | |
| %y (status_subcode) (default) | If present, provides more detail describing the status of the response. | 0 |
| %A (request_in_time) | The time when Media Flow Controller received the request. | 1289865134.735 |

Table 18 Accesslog Format Options (Continued)

| Field | Description | Examples |
|------------------------------|--|--------------------|
| %B (first_byte_out_time) | The time when Media Flow Controller sent the first byte out. | 1289865134.735 |
| %D (time_used_ms) | Time taken to serve the request, in milliseconds. | 60 |
| %E (time_used_in_seconds) | The elapsed real (wall clock) time used by the process. | 0.0 |
| %F (last_byte_out_time) | The time when Media Flow Controller sent the last byte out. | 1289865134.735 |
| %H (request_protocol) | The protocol the incoming request used. | HTTP / 1.1 |
| %I (bytes_in) | Bytes received, including request and headers; cannot be zero. | 300 |
| %M (data_out_ms) | The time when the client receives the full data (the difference between the First Byte Out and the Last Byte Out). | 50 |
| %N (Namespace name) | The namespace referenced in the URL. | youtube-ns |
| %O (bytes_out) | Bytes sent, including headers; cannot be zero. | 400 |
| %U (URL) | The URL path requested, not including any query string. | /data/my_data.html |
| %V (HTTP_host) (default) | The server name according to the UseCanonicalName setting. | mfc-cli22 |
| %X (remote_address) | Socket client IP address. | 10.123.22.14 |
| %Y (local_address) | Media Flow Controller local socket IP address referring to one of the network interfaces. | 10.123.22.211 |
| %Z (server_port) | Media Flow Controller port number; the port through which the client made the successful transaction (usually 80). | 80 |

The **Referrer** and **User-Agent** (default) fields are from the HTTP client request. Example:

```
10.1.1.101 10.1.1.11 - [30/Dec/2008:20:03:54 +0000] "GET /bbb HTTP/1.0"
200 712 "-" "test client"
```

The “referrer” gives the site that the client reports having been referred from; the “user-agent” is the identifying information that the client browser reports about itself.

Stream Log Format Options

Options are (use any combination) shown in [Table 19](#).

Table 19 Streamlog format Options

| Field | Description | Example s |
|-----------------|---|---------------|
| %a (avg b/w) | Average bandwidth in bytes per second. | 35390 |
| %c (s-ip) | Destination IP address in the RTSP request. | 192.155.14.14 |
| %d (r-ip) | Origin server IP address. | 123.14.14.55 |
| %f (filelength) | Length of the stream in seconds. | 55 |
| %h (c-ip) | Client Source IP address. | 207.19.19.20 |
| %i | Player information used in the header (User-Agent). | NSPlayer |

Table 19 Streamlog **format** Options (Continued)

| Field | Description | Example s |
|--------------------------|---|------------------------------------|
| %j (timestamp) | Timestamp (seconds since Epoch) when transaction ended. | 963873698.73 |
| %l (cs-uri-stem) | URI stem accessed; the URL up to the first question mark (?). | rtsp://abc/hello.rm |
| %o (c-os) | Client operating system. | Windows |
| %p (player) | Client media player version. | RealPlayer or 7.0.1.15 |
| %r (cs-uri) | Streaming URI (URL suffix) accessed. | rtsp://abc/hello.rm?there |
| %s (sc-status) (default) | The status code the server sends back to the client; reveals whether the request resulted in a successful response (codes beginning in 2), a redirection (codes beginning in 3), an error caused by the client (codes beginning in 4), or an error in the server (codes beginning in 5). See “Media Flow Controller Log Codes and Sub-Codes” on page 174 for the full list. | 401 |
| %t (localtime) | Local time, in DD/MM/YYYY:HH:MM:SS GGGG format (where GGGG is the time differential from GMT); indicates when the connection ended. | [03/Sep/2002:00:00:02 00000] |
| %u (cpu) | Client CPU type. | Pentium II |
| %v (c-osversion) | Client operating system version. | 98 |
| %x (transaction) | Transaction or reply code. | CLIENT_DATA_FROM_DISK |
| %A (action) | Action performed. | SPLIT, PROXY, CACHE, CONNECTING |
| %B (begin) (clip-start) | Time when client started receiving the stream. | 101 |
| %C (client-id) | Unique Client-side streaming identifier. | 12345678 |
| %D (dropped-packets) | Number of packets dropped. | 24 |
| %E (end) (clip-end) | Time when client stopped receiving the stream. | 52007 |
| %I | Number of bytes received in the request. | 214 |
| %K (resent-packets) | Number of packets resent to the Client. | 5 |
| %L (request_protocol) | The protocol used during connection (for example, RTSP or RTMP). | RTSP |
| %N | Namespace name. | stream |
| %O | Number of bytes transmitted in the response. | 412 |
| %P | Total number of packets delivered to the client. | 701 |
| %R (transport) | Transport protocol used during connection. | RTP_UDP, RTP_TCP, RAW_UDP, RAW_TCP |
| %S (stream-id) | Unique Server-side streaming identifier used to correlate the Streaming accesslog and Streaming details log entries pertaining to that server stream. | 123890000 |
| %T (time) | GMT time when the transaction ended. | 10:33:02 |
| %X (product) | Streaming product used to create and stream content. | WMT |

Error Log Options

The levels that may be set for the errorlog and the publishlog are shown in [Table 20](#).

Table 20 Error Log Levels

| Level Name (number) | Description |
|---------------------|---|
| Severe (1) | Failure, immediate attention required. Some Severe messages are Informational only and do not indicate an error. |
| Error (2) | Function worked incorrectly. |
| Warning (3) | Function worked, but not as expected. |
| MSG (4) | System activity. |
| DEBUG (5-7) | Progressively more detailed messages. |

Error Log Module Options

The errorlog module names and codes that may display are shown in [Table 21](#).

Table 21 Error Log Modules

| Module Name | Maps to Module or Modules |
|-----------------------|--|
| http | HTTP, HTTP Headers, Offline Origin Manager, Origin Manager |
| rtsp | RTSP, HTTP Headers, Origin Manager, Offline Origin Manager |
| media-cache | Media Manager, Buffer Manager, Caching Engine, File Manager, TFM (temporary file manager: services offline origin manager for files under 10MB stored until promoted to the first eligible disk cache), (cache) Analytics Manager, Media Promotion, GET Manager (queue consumer that loads the given object into the cache via GET and TFM PUT tasks), Disk Manager |
| network | Network, CP (connection pooling), Tunnel Delivery, TCP (transport control protocol) |
| nfs | NFS (network file system) |
| mfp-file | Media Flow Publisher File Manager |
| mfp-live | Media Flow Publisher Live Streaming manager |
| cluster | Cluster manager |
| auth-manager | EM (encryption manager), SSL (secure sockets layer) |
| virtual-player | SSP (server side player manager), VPE (video processing engine) |
| all | All modules |

Configuring Media Flow Controller Service Logs (CLI)

Most Media Flow Controller logs, allow you to enable/disable, copy (automatically), set file size (for auto-copy), set filename, set format/field options, and set syslog replication. See [“Configuring Media Flow Controller System Log” on page 198](#) for details on syslog options. Unless otherwise indicated, this procedure applies to the **accesslog**, **cachelog**, **errorlog**, **fmsaccesslog**, **fmsedgelog**, **fuselogs**, **streamlog**, and **tracelog** (“<*>log” = any service log).

Tip! You can schedule automatic uploads of completed logs with the **<*>log copy** command. A completed log is one that has reached its set **filesize** (default is 100). Only completed logs can be uploaded with **<*>log copy**; to see logs more frequently you could reduce the maximum filesize with this command **<*>log filesize 1**. In that way, after a log reaches 1 MB (rather than the default 100 MB) it uploads automatically to the set URL.

Tip! If this Media Flow Controller is going to be managed by a CMC server, set the auto-upload URL to the address of the CMC server and the filepath to **/log**.

1. Disable the log (these logs are enabled by default) and enable the log.

```
no <*>log enable
<*> log enable
```

2. Change the default log name, **<logfile>#.yyyymmdd_hour:min:sec**, numbered sequentially by creation time.

```
<*>log filename <new_name>.log
```

3. Change the log format (**accesslog** and **streamlog** only). See [“Access Log Format Options” on page 183](#), and [“Stream Log Format Options” on page 185](#) for details.

```
<*>log format <field1 field2 ...>
```

4. Optional. Change the log **level** or **module**; only for **errorlog**. The **level** determines how many messages are logged; the higher the number (7 is highest) the more messages; default is 1: only severe events are recorded. The **module** focuses the log, default is **all**. If you set **module** options, **errorlog** only records for those modules. See [“Error Log Options” on page 187](#) and [“Error Log Module Options” on page 187](#) for CLI details.

```
errorlog level 2
errorlog module http
errorlog module rtsp
errorlog module media-cache
```

5. Optional. Set automatic rotating of service logs based on log **filesize** or **time-interval**. If you set the **copy** option, when the log reaches its set **filesize** (in megabytes), or the set **time-interval**, it is copied to the specified machine via SCP (an SCP server must be installed on the target machine). First set the **copy** destination, then set the **rotate** criteria. If you do not set a **copy** destination, the rotated log is deleted.

```
<*>log copy <SCP>
<*>log rotate {filesize <integer> | time-interval <minutes>}
```

6. Alternatively, enable hourly log rotation; this is the same as setting **rotate time-interval 60**. If on-the-hour is enabled, it takes precedence over any **rotate** criteria configuration.

```
<*>log on-the-hour {disable | enable}
```

7. Optional. Merge log entries with syslog entries, making them available through the Web interface **System Logs** page.

```
<*>log syslog replicate enable
```

- Restart the log service.

```
service restart mod-log
```

- Upload the current service log.

```
upload <*>log {current | all} <scp://  
username[:<password>]@<hostname><path>
```

Example using **accesslog**:

```
MFC (config) # no accesslog enable  
MFC (config) # accesslog enable  
MFC (config) # accesslog filename sv05accesslog.log  
MFC (config) # accesslog format %h %V %u %t %s %b %N  
MFC (config) # accesslog copy scp://joe@sv01/home/joe  
Password: *****  
MFC (config) # accesslog rotate filesize 75 time-interval 1  
MFC (config) # accesslog on-the-hour enable  
MFC (config) # accesslog syslog replicate enable  
MFC (config) # service restart mod-log  
MFC (config) # upload accesslog current scp://joe@sv01/home/joe  
Password: *****  
MFC (config) #
```

To make these configurations using the Web interface, go to the **Service Config** tab, **<*>log** page. After making changes use **EZconfig** page **Service Restart** area to restart the appropriate log service. Be sure to click **Apply** at each section and **Save** at the top right of the page to make the changes persistent across reboots/restarts.

For CLI details on the service logs, see [accesslog](#), [cachelog](#), [errorlog](#), [fmsaccesslog](#), [fmsedgelog](#), [fuselog](#), [streamlog](#), and [tracelog](#).

Reading Media Flow Controller Service Logs Overview

- [Reading the Service Log \(accesslog\)](#)
- [Reading the Cache Log \(cachelog\)](#)
- [Reading the Error Log \(errorlog\)](#)
- [Reading the FMSAccess Log \(fmsaccesslog\)](#)
- [Reading the FMSEdge Log \(fmsedgelog\)](#)
- [Reading the FMSCollector Log / fuselog](#)
- [Reading the Stream Log \(streamlog\)](#)
- [Reading the Trace Log \(tracelog\)](#)



NOTE: The Web interface name for the log is given, and the command line interface (CLI) name is given in parentheses.

Reading the Service Log (accesslog)

The Access Log, referred to as the Service Log in the Web interface, records each HTTP transaction going through Media Flow Controller. This log identifies HTTP traffic. Media Flow Controller supports NCSA or Custom log formats.

See [“Configuring Media Flow Controller Service Logs \(CLI\)” on page 188](#) for implementation details including how to upload the log.

See [accesslog](#) for CLI details.

The accesslog (shown as **Service Log** in the Web interface) provides this information, in this order (by default—the order changes if you change the accesslog format). Unavailable information is indicated by a dash (-).

- (%h) Client source IP address (remote host)
- (%V) HTTP host (server name)
- (%u) Remote user
- (%t) Timestamp
- (“%r”) Request line, in quotes
- (%s) Status code (see [“Logging Status \(%s\) HTTP Codes” on page 174](#))
- (%b) Bytes out
- (“%{Referrer}i”), Referrer, in quotes. The site the client reports having been referred from.
- (“%{User-Agent}i”), Agent, in quotes. Identifying information that the client browser reports about itself.
- (%y) Status sub-code (see [“Additional Logging Status Sub-Codes” on page 175](#))
- (%c) Cache Hit Indicator
- (%A) Request in time
- (%B) First Byte Out Time
- (%F) Last Byte Out Time
- (%M) Data Out in milliseconds

Example:

```
10.15.2.211 10.15.2.14 - [15/Nov/2010:23:50:38 +0000] "GET /user/2M HTTP/1.0" 200 2097152 "-"
"Wget/1.10.2 (Red Hat modified)" 0 Origin 1289865038.272 1289865038.288 1289865038.334 46
10.15.2.211 10.15.2.14 - [15/Nov/2010:23:51:00 +0000] "GET /user/8M HTTP/1.0" 200 8388608 "-"
"Wget/1.10.2 (Red Hat modified)" 0 Origin 1289865060.345 1289865060.365 1289865060.640 275
10.15.2.211 10.15.2.14 - [15/Nov/2010:23:52:14 +0000] "GET /user/8M HTTP/1.0" 200 8388608 "-"
"Wget/1.10.2 (Red Hat modified)" 0 Buffer 1289865134.735 1289865134.735 1289865134.785 50
```

Reading the Cache Log (cachelog)

Media Flow Controller supplies a log of cache activity; for example, an object has been added, deleted, or modified in the cache.

See [“Configuring Media Flow Controller Service Logs \(CLI\)” on page 188](#) for implementation details including how to upload the log.

To upload a cache log, run **upload cachelog scp://<URL>**. Now you can access the cache log at the specified system, move it (if needed) and view its contents. See [“Terminology” on page 31](#) for the **scp** URL format and requirements.

The cache log supplies this information:

- Date and time stamp
- Event string, may include the following events and other information

For each type of event, **Add**, **Attribute Update**, and **Delete**; a single entry is written:

- ADD entries include:
 - a. Date in square brackets
 - b. Type (**ADD**)
 - c. URI name in double quotes
 - d. Cache tier name
 - e. Cache name
 - f. Content length in bytes
 - g. Expiry time for this URI, in UTC format with square brackets
- ATTR_UPDATE (Attribute Update) entries include:
 - a. Date in square brackets
 - b. Type (**ATTR_UPDATE**)
 - c. URI name in double quotes
 - d. Cache tier name
 - e. Cache name
 - f. Expiry time for this URI, in UTC format with square brackets
- DELETE entries include:
 - a. Date in square brackets
 - b. Type (**DELETE**)
 - c. URI name in double quotes
 - d. Cache tier name
 - e. Cache name

Example:

```
[Fri Jun 26 19:37:09.754 2009] ADD "/http-cl18:ed239a85/100k-files/117/29" SAS dc_3 32768 [Fri Jun 26 19:38:14 2009]
[Fri Jun 26 19:38:23.257 2009] ATTR_UPDATE "/http-cl18:ed239a85/100k-files/588/45" SAS dc_3 [Fri Jun 26 19:39:28 2009]
[Fri Jun 26 19:50:59.072 2009] DELETE "/http-cl18:ed239a85/100k-files/381/69" SAS dc_4
```



NOTE: Attribute update (ATTR_UPDATE) is performed with a **namespace origin-request cache-revalidation** on an individual URI. If this namespace option is set to **deny**, no Attribute Update entries occur. Each URI has its own expiry time. If the revalidation indicates that a URI is still valid, the Attribute Update event updates the URI expiry time; this makes a log entry.

Reading the Error Log (errorlog)

The Media Flow Controller errorlog records service-related error messages, such as cache and delivery problems; this log is for Technical Support debugging. The Media Flow Controller system log (syslog) records system-related messages and errors, such as user logins and

CPU problems; see [“Configuring Media Flow Controller System Log” on page 198](#) for more information.

See [“Configuring Media Flow Controller Service Logs \(CLI\)” on page 188](#) for implementation details including how to upload the log.

By default, the errorlog name is “error.log.” If Media Flow Controller is launched from the serial console, all error messages print out on the console; if Media Flow Controller is launched from the CLI, all error messages are logged in the file /var/log/messages.

The Error Logs provide freeform information; including these fields:

- Date and time, in brackets
- Media Flow Controller module and message level in this form: **module.level**, in brackets (see tables [Table 21, “Error Log Modules](#) and [Table 20, “Error Log Levels](#) for descriptions); module code may not display.
- Function name and source line number (ends with colon)
- Message, may include codes or sub-codes given in [Table 15, “Logging Status \(%s\) HTTP Codes](#) and [Table 16, “Additional Logging Status Sub-Codes](#).

Example:

```
[Tue Jan 5 18:27:31.285 2010][MOD_HTTPHDRS.MSG] get_nth_list_element:1697: find_nth_name_value()
failed rv=1
[Tue Jan 5 18:27:31.285 2010][MOD_HTTPHDRS.MSG] mime_hdr_get_nth_unknown:989:
get_nth_list_element() failed rv=2
[Tue Jan 5 18:27:31.285 2010][MOD_HTTPHDRS.MSG] mime_hdr_get_unknown:930:
find_name_value_by_name() failed rv=2
```

See also [“Media Flow Controller Log Codes and Sub-Codes” on page 174](#).

Reading the FMSAccess Log (fmsaccesslog)

FMSAccess Log lists all FMS server command executions. Streaming fmsaccesslog events include play, pause, seek, and stop events; session fmsaccesslog events include connect, disconnect, and connect-pending events, by default. This log is generated by the FMS server (must be installed, see [“Installing and Using FMS in Media Flow Controller \(CLI\)” on page 110](#)). This log is written to /nkn/adobe/fms/logs/**access.<nn>.log**, by default (you can access this using **application fms shell**). See **fmsaccesslog** for CLI details.

See [“Configuring Media Flow Controller Service Logs \(CLI\)” on page 188](#) for implementation details including how to upload the log.

The FMSAccess log provides this information:

- **accesslog events**—Play, pause, seek, stop, connect, disconnect, connect-pending
- **accesslog category**—Session, stream, server, vhost, application, authorization
- **date**—Date when the event occurred
- **time**—Time when the event occurred
- **x-pid**—Server process ID
- **c-ip**—Client IP address
- **cs-bytes**—The number of bytes transferred from the client to the server
- **sc-bytes**—The number of bytes transferred from the server to the client

- **x-sname**—Stream name
- **sc-stream-bytes**—The number of bytes transferred from the server to the client per stream.
- **x-file-size**—Stream size in bytes.
- **x-file-length**—Stream length in seconds
- **x-status**—Request status codes
- **x-comment**—Any comments about session
- **c-proto**—Connection protocol RTMP or RTMPT
- **s-uri**—URI of the FMS application

Example:

```

session connect 2010-05-27 09:32:13 23835 10.4.1.101 3073 3073 - - - - 200
session connect 2010-05-27 09:32:13 23835 10.4.1.101 3073 3073 - - - - 200
session connect 2010-05-27 09:32:13 23835 10.5.1.101 3073 3073 - - - - 200
session connect 2010-05-27 09:32:13 23835 10.5.1.101 3073 3377 - - - - 200
session connect 2010-05-27 09:32:13 23835 10.4.1.101 3073 3073 - - - - 200
stream play 2010-05-27 09:32:13 23835 10.5.1.101 3175 3451 2mbpsload/
2mbpsMP4_159.mp4 0 148586967 596.459000 200
stream stop 2010-05-27 10:03:27 14959 10.3.1.101 3175 59388577
2mbpsload/2mbpsMP4_100.mp4 59363080 148586967 596.459000 408
session disconnect 2010-05-27 10:03:27 14959 10.3.1.101 3175 59309712 - -
- - 200
session disconnect 2010-05-27 10:03:27 14959 10.3.1.101 3175 59309676 - -
- - 200
session disconnect 2010-05-27 10:03:27 14959 10.3.1.101 3175 59309712 - -
- - 200
stream stop 2010-05-27 10:03:27 14959 10.3.1.101 3175 57711996
2mbpsload/2mbpsMP4_111.mp4 57687110 148586967 596.459000 408 -
stream stop 2010-05-27 10:03:27 14959 10.1.1.101 3175 60830983
2mbpsload/2mbpsMP4_231.mp4 60804903 148586967 596.459000 408 -

```

See also [“Media Flow Controller Log Codes and Sub-Codes” on page 174.](#)

Reading the FMSEdge Log (fmsedgelog)

FMSEdge Log is used for diagnostic purposes and lists transactions to the FMS edge server; for example, “Connection rejected by server,” “Edge disconnected from core,” “Listener started for clients.” This log is generated by the FMS server (must be installed, see [“Installing and Using FMS in Media Flow Controller \(CLI\)” on page 110](#)). This log is written to /nkn/adobe/fms/logs/edge.<nn>.log, by default (you can access this using **application fms shell**). See [fmsedgelog](#) for CLI details.

See [“Configuring Media Flow Controller Service Logs \(CLI\)” on page 188](#) for implementation details including how to upload the log.

The FMSEdge log provides this information:

- **Date**—Date of the event
- **Time**—Time of the event
- **x- pid**—Server process ID

- **x-status**—The code is category and message ID (i) information and message ID. See [Flash Media Server Diagnostic Log Messages](#) for descriptions of all message ID.
- **x-ctx**—Event dependent context information

Example:

```

2010-05-27 09:57:23 13826 (i)2581173 Host: QA10 IPv4: 127.0.0.1 -
2010-05-27 09:57:23 13826 (i)2631174 Listener started ( _defaultRoot__edge1 ) :
localhost:19350/v4 -
2010-05-27 09:57:23 13826 (i)2581252 Registering core (13829). -
2010-05-27 09:57:24 13826 (i)2631174 Listener started ( _defaultRoot__edge1 ) :
1935/v4 -
2010-05-27 09:57:24 13826 (i)2631174 Listener started ( _defaultRoot__edge1 ) :
8888/v4 -
2010-05-27 09:58:29 13826 (i)2581252 Registering core (14959). -
2010-05-27 10:28:34 13826 (i)2581250 Edge disconnected from core (14959).

```

Reading the FMSCConnector Log / fuselog

The Fuse Log, referred to as the FMSCConnector Log in the Web interface, records RTMP transaction details including what URIs are accessed and how many bytes are returned by the FUSE module. This log is generated by Media Flow Controller. See [fuselog](#) for CLI details.

See [“Configuring Media Flow Controller Service Logs \(CLI\)” on page 188](#) for implementation details including how to upload the log.

The fuse log supplies this information:

- Date and time stamp
- Event string, includes event type and information

Example:

```

[Thu Sep 23 22:44:22.349 2010] Opened uri : /fms/bunny_avc_aac_hinted.mp4
[Thu Sep 23 22:44:22.427 2010] Opened uri : /fms/bunny_avc_aac_hinted.mp4
[Thu Sep 23 22:44:22.429 2010] Opened uri : /fms/bunny_avc_aac_hinted.mp4
[Thu Sep 23 22:44:22.432 2010] Opened uri : /fms/bunny_avc_aac_hinted.mp4
[Thu Sep 23 22:44:22.625 2010] Opened uri : /fms/bunny_avc_aac_hinted.mp4

```

Reading the Stream Log (streamlog)

This log records RTSP streaming transactions. See [streamlog](#) for CLI details.

See [“Configuring Media Flow Controller Service Logs \(CLI\)” on page 188](#) for implementation details including how to upload the log.

The stream log provides the information you configure with **streamlog format** **<format_options>** (see [“Stream Log Format Options” on page 185](#)); by default:

- **%h**—Client source IP address
- **%c**—Destination IP address
- **%t**—Local time
- **%x**—Transaction code. Possible entries are:
 - OPTIONS
 - DESCRIBE

- SETUP
- PLAY
- PAUSE
- GET_PARAMETER
- TEARDOWN
- BROADCAST_PAUSE
- ANNOUNCE
- RECORD
- REDIRECT
- SET_PARAMETERS
- DESCRIBE_RESP
- SETUP_RESP
- PLAY_RESP
- TEARDOWN_RESP
- UNKNOWN
- %r—Streaming URI (URL suffix)
- %s—Status
- %I—Number of bytes received
- %O—Number of bytes transmitted

Example:

```
10.157.42.147 10.157.42.145 [23/Sep/2010:17:39:12 +0000] OPTIONS "rtsp://10.157.42.145/rtsp/sanity_video/sample_h264_1mbit.mp4" 415 139 136
10.157.42.147 10.157.42.145 [23/Sep/2010:17:39:47 +0000] OPTIONS "rtsp://10.157.42.145/rtsp/sanity_video/sample_h264_1mbit.mp4" 200 139 169
10.157.42.147 10.157.42.145 [23/Sep/2010:17:39:47 +0000] DESCRIBE "rtsp://10.157.42.145/rtsp/sanity_video/sample_h264_1mbit.mp4" 200 165 862
10.157.42.147 10.157.42.145 [23/Sep/2010:17:39:47 +0000] SETUP "rtsp://10.157.42.145/rtsp/sanity_video/sample_h264_1mbit.mp4/trackID=3" 200 203 275
10.157.42.147 10.157.42.145 [23/Sep/2010:17:39:47 +0000] SETUP "rtsp://10.157.42.145/rtsp/sanity_video/sample_h264_1mbit.mp4/trackID=4" 200 233 275
10.157.42.147 10.157.42.145 [23/Sep/2010:17:39:47 +0000] PLAY "rtsp://10.157.42.145/rtsp/sanity_video/sample_h264_1mbit.mp4" 200 213 391
10.157.42.147 10.157.42.145 [23/Sep/2010:17:40:57 +0000] TEARDOWN "rtsp://10.157.42.145/rtsp/sanity_video/sample_h264_1mbit.mp4" 200 170 138
10.157.42.147 10.157.42.145 [23/Sep/2010:18:34:18 +0000] OPTIONS "rtsp://10.157.42.145/rtsp/sanity_video/sample_h264_1mbit.mp4" 200 139 169
```

Reading the Trace Log (tracelog)

Media Flow Controller includes a delivery trace facility to help diagnose the handling of a particular HTTP request. See [tracelog](#) for CLI details.

The tracing is done by logging trace points of internal steps (for example, namespace lookup, cache lookup, and so forth). A request is traced if the following conditions are true:

- The global trace flag is enabled via the CLI:

```
delivery protocol http trace enable
```
- The HTTP request includes the **X-NKN-Trace** header; for example if using Wget:

```
wget --header "X-NKN-Trace:" <URL>
```

The HTTP modules detect the above conditions and set the flag "HRF_TRACE_REQUEST" as well as other flags that direct each relevant module to log meaningful trace points. The trace points (as of Release 2.0.7) given are shown with variables that are instantiated with data before displaying:

OM=Origin Manager, **%s**=string data, **%d**=integer data, **%ld**=long integer data.

The trace points described in [Table 22](#), are currently (as of Release 2.0.7) defined in Media Flow Controller.

Table 22 Delivery Protocol HTTP Trace Points

| Trace Point | Description |
|---|---|
| "Object %s bytes %ld to %ld served from %s" | NFS trace. |
| "Unmount problem: Stale NFS handle %s" | NFS; the given NFS mount point is not valid |
| "Mount failed for mount command: %s" | NFS mount failed. |
| "Could not unmount old mount configuration. Please check whether someone is accessing this directory: %s" | NFS mount failed. |
| "Namespace configuration does not exist" | The requested namespace parameter is undefined |
| "URI prefix does not exist in namespace configuration" | The requested uri-prefix is not configured in the requested namespace. |
| "HTTP config does not exist in namespace configuration" | The requested namespace does not have HTTP delivery protocol or origin-server configured. |
| "Received TRACE URI: %s" | NFS trace. |
| "Programming error. Could not get URI " "postfix | NFS trace. |
| "Could not find the file in the " "directory. Please check filename: %s errno = %d" | The requested file was not found in the specified directory. |
| "Could not mount file. Please check " "filename: %s. errno = %d" | NFS; the requested file could not be mounted. |
| "End TRACE URI: %s. End NFS_stat. Success." | NFS. |
| "TRACE URI: %s. End NFS stat Error: URI not found" | NFS. |
| "End TRACE URI: %s. Error while doing STAT" | NFS. |
| "NFS GET: Received TRACE URI: %s" | NFS. Received trace URI |
| "NFS GET: Programming error: NFS object not found in get after stat succeeded" | NFS. Object not found |
| "NFS GET: Programming error: Error getting object after stat finished. " | NFS. Error getting object after stat finished. |
| "NFS GET: Cannot read content from file." | NFS. Requested content not in file. |

Table 22 Delivery Protocol HTTP Trace Points (Continued)

| Trace Point | Description |
|---|--|
| "NFS GET: Cannot get attributes for this file" | NFS. Requested file attributes missing. |
| "NFS GET: TRACE URI: %s. End NFS get: Success" | NFS. Get successful. |
| "NFS GET: TRACE URI: %s. End NFS get: Error: Server busy" | NFS. Server busy. |
| "NFS GET: TRACE URI: %s. End NFS get: Error getting uri" | NFS. Error getting URI. |
| "OM returning %s bytes for object %s offset=%ld length=%ld" | Origin Manager. Returning cacheable/non-cacheable given bytes to the buffer manager for the given object starting at the given offset and length. |
| "OM connect failed %s:%hu for object %s" | Origin Manager. Unable to connect to the origin server identified by the given hostname/IP address and port for the given object. |
| "OM http_response object %s\n%s" | Origin Manager. HTTP origin server response for given object and headers. |
| "OM connect failed %s:%hu for object %s" | Origin Manager. Socket connect to the given IP address and port failed for the given object. |
| "OM gethostbyname_r(%s) failed" | Origin Manager. gethostbyname_r call failed for the given hostname/IP address with the given error code. |
| "OM connect (%s) failed" | Origin Manager. Socket connect to given hostname/IP address failed with the given error code. |
| "OM connecting %s:%hu for object %s" | Origin Manager. Socket connect to the given IP address and port is pending for the given object. |
| "Hash authentication failed for virtual player: %s" | Server Side Player. Video will not be delivered. |
| "Hash authentication successful for virtual player: %s" | Server Side Player. Video will be delivered. |
| "Fast start buffer size set to %d bytes" | Server Side Player. Configured fast--start value. |
| "Fast start not enabled" | Server Side Player. The fast-start option was requested but not enforced in the virtual player. |
| "AFR Set to %d Bytes/sec" | Server Side Player. assured-flow rate was set to the given value for delivering this video. |
| "AFR Failed. %d b/w unavailable for profile %s" | Server Side Player. The requested assured-flow rate was unable to be met due to a lack of bandwidth for a given video/profile; the video was not delivered since Media Flow Controller cannot honor the configured assured-flow rate . |
| "Virtual Player: %s of Type-%d has been invoked" | Virtual Player. Virtual player of the named type was invoked. |
| "No Virtual Player associated with this namespace" | Virtual Player. The requested virtual player option was unavailable because no virtual player is assigned to the requested namespace. |
| "Health probe request enabled and no cache flag is set" | Virtual Player. For this request, the fetched file will not be cached in Media Flow Controller. |

Table 22 Delivery Protocol HTTP Trace Points (Continued)

| Trace Point | Description |
|--|---|
| "URL %s%s matched namespace %s" | Server Side Player. The incoming request was mapped to a configured namespace. |
| "URL %s%s did not match any namespace" | Server Side Player. The incoming request could not be mapped to any configured namespace. |
| "HTTP Response object %s\n%s" | HTTP Response object given. |
| "AFR Failed, b/w unavailable" | Assured-flow rate could not be met due to lack of available bandwidth. |

To upload a trace log, run **upload tracelog scp://<URL>**. Now you can access the trace log at the specified system, move it (if needed) and view its contents. See ["Terminology" on page 31](#) for the **scp** URL format and requirements.

Configuring Media Flow Controller System Log

The system log (syslog) that records all system activity such as user logins, configuration changes, and system condition changes. It does not record service activity or errors. The Media Flow Controller errorlog records service related errors but is mostly useful for debugging by Juniper Networks Support. Media Flow Controller provides several service-specific logs, detailed in [Chapter 10, "Troubleshooting Media Flow Controller."](#)

Specify, on a global or local level, what system data is collected when; when log files are deleted, or rotated; where log files are uploaded to; the format for logs; what traps are sent to syslog; and whether or not to accept log messages from remote servers.

System Log Severity Levels and Classes

The logging commands provide pre-defined log severity levels so you can refine what messages are logged, and pre-defined log classes that divide messages up according to their origin. Logging **severity-level** options are:

- **emerg**—System is unusable (requires immediate action)
- **alert**—System may become unusable (requires immediate action)
- **crit**—Critical conditions (requires immediate action)
- **err**—Error conditions (minor issue; for example, "disk went bad")
- **warning**—Warning conditions (functionality OK, but sub-optimal)
- **notice**—Normal but significant conditions (default)
- **info**—Informational messages (administrator actions)
- **debug**—Debug-level messages (all messages)

Logging **class** options are:

- **mgmt-core**—Management daemon (mgmtd) only
- **mgmt-back**—Other back end components
- **mgmt-front**—Front end components, utilities, and tests

Configuring Media Flow Controller System Logging (CLI)

See **logging** for CLI details. To configure the Media Flow Controller syslog:

1. Set the minimum severity of log messages to be saved in log files on local persistent storage (regardless of source), the severity level at which user-executed CLI commands are logged, or disable local logging altogether.


```
logging local <severity_level>
logging level cli commands <severity_level>
logging local none
```
2. Set or remove (with **no**) a per-class override on the global logging level for logged messages local or to a specified remote syslog server; all classes without an override use the global logging level set with **logging local <severity_level>** (default is notice). Use the **no** variant or set **none** as the severity level to disable logging from that class entirely.


```
logging local override class <class> priority <severity_level>
logging <IP_address> trap override class <class> priority <severity_level>
```
3. Set or remove (with **no**) a remote syslog server to receive log messages; and a severity level of logged messages sent to all, or a specified, remote syslog server.


```
logging <IP_address>
logging trap <severity_level>
logging <IP_address> trap <severity_level>
```
4. Stop sending log messages to any, or a specified, remote syslog server.


```
logging trap none
logging <IP_address> trap none
```
5. Allow (or disable with **no**) this system to receive log messages from another host; disabled by default. If enabled, only log messages matching or exceeding the minimum severity specified with **logging local <log level>** are logged, regardless of what is sent.


```
logging receive
```
6. Set (or reset default with **no**) the format in which log messages should be set. The default is **standard**.


```
logging format standard
logging format welf
```
7. Set (or remove with **no**) the firewall name that should be associated with each message logged in WELF (Web trends Enhanced Log Format). If no firewall name is set, the hostname is used by default. Neither of these commands enables WELF logging if it is not already enabled with **logging format welf**.


```
logging format welf fw-name <firewall_name>
```
8. Include (or disable with **no**) an additional field in each log message showing the number of seconds since the Epoch; default is **disabled**. This is independent of the standard syslog datetime at the beginning of each message in the format "Feb 25 18:00:00". Aside from indicating the year at full precision, its main purpose is to provide subsecond precision. The precision can be controlled with the two **digits** commands which control the number of digits to the right (**fractional**) and left (**whole**) of the decimal point; **all** = no limit. Except for the year, all of these digits are redundant with syslog's own datetime.


```
logging fields seconds enable
logging fields seconds whole-digits {1 | 6 | all}
logging fields seconds fractional-digits {1 | 2 | 3 | 6}
```

9. Configure when log files on local persistent storage should be automatically rotated. Choose one of two mutually exclusive options: rotation based on time, or active log file size. Default is time = **daily** (once per day at midnight).


```
logging files rotation criteria frequency {daily | weekly | monthly}
logging files rotation criteria size <log_file_size_threshold>
logging files rotation criteria size-pct
    <log_file_size_percent_threshold>
```
10. Configure how many old log files are kept. When the number of log files exceeds this number (either at rotation time, or when this setting is lowered), the system deletes as many as necessary to bring it down to this number, starting with the oldest.


```
logging files rotation max-num <maximum_number_of_files_to_keep>
```
11. Force an immediate rotation of the all log files.


```
logging files rotation force
```
12. Delete a specified number of the oldest log files.


```
logging files delete oldest [<number of files to delete>]
```
13. Upload a log file to a remote host. The word **current** specifies the current log file. To specify an archived log file, give its number instead, as displayed by **show log files**.


```
logging files upload {current | <file number>} <URL>
```
14. View a local log file. If **<file_number>** is specified, view an archived log file, where the number is from 1 up to the number of archived log files (10 is the default maximum allowed); the higher the number, the more recent the log file. If **[not] matching <regex>** is specified, the file is filtered to only include lines either matching, or not matching, the provided regular expression. Enclose all **regex** entries in double quotes.


```
show log [files <file_number>] [[not] matching <regex>]
```
15. Display the last few lines of the current log file, and then continue to display new lines as they come in, until you press Ctrl+C. If **[not] matching <regex>** is specified, only log lines matching, or not matching, the provided regular expression are printed. Enclose all **regex** entries in double quotes.


```
show log continuous [[not] matching <regex>]
```
16. Display logging configuration settings or a list of local log files.


```
show logging
show log files
```

Example:

```
MFC (config) # logging local notice
MFC (config) # logging level cli commands notice
MFC (config) # logging 123.54.10.12
MFC (config) # logging 123.54.10.12 trap emerg
MFC (config) # logging local override class mgmt-front priority info
MFC (config) # logging 123.54.10.12 trap override class mgmt-front
    priority info
MFC (config) # logging 123.54.10.12 trap none
MFC (config) # logging receive
MFC (config) # logging format standard
MFC (config) # logging fields seconds enable
MFC (config) # logging files rotation criteria frequency daily
MFC (config) # logging files rotation max-num 12
MFC (config) # logging files rotation force
MFC (config) # logging files delete oldest 5
```



```
MFC (config) # logging files upload current scp://joe@sv01/home/joe
Password: *****
MFC (config) #
```

Reading the Media Flow Controller System Log

Media Flow Controller employs a syslog utility for tracking and logging all manner of system messages from the merely informational to the extremely critical; this log is not specific to service activity.

Set counter thresholds with **stats** commands; set e-mail notifications with **email** commands. See [“System Log Severity Levels and Classes” on page 198](#), for severity level options.

See [“Configuring Media Flow Controller System Logging \(CLI\)” on page 199](#) for implementation details including how to upload the log.

The System Log keeps track of all system activities, similar to syslog on a UNIX system. The System Log provides this information, in this order:

- Date and time
- Media Flow Controller hostname and process name (ends with colon)
- Process date and time, in brackets (may be missing)
- Event information, may include severity level in brackets, see [“System Log Severity Levels and Classes” on page 198](#), for levels

Example:

```
Jan  6 00:10:00 MFC httpd: [Wed Jan 06 00:10:00 2010] [notice] Apache configured -- resuming normal
operations
Jan  6 05:58:50 MFC pm[4617]: [pm.NOTICE]: Output from nknlogd (Nokeena Log Manager): Debuglog
socket 17 closed
Jan  6 08:58:27 MFC login: ROOT LOGIN ON ttyS0: user admin (System Administrator)
```

Reading the Media Flow Controller Tech-Support Log

The **tech-support** command outputs a super-set of syslog information in the form of a compressed (.tgz) file. There is no configuration. The contents of the **nkn_tech-support.tgz** file include the following:

```
active.db
active.txt
build_version.sh
build_version.txt
config
cpuinfo.log
dmesg
iomem.log
ioports.log
list-var_opt_tms.txt
lsof.txt
lspci-vvv.log
lspci-vvvn.log
messages
```

```

messages.10.gz
messages.1.gz
messages.2.gz
messages.3.gz
messages.4.gz
messages.5.gz
messages.6.gz
messages.7.gz
messages.8.gz
messages.9.gz
mfdb
mfdb.txt
modprobe.d
nkn cnt.log.1
nkn cnt.log.2
output
root_cli_history
running-config.txt
scsi.log
sysinfo.txt
systemlog
version.log

```

To view the Media Flow Controller tech-support output, run **tech-support scp://<URL>**. Now you can access the tech-support output at the specified system, move it (if needed) and view its contents. See [“Terminology” on page 31](#) for the **scp** URL format and requirements. The file is created with the name **nkn_tech-support**. We highly recommend that, if you have multiple Media Flow Controllers, each Media Flow Controller be configured with a different path variable (use a different directory) so that **tech-support** data from one Media Flow Controller does not overwrite data from another; if different hosts are used for each Media Flow Controller, then the same path name can be used.

Configuring Media Flow Controller Log Statistics Thresholds (CLI)

Use the **stats** commands to set thresholds for syslog events, and export parameters for which statistics should be exported when. See [stats](#) for CLI details.

1. Enable or disable (with **no**) the specified alarm. Three alarms are disabled by default: **disk_io** (Disk I/O per second too high), **intf_util** (network utilization too high), and **memory_pct_used** (too much memory in use); all other alarms are enabled by default.


```
stats alarm <alarm_ID> enable
```

Example:

```
MFC (config) # no stats alarm total_byte_rate enable
MFC (config) # stats alarm total_byte_rate enable
```
2. Change the thresholds that initiate or terminate (clear) the specified alarm. The units for the **cpu_util_indiv** alarm are hundredths of a point of the one-minute load average. For example, setting it to **100** causes an alarm if the one-minute load average is ever over 1.0 when it is sampled. The units for the **paging** alarm are number of pages read from or written to the swap partition that has occurred over the past 20 seconds. Type **show stats alarm** to view alarm IDs and current status; to see default threshold values, enter **show**

stats alarm <alarm_ID>. Alarm threshold defaults have been adjusted for Media Flow Controller.

```
stats alarm <alarm_ID> rising error-threshold <threshold>
stats alarm <alarm_ID> rising clear-threshold <threshold>
stats alarm <alarm_ID> falling error-threshold <threshold>
stats alarm <alarm_ID> falling clear-threshold <threshold>
```

3. Set or reset the alarm event rate-limit maximum counts for the three types of counts (**short, medium, long**) for alarms; defaults are short= 5, medium=20, long= 50.

```
stats alarm <alarm_ID> rate-limit count long {50 | <duration>}
stats alarm <alarm_ID> rate-limit count medium {20 | <duration>}
stats alarm <alarm_ID> rate-limit count short {5 | <duration>}
stats alarm <alarm_ID> rate-limit reset
```

4. Configure or reset alarm event rate-limit duration windows for the three types of durations for alarms; defaults are short=3600, medium=86400, long=604800.

```
stats alarm <alarm_ID> rate-limit window long {604800 | <duration>}
stats alarm <alarm_ID> rate-limit window medium {86400 | <duration>}
stats alarm <alarm_ID> rate-limit window short {3600 | <duration>}
```

5. Change the amount of time between samples for the specified group of sample data. Use **show stats sample** to view sample IDs and current sampling interval.

```
stats sample <sample_ID> interval <poll_interval_time_in_seconds>
```

6. Clear all memory of the specified sample, CHD (computed historical datapoint), or alarm, or of all of those together. Clearing a sample or CHD deletes all of the gathered data. Clearing an alarm resets it to a non-error state, clears the watermarks, and forgets the event history. Type **show stats chd** to view CHD IDs, status and default threshold values.

```
stats clear-all
stats sample <sample_ID> clear
stats chd <CHD_ID> clear
stats alarm <alarm_ID> clear
```

7. Export statistics to **csv** (comma-separated value) file; the only format supported for Release 2.0.7. The dataset to be exported is determined by the specified report name. If a filename is specified, the stats are exported to a file of that name; otherwise a name is given that contains the report name and time and date of the export. Either one, both, or neither of the **after** and **before** parameters may be specified, placing boundaries on the timestamps of the instances to be exported; they may come in either order relative to each other. Use **show files stats** to see the file. Use **file stats <filename> upload <URL>** to send the file to another system. See also [“Stats Reports Names Options” on page 204](#).

```
stats export csv <report_name> [filename <filename>] [after <date> <time>]
[before <date> <time>]
```

Example:

```
MFC (config) # stats export csv cpu_util
MFC (config) # show files stats
cpu_util-20090401-161206.csv
MFC (config) # file stats upload cpu_util-20090401-161206.csv scp://
joe@sv01/home/joe
Password: *****
MFC (config) #
```

8. List all stats of the specified type or a particular stat. If **cpu** is specified, basic statistics about CPU utilization are displayed.

```
show stats {alarm [<alarm_ID>] | chd [<CHD_ID>] | cpu | sample [<sample_ID>]}
```



NOTE: You cannot set log statistics thresholds using the Web interface in Media Flow Controller Release 2.0.7.

Stats Reports Names Options

These are the report names you can use to export stats; the reports list all related alarms, chds, and samples. See [stats](#) for CLI details.

- **memory**—Memory utilization
- **paging**—Paging I/O (input/output)
- **cpu_util**—CPU utilization
- **bandwidth_day_avg**—Average bandwidth usage
- **bandwidth_day_peak**—Peak bandwidth usage
- **connection_day_avg**—Average connection count
- **connection_day_peak**—Peak connection count

Measurement Counters (stats samples)

These **stats samples** collect information used by stats alarms. The options for stats **sample ID** are shown in [Table 23](#); sampling interval defaults shown in **bold**. For CLI details, see [stats](#) in the [Chapter 14, “Media Flow Controller CLI Command Reference.”](#)

Table 23 Stats Samples

| Stat Sample | Description |
|-------------------------------------|--|
| <code>cache_byte_count</code> | Bandwidth being served from RAM/buffer cache; default = 10 second . |
| <code>cache_byte_count_day</code> | Bandwidth served from RAM/buffer cache for last 24 hours; default = 5 minutes . |
| <code>cache_byte_count_week</code> | Bandwidth served from RAM/buffer cache for last 7 days; default = 30 minutes . |
| <code>connection_count</code> | Total active connections; default = 10 seconds . |
| <code>cpu_util</code> | CPU utilization: milliseconds of time spent; default = 15 seconds . |
| <code>disk_byte_count</code> | Bandwidth being served from Disk; default = 10 seconds . |
| <code>disk_byte_count_day</code> | Bandwidth being served from Disk for the last 24 hours; default = 5 minutes . |
| <code>disk_byte_count_week</code> | Bandwidth being served from Disk for the last 7 days; default = 30 minutes . |
| <code>disk_io</code> | Disk I/O (input/output, in kilobytes); default = 15 seconds . |
| <code>fs_mnt_bytes</code> | Filesystem usage, in bytes; default = 1 minute . |
| <code>fs_mnt_inodes</code> | Filesystem usage, in inodes; default = 1 minute . |
| <code>http_transaction_count</code> | Number of HTTP transactions (GET requests) per second; default = 10 seconds . |
| <code>interface</code> | Network interface statistics; default = 30 seconds . |

Table 23 Stats Samples (Continued)

| Stat Sample | Description |
|--------------------------------------|--|
| <code>intf_day</code> | Network interface statistics for the last 24 hours; default = 5 minutes . |
| <code>intf_month</code> | Network interface statistics for the last 30 days; default = 4 hours . |
| <code>intf_util</code> | Network interface utilization, in bytes; default = 5 seconds . |
| <code>intf_week</code> | Network interface statistics for the last 7 days; default = 30 minutes . |
| <code>memory</code> | System memory utilization, in bytes; default = 20 seconds . |
| <code>num_of_connections</code> | Current number of HTTP connections; default = 10 seconds . |
| <code>origin_byte_count</code> | Bandwidth being served from Origin; default = 10 seconds . |
| <code>origin_byte_count_day</code> | Bandwidth served from Origin for the last 24 hours; default = 5 minutes . |
| <code>origin_byte_count_week</code> | Bandwidth served from Origin for the last 7 days; default = 30 minutes . |
| <code>paging</code> | Paging activity: page faults; default = 20 seconds . |
| <code>perdiskbytes</code> | Number of bytes being served from each disk drive; default = 10 seconds . |
| <code>peroriginbytes</code> | Number of bytes being served from origin; default = 10 seconds . |
| <code>perportbytes</code> | By-port I/O (input/output), in kilobytes per second; default = 10 seconds . |
| <code>proc_mem</code> | Memory used by Media Flow Controller processes; default = 30 seconds . |
| <code>total_bytes</code> | Total data bandwidth being served in the system; default = 10 seconds . |
| <code>total_bytes_day</code> | Total data bandwidth served in the last 24 hours; default = 5 minutes . |
| <code>total_bytes_week</code> | Total data bandwidth served in the last 7 days; default = 30 minutes . |
| <code>total_cache_byte_count</code> | Total data bandwidth being served from cache; default = 10 seconds . |
| <code>total_disk_byte_count</code> | Total data bandwidth being served from disk; default = 10 seconds . |
| <code>total_origin_byte_count</code> | Total data bandwidth being served from origin; default = 10 seconds . |

Configuring Media Flow Controller Stats Alarms

This section describes the **stats** alarms you can set. These may reach you through an [email event name Options](#) (configurable).

Alarms can be in one of two states:

- **OK**—The alarm is in a normal state.
- **ERROR**—The alarm is already triggered and it is in the error state.

You can specify the **error-threshold** and **clear-threshold** levels for alarms using the **stats** commands; for example:

```
stats alarm total_byte_rate rising error-threshold 10
stats alarm total_byte_rate rising clear-threshold 1
```

In this example, after the **total_byte_rate** stat alarm goes beyond **10**, the alarm state changes to ERROR. The state changes to OK only when the **total_byte_rate** stat alarm comes to less than or equal to **1**.

Use the **stats** commands to set “error” and “clear” thresholds for **alarms**. [Table 24](#) shows Media Flow Controller **stats alarms**; all defaults are configurable. For details on **stats chds** (computed historic datapoints) and **stats samples**, see **stats** in the [Chapter 14, “Media Flow Controller CLI Command Reference.”](#)

For configuration details, see [“Configuring Media Flow Controller Log Statistics Thresholds \(CLI\)” on page 202.](#)

1. View current alarm, chd, and sample defaults.

```
show stats [alarm | chd | sample]
```
2. View alarm thresholds, chd range and interval values, and sampling interval defaults.

```
show stats [alarm <alarm_ID>] [chd <chd_ID>] [sample <sample_ID>]
```
3. Enable a stat alarm that is disabled by default.

```
stats alarm <alarm_ID> enable
```
4. Set rising and falling error-thresholds and clear-thresholds, as appropriate. Some alarms operate on a “rising” basis: that is, the alarm is triggered when something *rises above* the error-threshold and is cleared when it *falls back* to the clear-threshold. Other alarms operate on a “falling” basis: the alarm is triggered when something *falls below* the error-threshold and is cleared when it *rises back* to the clear-threshold.

```
stats alarm <alarm_ID> rising error-threshold <threshold>
stats alarm <alarm_ID> falling clear-threshold <threshold>
stats alarm <alarm_ID> falling error-threshold <threshold>
stats alarm <alarm_ID> rising clear-threshold <threshold>
```
5. Export statistics to **csv** (comma-separated value) file; the only format supported for Release 2.0.7. The dataset to be exported is determined by the specified report name. If a filename is specified, the stats are exported to a file of that name; otherwise a name is given that contains the report name and time and date of the export. Either one, both, or neither of the **after** and **before** parameters may be specified, placing boundaries on the timestamps of the instances to be exported; they may come in either order relative to each other. Use **show files stats** to see the file. Use **file stats <filename> upload <URL>** to another system.

```
stats export csv <report_name> [filename <filename>] [after <date> <time>]
[before <date> <time>]
```

Example:

```
MFC (config) # stats export csv cpu_util
MFC (config) # show files stats
cpu_util-20090401-161206.csv
MFC (config) # file stats upload cpu_util-20090401-161206.csv scp://
joe@sv01/home/joe
Password: *****
MFC (config) #
```

Table 24 Media Flow Controller Stats Alarms

| Statistic | Description |
|--|---|
| (unless otherwise noted, default rising error threshold is 200000000 Bps; rising clear threshold is 100000000 Bps) | |
| Cache Usage Alarms | |
| <code>cache_byte_rate *</code> | Total data bandwidth being served from RAM/buffer cache. |
| <code>avg_cache_byte_rate *</code> | Total number of bytes served divided by system up time. |
| Origin Usage Alarms | |
| <code>origin_byte_rate *</code> | Current total data bandwidth being served from origin. |
| <code>avg_origin_byte_rate *</code> | Total number of bytes fetched from origin divided by system up time. |
| <code>peroriginbyte_rate</code> | Total amount of data fetched per origin server divided by system up time; cumulative (not per-origin). |
| Disk Usage Alarms | |
| <code>disk_byte_rate *</code> | Current total data bandwidth being served from disk in the system. |
| <code>avg_disk_byte_rate</code> | Total number of bytes served from all disks divided by system up time. |
| <code>perdiskbyte_rate</code> | Total number of bytes served from each disk drive divided by system up time on a per-disk basis; cumulative (not per-disk). |
| <code>disk_io *</code> | Disk I/O (input/output) in kilobytes per second.* Default rising error threshold is 5120 kilobytes per sec; default rising clear threshold is 4608 kilobytes per sec. |
| CPU Usage Alarms | |
| <code>cpu_util_indiv</code> | Average CPU utilization. Units for this alarm are hundredths of a point of the one-minute load average. For example, setting it to 100 causes an alarm if the one-minute load average is ever over 1.0 when it is sampled. Default rising error threshold is 90% ; rising clear threshold is 70% . |
| <code>nkn_cpu_util_ave</code> | CPU utilization across all cores too high |
| <code>paging</code> | Paging activity (page faults). Units for this alarm are number of pages read from, or written to, the swap partition. The alarm is on the amount of paging activity that has occurred over the past 20 seconds. Default rising error threshold is 2000 page faults; rising clear threshold is 1000 page faults. |
| <code>total_byte_rate</code> | Total data bandwidth being served in the system; does not include management traffic on Ethernet port. |
| <code>fs_mnt</code> | Percent free filesystem space. Default falling error threshold is 7% of disk space free; falling clear threshold is 10% of disk space free. |
| <code>memory_pct_used *</code> | Percent of physical memory in current in use.* Default rising error threshold is 90% of physical memory used; rising clear threshold is 87% . |
| Network Interface (Port) Usage Alarms | |
| <code>intf_util *</code> | Network utilization (in B/ps).* Default rising error threshold is 10485760 bytes per sec; rising clear threshold is 9437184 bytes per sec. |
| * Disabled by default; all others are enabled by default | |

Table 24 Media Flow Controller Stats Alarms (Continued)

| Statistic | Description |
|--|--|
| <code>perportbyte_rate *</code> | By-port I/O (input/output, in kilobytes) per second. |
| <code>connection_rate *</code> | Incoming connections per second, arrived by summing up all accepted connections and dividing by system up time. Default rising error threshold is 20000 per sec; rising clear threshold is 10000 per sec. |
| <code>http_transaction_rate *</code> | Number of HTTP transactions (GET requests) per second; (number of GET requests received so far divided by system up time). Default rising error threshold is 40000 per sec; rising clear threshold is 20000 per sec. |
| * Disabled by default; all others are enabled by default | |

Configuring Media Flow Controller Fault Notifications (CLI)

Configure e-mail notifications so key people receive notice of system errors and changes.

The **email** commands allow you to set e-mail addresses for fault notifications of specified, pre-defined, events. You can also enable autosupport e-mails: when certain failures occur Juniper Networks support automatically receives an e-mail. See [email](#) for CLI details including [“email event name Options” on page 384](#) and [“email class Options” on page 385](#).

To configure fault notifications with the CLI:

- Configure who should receive e-mail event notifications. The **class** option lets you simply choose a class of events for the specified recipients. The **event** option lets you add specified events to the **info** event class. The **detail** option only applies to process crash events; by default it is enabled.


```
email notify recipient <email_address> [class {[failure] [info]}] [detail]
email notify event <event_name>
```
- Set **domain** (default is configured **ip domain-list**), **mailhub** (SMTP server), **mailhub-port**, return address, and include or exclude (with **no**) the return host in e-mail notifications (default is include). The **mailhub** option must be sent for notifications to work.


```
email domain {<hostname> | <IP_address>}
email mailhub {<hostname> | <IP_address>}
email mailhub-port <port_number>
email return-addr <username>
email return-host
```
- Disable or enable event e-mails sent to Juniper Networks for certain pre-configured events; default is enabled. Use **no email autosupport enable** to disable.


```
email autosupport enable
```
- Manage undeliverable e-mails. Use **cleanup max-age** to set when to permanently delete.


```
dead-letter enable [cleanup max-age <duration>]
```
- Send a test e-mail to all the configured e-mail notify recipients.


```
email send-test
```
- To verify configurations:


```
show email
```

Example:

```
MFC (config) # email notify recipient bobo@example.com
```



```
MFC (config) # email notify event disk-io-high
MFC (config) # email domain example.com
MFC (config) # email mailhub mailgate1.example.com
MFC (config) # email mailhub-port 22
MFC (config) # email return-addr support@example.com
MFC (config) # email return-host
MFC (config) # email autosupport enable
MFC (config) # email dead-letter enable
MFC (config) # show email

Mail hub:          mailgate1.example.com
Mail hub port:    22
Domain:          example.com
Return address:  support@example.com
Include hostname in return address: yes

Dead Letter settings:
  Save dead.letter files: yes
  Dead letter max-age: (none)

Failure events for which emails will be sent:
  process-crash: A process in the system has crashed
  smart-warning: Smartd warnings
  unexpected-shutdown: Unexpected system shutdown

Informational events for which emails will be sent:
  liveness-failure: A process in the system was detected as hung
  process-exit: A process in the system unexpectedly exited
  cpu-util-ok: CPU utilization has fallen back to normal levels
  cpu-util-high: CPU utilization has risen too high
  disk-io-ok: Disk I/O per second has fallen back to acceptable levels
  disk-io-high: Disk I/O per second has risen too high
  disk-space-ok: Filesystem free space is back in the normal range
  disk-space-low: Filesystem free space has fallen too low
  netusage-ok: Network utilization has fallen back to acceptable levels
  netusage-high: Network utilization has risen too high
  memusage-ok: Memory usage has fallen back to acceptable levels
  memusage-high: Memory usage has risen too high
  cpu-util-ave-ok: Average CPU utilization has fallen back to normal levels
  cpu-util-ave-high: Average CPU utilization has risen too high
  paging-ok: Paging activity has fallen back to normal levels
  paging-high: Paging activity has risen too high

Email notification recipients:
  bobo@example.com (all events, in detail)

Autosupport emails
  Enabled: yes
  Recipient:
    support@example.com
  Mail hub:
    mail.example.com
  Events to send:
    process-crash: A process in the system has crashed
    liveness-failure: A process in the system was detected as hung
```

CHAPTER 10

Troubleshooting Media Flow Controller

- [Viewing Information Using Show Commands](#)
- [Internal Watchdog](#)
- [Testing Network Connectivity](#)
- [Testing Media Flow Controller Delivery Functions](#)
- [Testing HTTP Origin Fetch](#)
- [Testing NFS Origin Fetch](#)
- [Testing a Specific Transaction](#)
- [Enabling Debug Operations](#)
- [Troubleshooting Media Flow Controller Invalid Licenses](#)
- [Troubleshooting namespace match uri Configuration](#)
- [Troubleshooting namespace domain Configuration](#)
- [Troubleshooting File Not Getting Cached](#)
- [Troubleshooting Cache Promotion Not Happening](#)
- [Troubleshooting Incoming Requests' URL Length](#)
- [Troubleshooting Accesslog SFTP](#)
- [Troubleshooting Lost Admin Password](#)
- [Troubleshooting No Web Interface Access](#)
- [Troubleshooting Accesslog Rotation Intervals](#)

See also [Chapter 9, “Configuring and Using Media Flow Controller Logs and Alarms”](#).

Viewing Information Using Show Commands

Purpose

Find information about Media Flow Controller performance issues.

Media Flow Controller CLI **show** commands let you find the system information you need. This section excludes commands that show settings deemed not useful for fault management, including **show** commands for **banner**, **cli**, **clock**, **delivery**, **email**, **ip**, **license**, **logging**, **ntp**, **radius-server**, **snmp-server**, **ssh**, **tacacs-server**, **telnet**, **terminal**, **username**, **web**, and **whoami**. See [Table 15, “Logging Status \(%s\) HTTP Codes”](#) and [Table 16, “Additional Logging Status Sub-Codes”](#) for descriptions of HTTP response codes and sub-codes.

Action

show analytics—Analytics manager (default settings shown); these include:

- Cache Promotion—Enabled
- Cache Ingest Hit Threshold—3
- Cache Ingest Size Threshold—0
- Cache Ingest Last Eviction Time Diff—1
- Cache Evict Aging Time—10
- Cache Promotion Hotness Increment—100
- Cache Promotion Hotness Threshold—3
- Cache Promotion Hit Increment—100
- Cache Promotion Hit Threshold—10

You can configure some of these settings with **analytics** commands.

show bootvar—The installed images on partition 1 and partition 2, from which partition was the last boot, and which partition is set as the next boot.

show configuration—Lists the CLI commands needed to bring the state of a fresh system up to match the current persistent (saved) state of this system. A short header is included, containing the name and version number of the configuration, in a comment. Commands that have not been saved, or would set something to its default, are not included, so this command on a fresh configuration produces no output, except the header. Includes arguments:

- **files [<filename>]**—If no **filename** is specified, list the configuration files in persistent storage. If **filename** is specified, list the commands to recreate the configuration in that file; only non-default commands are shown.
- **full**—Same as **show configuration** but includes commands that set default values.
- **running**—Same as **show configuration** except that it applies to the currently running configuration, rather than the active saved configuration.
- **text files**—List text-based configuration files.

show counters—Lists the following information. See [“Media Flow Controller Log Codes and Sub-Codes” on page 174.](#) for descriptions of HTTP response codes:

```
Total number of Active Connections
Total Bytes served from RAM cache
Origin Server Counters
    Total Bytes served from Origin Server
    Total Bytes served from HTTP Origin Server
    Total Bytes served from NFS Origin Server
    Total Bytes served from RTSP Origin Server
Total Bytes served from Disk cache
Total Bytes served
Total Bandwidth
Ingest Fetch Count
Ingest Bytes Fetched
Total Disk Read Operations
Total Disk Write Operations
Virtual Player Number of Seeks
Virtual Player Hash Verification Failed Errors
```

```
Total Number of ports
Active Connections on Port    eth0
Active Connections on Port    lo
Number of Requests on Namespace    newTest
    2225
Total number of Active HTTP Connections
Total number of HTTP Connections
Total number of HTTP Transactions
Total number of HTTP 200 responses
Total number of HTTP 206 responses
Total number of HTTP 302 responses
Total number of HTTP 304 responses
Total number of HTTP 400 responses
Total number of HTTP 404 responses
Total number of HTTP 416 responses
Total number of HTTP 500 responses
Total number of HTTP 501 responses
Total number of HTTP Timeouts
Total HTTP Well finished count
Total Number of Cache-Miss
    22
HTTP TUNNEL STATS
Total Connections
Total Active Connections
Total Bytes Served
Total Errors
ERROR COUNTERS
    Number of Scheduler Errors on get data
    Number of HTTP deadline missed tasks
PROXY ERRORS
    OM Error Connection Failed Count
    HTTP Parse Error Count
Cluster L7 statistics
    Total number of Redirects
    Total number of Redirect errors

Total number of Active RTSP Connections
Total number of RTSP Transactions
Total Number of RTP Packets Forwarded
Total number of RTSP 200 responses
Total number of RTSP 400 responses
Total number of RTSP 404 responses
Total number of RTSP 500 responses
Total number of RTSP 501 responses

RTCP Statistics
Total Number of RTCP Packets Forwarded
Total Number of Sender Report sent
Total Number of Receiver Report sent
Total Number of RTCP Packets Received
Total Number of Sender Report Received
Total Number of Receiver Report Received

PROXY ERRORS
    OM Error Connection Failed Count
```

RTSP Parse Error Count

show files—List available files or their counters. Includes these arguments:

- **debug-dump**—List of debug dump files.
- **stats**—List of statistics reports.
- **system**—Filesystem information; includes these statistics for **/config** and **/var**:
 - Bytes Total
 - Bytes Free
 - Bytes Used
 - Bytes Available
 - Bytes Percent Free
 - Inodes Total
 - Inodes Free
 - Inodes Used
 - Inodes Percent Free
- **tcpdump**—List **tcpdump** files.

show hosts—Hostname, DNS configuration, and static host mappings.

show images—Information about system images and boot parameters.

show interface [configured]—Information about each system interface including:

- **Admin** state—Whether or not the interface is enabled.
- **Link** state—**Up** means there is a cable plugged into that interface and it is “live,” being connected to something which is turned on at the other side; for example, a switch, router, or another computer.
- Current **TX** (transmissions out) and **RX** (transmissions received) statistics.

Use **configured** to see the settings of the interfaces, rather than their runtime state.

show log—View event logs including commands executed during this session.

show media-cache disk—The **list** argument lists available caches and information on each including **Device** (name), **Type**, (cache), **Tier** (1=SSD, 2=SAS, 3=SATA), **Active** status, **Enabled** status, **Free Space**, and **Disk State**. Use **show media-cache disk <disk_name>** for details on a particular disk.

show media-cache free-block threshold—The **free-block threshold** argument lists the free-block threshold of the disk caches.

show memory—Memory usage; includes Total, Used, and Free for Physical and Swap.

show namespace {list | <namespace_name> [counters | object]}—Get namespace-specific information:

- **list**—List the configured namespaces and their UIDs.
- **<namespace_name>**—List settings for the specified namespace.
- **<namespace_name> object list {all | uri | pattern}**—List objects in the specified namespace.
- **<namespace_name> counters**—List the specified namespace's counters including:

HTTP Resource Monitoring Counters:

Client Active Sessions:

```

Current Bandwidth:
Served Bytes:
Transaction Per sec:
Cache Hit Ratio(Bytes):
Cache Hit Ratio(Req):
HTTP Client Counters:
  Number of requests:
  Number of responses:
  Total Bytes Received:
  Total Bytes Sent:
    From Memory Cache:
    From Disk Cache:
    From Origin:
  Cache Hits:
  Cache Misses:
  Responses with 2xx status code:
  Responses with 3xx status code:
  Responses with 4xx status code:
  Responses with 5xx status code:
HTTP Origin Counters:
  Number of requests:
  Number of responses:
  Total Bytes Received:
  Total Bytes Sent:
  Responses with 2xx status code:
  Responses with 3xx status code:
  Responses with 4xx status code:
  Responses with 5xx status code:

```

show network—Network configurations; includes time out, max connections, session assured flow rate, and session max bandwidth settings.

show ram-cache—Buffer-manager configuration; defaults are sync-interval = **86400 seconds**, object minimum size = **0** (zero), revalidate-window = **120 seconds**, and maximum RAM cache size = **0** (zero) or **AUTO**. Also lists current RAM cache size.

show running-config—Lists commands to recreate the currently running configuration.

show resource-pool global-pool—List the currently-available system-wide resources including maximum allowed concurrent sessions, maximum available bandwidth, and cache tier memory limits per disk.

show service—Configuration and status (current status, number of failures, last terminated, and uptime) for these services: Media Flow Publisher file publishing (**mod-file-publisher**), pre-stage FTP (**mod-ftp**), Media Flow Publisher live publishing (**mod-live-publisher**), accesslog and errorlog (**mod-log**), offline origin manager (**mod-oom**), delivery (**mod-delivery**), and FMS service if FMS is installed (**mod-rtmp-fms** and **mod-rtmp-admin**).

show statistics—Key statistics; including:

- **Current Bandwidth (MB/Sec)**—The rate at which Media Flow Controller is currently delivering the service.
- **Current Cache Bandwidth (MB/Sec)**—The delivery bandwidth at which Media Flow Controller is delivering from cache, excluding deliveries from the origin directly.
- **Current Disk Bandwidth (MB/Sec)**—The delivery bandwidth coming from objects in the disk; should be a subset of Current Cache Bandwidth.

- **Current Origin Bandwidth (MB/Sec)**—The delivery bandwidth for objects being fetched from the origin and directly delivered.
- **Avg Number of Connections Per Sec**—On average, the connection accept rate.
- **Avg HTTP Transactions per Sec**—On average, number of completed HTTP transactions.
- **Avg Cache Bandwidth (MB/Sec)**—On average, the data fetch rate from buffer to network for delivery.
- **Avg Disk Bandwidth (MB/Sec)**—On average, the data fetch rate from disk.
- **Avg Origin Bandwidth (MB/Sec)**—On average, the data fetch rate from origin (happens only when there is a cache miss).
- **Avg Proxy Rate (Origin Manager) (MB/Sec)**—Total bytes received from origin, sampled every 5 seconds and averaged since Media Flow Controller start.
- **Current Proxy Rate in this sec (MB/Sec)**—Rate of GET requests made in one second.
- **Per Disk Bandwidth (MB/Sec)**—Lists available disks and the bandwidth for each.
- **Per port statistics (TX Bytes Per Sec on <interface>)**

show stats <type>—List statistics (stats) settings and gathered data; includes these statistic types (see [stats](#) for CLI details on alarm, chd, and sample types):

- **alarm**—List status and configuration of statistics-based alarms.
- **chd**—List configuration of statistics CHDs.
- **cpu**—List CPU statistics.
- **sample**—List configuration of statistics samples.

show system—Lists system configuration (debug level and mod).

show users—Lists information about user logins.

show version—Lists version information for current system image.

show virtual-player [list] [<name>]—Lists virtual player settings; use **list** to see a list of defined virtual players and their type; use **<name>** to see settings.

Internal Watchdog

Media Flow Controller now has a watchdog to monitor the service layer, including monitoring the health of data delivery to users, origin fetch, and disk input/output (I/O) functions. This internal watchdog mechanism checks the proper functioning of the service layer at a regular frequency and restarts the service if the check fails. This ensures that Media Flow Controller only loses service for a short duration if the service layer is stuck due to a defect.

As a part of this feature there is a new default namespace called **mfc_probe**. This namespace is configured as a reverse proxy with the internal management Web server as its origin server. There is a canned object file in this origin server that you can request to trigger a full test of the service path. You can also use this to configure any probes to request this canned object; for example, load balancers typically expect probes to be configured and the **mfc_probe** namespace can be used as such.

The probe URI is:

```
http://<MFC_IP>/mfc_probe/mfc_probe_object.dat
```

Media Flow Controller watchdog probes the service layer by sending an HTTP request (probe) to the default **mfc_probe** namespace every 5 seconds. If a response arrives within the timeout period, the service layer is alive; if there is no response three times in a row, Media Flow Controller determines that the service layer has gotten stuck and it is restarted. While restarting the service layer, a core file is generated at /coredump/snapshots. By default the **mfc_probe** namespace is active; if you want to disable it, use **namespace mfc_probe status inactive**. Probe activity is not logged to the accesslog.

Testing Network Connectivity

Purpose

To make sure your network connections are configured and behaving properly.

Action

- Confirm that your computer has the appropriate settings for DNS servers, the correct hostname, and an available IP address with the proper subnet mask (**show hosts**); the proper default gateway (**show ip default-gateway**).
- Check network interfaces (**show interfaces**). Make sure the links are up, the addresses are correct, default routes are set correctly (**show ip route**). Check cable connectivity.
- Check ARP tables (**show arp**).
- Check DNS /hostname configurations (**show hosts**).
- Check that applications like SSH and Firefox are working.
- Test connections to remote servers; use **ping (Ctrl+c to stop ping)**:
 - a. Ping the loopback address (by using the ping 127.0.0.1 command) to verify that TCP/IP is installed and working correctly on the local computer.
 - b. Ping the local computer IP address to verify it was added to the network correctly.
 - c. Ping the IP address of the default gateway to verify that the gateway is functional and it is possible to connect to a local host on the local network.
 - d. Ping the IP address of another remote host to verify that you can communicate through a router.
- Check the network path to a destination with **traceroute**:
 - a. At the command prompt, type **traceroute <IP_address_of_remote_network_host>**, and then press ENTER.
 - b. Examine the results to determine the length of time that the packet took to reach each network segment and the point at which the connection may stop working.

Table 25 TCP/IP Diagnostic Utilities

| Utility | Used to... |
|-----------------|---|
| show arp | View the ARP (address resolution protocol) table on the local computer to detect invalid entries. |

Table 25 TCP/IP Diagnostic Utilities (Continued)

| Utility | Used to... |
|--|--|
| <code>show hosts</code> | View hostname, DNS configuration, and static host mappings |
| <code>show ip {default-gateway route}</code> | View current TCP/IP network configuration values |
| <code>ping</code> | Verify whether TCP/IP is configured correctly and that a remote TCP/IP system is available |
| <code>tracert</code> | Check the route to a remote system |

Testing Media Flow Controller Delivery Functions

This test, using [Wget](#) (a free file retrieval program), demonstrates how Media Flow Controller is using a configured namespace to fetch and deliver content via HTTP or NFS, and Media Flow Controller caching mechanisms.

To perform this test you need a client machine with Wget installed that can also serve as the origin server, and a Media Flow Controller, and connectivity between the two of them. The setup for this example procedure includes creating data files (**test.txt**) and placing them in a directory (**testresults/joe**) on a Unix machine with Wget installed, serving as client and origin (**172.16.254.1 / sv05**).

The actions for this example procedure are configuring a namespace (**testHttp**), requesting files via the Media Flow Controller (**172.16.254.2 / MFC**), and observing results using **show counters** and the accesslog on the Media Flow Controller Web interface.

Testing HTTP Origin Fetch

Purpose

Test the Media Flow Controller HTTP origin fetch function.

Action

Prepare for the test by doing the following, then follow the steps as illustrated in [Figure 9](#).

- Log into the client/origin UNIX machine and go to a test directory; for example, **testresults/joe**; create a simple text file, **test.txt**, and add some content to give the file some weight.
- Log into the Media Flow Controller and configure a namespace, **testHttp**; specify a **uri-prefix** with a domain, delivery protocol, and origin server, and make the namespace active. Example:

```
MFC (config) # namespace testHttp
MFC (config namespace testHttp) # delivery protocol http
MFC (config namespace testHttp) # domain any
MFC (config namespace testHttp) # match uri /testresults/joe
MFC (config namespace testHttp) # origin-server http sv05
MFC (config namespace testHttp) # status active
MFC (config namespace testHttp) # exit
```

- From the client/origin machine, use **wget** to fetch the file locally (verify Wget). Example:


```
[joe@sv05 joe]$ wget http://172.16.254.1/testresults/joe/test.txt
--13:12:58-- http://172.16.254.1/testresults/joe/test.txt
Connecting to 172.16.254.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 165 [text/plain]
Saving to: `test.txt.2'
100%[=====] 165      --.-K/s   in 0s
13:12:58 (15.7 MB/s) - `test.txt.2' saved [165/165]
```
- Now use **wget** to fetch the file via Media Flow Controller. When Media Flow Controller receives the first request for that namespace, it begins logging. Media Flow Controller receives the request, matches the **uri-prefix** to the namespace, and uses that namespace's defined origin server to retrieve the content. Use **show counters** on the Media Flow Controller to see what happened. Example (output truncated):


```
[joe@sv05 joe]$ wget http://172.16.254.2/testresults/joe/test.txt
--13:18:00-- http://172.16.254.2/testresults/joe/test.txt
Connecting to 172.16.254.2:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 165 [text/plain]
Saving to: `test.txt.3'
100%[=====] 165      --.-K/s   in 0s
13:18:00 (26.2 MB/s) - `test.txt.3' saved [165/165]
[joe@sv05 joe]$
MFC (config) # show counters
Total number of Active Connections      : 0
Total Bytes served from RAM cache       : 0 Bytes
Total Bytes served from Origin Server: 165 Bytes
    Total Bytes served from HTTP Origin Server : 165 Bytes
    Total Bytes served from NFS Origin Server : 0 Bytes
Total Bytes served from Disk cache      : 0 Bytes
Total Bytes served                      : 165 Bytes
Total number of HTTP Connections        : 1
Total number of HTTP Transactions       : 1
Total number of HTTP 200 responses      : 1
Total HTTP Well finished count         : 1
MFC (config) #
```
- Run the test again to see Media Flow Controller serve the content from RAM. Example (output truncated):


```
[joe@sv05 joe]$ wget http://172.16.254.2/testresults/joe/test.txt
--14:07:21-- http://172.16.254.2/testresults/joe/test.txt
Connecting to 172.16.254.2:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 165 [text/plain]
Saving to: `test.txt.4'
100%[=====] 165      --.-K/s   in
0.002s
14:07:21 (83.4 KB/s) - `test.txt.4' saved [165/165]
[joe@sv05 joe]$
MFC (config) # show counters
Total number of Active Connections      : 0
Total Bytes served from RAM cache       : 165 Bytes
Total Bytes served from Origin Server   : 165 Bytes
    Total Bytes served from HTTP Origin Server : 165 Bytes
    Total Bytes served from NFS Origin Server : 0 Bytes
Total Bytes served from Disk cache      : 0 Bytes
```

```
Total Bytes served           : 330 Bytes
Total number of HTTP Connections : 2
Total number of HTTP Transactions : 2
Total number of HTTP 200 responses : 2
Total HTTP Well finished count : 2
```

- Run the test once more to see Media Flow Controller serve the content from Disk; first restart the delivery service so everything in RAM is moved to disk. Example:

```
MFC (config) # service restart mod-delivery
[joe@sv05 joe]$ wget http://172.16.254.2/testresults/joe/test.txt
--16:17:55-- http://172.16.254.2/testresults/joe/test.txt
Connecting to 172.16.254.2:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 165 [text/plain]
Saving to: `test.txt.3'
100%[=====>] 165          --.-K/s   in 0s
16:17:55 (13.1 MB/s) - `test.txt.3' saved [165/165]
[joe@sv05 joe]$
MFC-cl11 (config) # show counters
Total number of Active Connections           : 0
Total Bytes served from RAM cache           : 0 Bytes
Total Bytes served from Origin Server       : 0 Bytes
    Total Bytes served from HTTP Origin Server : 0 Bytes
    Total Bytes served from NFS Origin Server : 0 Bytes
Total Bytes served from Disk cache : 165 Bytes
Total Bytes served           : 495 Bytes
Total number of HTTP Connections : 3
Total number of HTTP Transactions : 3
Total number of HTTP 200 responses : 3
Total HTTP Well finished count : 3
```

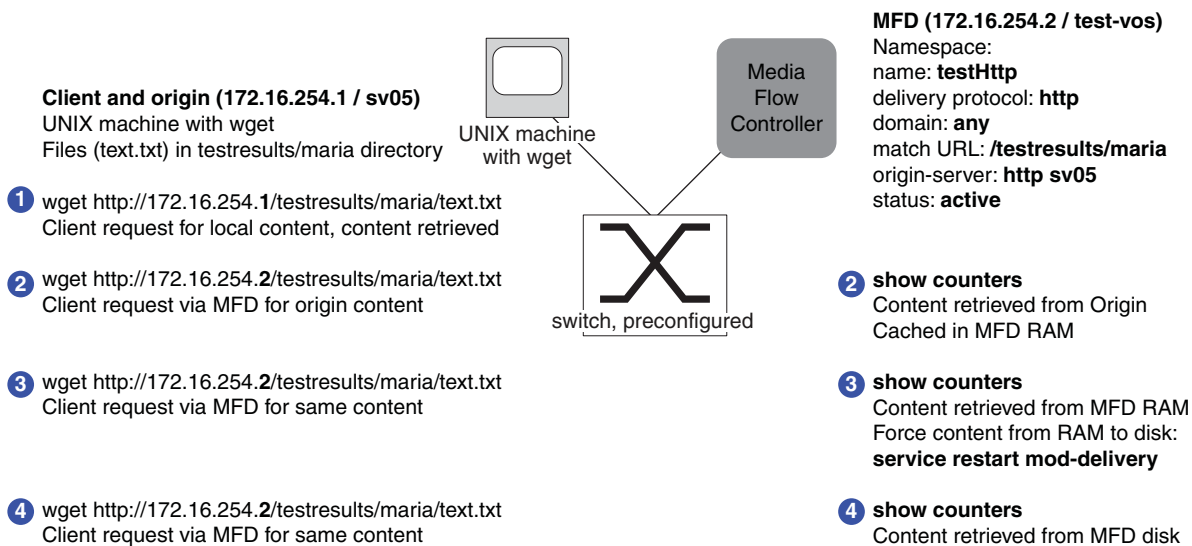


Figure 9 wget Test for Media Flow Controller HTTP Delivery and Cache

Testing NFS Origin Fetch

Purpose

Test the Media Flow Controller NFS origin fetch function.

NFS origin fetch is very similar to HTTP origin fetch, but the namespace configuration differs slightly. NFS has much more functionality than HTTP. When you configure the namespace, you give the URI origin-server NFS IP address (or hostname) and full path; the **uri-prefix** can be anything (for example **nfs1**) and NFS automatically creates that directory when the first request comes in. The request must include the configured **uri-prefix**.

Action

Prepare for the test by doing the following, then follow the steps as illustrated in [Figure 9](#) (note the **wget** path change for the NFS test).

- Log into the client/origin machine and go to a test directory; for example, **testresults/joe**; create a simple text file, **test.txt**, and add some content to give the file some weight.
- On the Media Flow Controller, create a new namespace, **testNfs**, and specify a uri-prefix with a domain, delivery protocol, and origin server; then make the namespace active.

Example:

```
MFC (config) # namespace testNfs
MFC (config namespace testNfs) # domain any
MFC (config namespace testNfs) # match uri /nfs1
MFC (config namespace testNfs) # origin-server nfs sv05:home/joe
MFC (config namespace testNfs) # status active
MFC (config namespace testNfs) # exit
```

1. From the client/origin machine, use **wget** to fetch the file locally (verify Wget). Example:

```
[joe@sv05 joe]$ wget http://172.16.254.1/testresults/joe/test.txt
--13:12:58-- http://172.16.254.1/testresults/joe/test.txt
Connecting to 172.16.254.1:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 165 [text/plain]
Saving to: `test.txt.2'
100%[=====>] 165          --.-K/s   in 0s
13:12:58 (15.7 MB/s) - `test.txt.2' saved [165/165]
```

2. From the client/origin machine, use **wget** to fetch the file via Media Flow Controller. When Media Flow Controller receives the first request for that namespace, it begins logging. Media Flow Controller receives the request, matches the **uri-prefix** to the namespace, and uses that namespace's defined origin server to retrieve the content. Use **show counters** on the Media Flow Controller to see what happened. Example (output truncated):

```
[joe@sv05 joe]$ wget -O newtest http://172.16.254.2/nfs1/test.txt --
17:34:26-- http://172.16.254.2/nfs1/test.txt
Connecting to 172.16.254.2:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 165 [text/html]
Saving to: `newtest'
100%[=====>] 165          --.-K/s   in 0s
17:34:26 (26.2 MB/s) - `newtest' saved [165/165]
[joe@sv05 joe]$
MFC (config) # show counters
```

```
Total number of Active Connections      : 0
Total Bytes served from RAM cache       : 0 Bytes
Total Bytes served from Origin Server   : 165 Bytes
    Total Bytes served from HTTP Origin Server : 0 Bytes
    Total Bytes served from NFS Origin Server : 165 Bytes
Total Bytes served from Disk cache      : 0 Bytes
Total Bytes served                      : 165 Bytes
Total number of HTTP Connections       : 1
Total number of HTTP Transactions      : 1
Total number of HTTP 200 responses     : 1
Total HTTP Well finished count        : 1
MFC (config) #
```

3. Run the test again to see Media Flow Controller serve the content from RAM. Example (output truncated):

```
[joe@sv05 joe]$ wget -O newtest http://172.16.254.2/nfs1/test.txt
--14:07:21-- http://172.16.254.2/nfs1/test.txt
Connecting to 172.16.254.2:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 165 [text/plain]
Saving to: `newtest'
100%[=====>] 165          --.-K/s   in
0.002s
14:07:21 (83.4 KB/s) - `newtest' saved [165/165]
[joe@sv05 joe]$
MFC (config) # show counters
Total number of Active Connections      : 0
Total Bytes served from RAM cache       : 165 Bytes
Total Bytes served from Origin Server   : 165 Bytes
    Total Bytes served from HTTP Origin Server : 0 Bytes
    Total Bytes served from NFS Origin Server : 165 Bytes
Total Bytes served from Disk cache      : 0 Bytes
Total Bytes served                      : 330 Bytes
Total number of HTTP Connections       : 2
Total number of HTTP Transactions      : 2
Total number of HTTP 200 responses     : 2
Total HTTP Well finished count        : 2
MFC (config) #
```

4. Run the test once more to see Media Flow Controller serve the content from Disk; first restart the delivery service so everything in RAM is moved to disk. Example:

```
MFC (config) # service restart mod-delivery
[joe@sv05 joe]$ wget -O newtest http://172.16.254.2/nfs1/test.txt
--16:17:55-- http://172.16.254.2/nfs1/test.txt
Connecting to 172.16.254.2:80... connected.
HTTP request sent, awaiting response... 200 OK
Length: 165 [text/plain]
Saving to: `newtest'
100%[=====>] 165          --.-K/s   in 0s
16:17:55 (13.1 MB/s) - `newtest' saved [165/165]
[joe@sv05 joe]$
MFC-cl11 (config) # show counters
Total number of Active Connections      : 0
Total Bytes served from RAM cache       : 0 Bytes
Total Bytes served from Origin Server   : 0 Bytes
    Total Bytes served from HTTP Origin Server : 0 Bytes
    Total Bytes served from NFS Origin Server : 165 Bytes
```

```
Total Bytes served from Disk cache : 165 Bytes
Total Bytes served                   : 495 Bytes
Total number of HTTP Connections     : 3
Total number of HTTP Transactions    : 3
Total number of HTTP 200 responses   : 3
Total HTTP Well finished count      : 3
```

Testing a Specific Transaction

Problem

A specific video is having delivery problems.

Solution

Use the trace log facility to debug.

Media Flow Controller includes a trace facility to help diagnose the handling of a particular HTTP request. The tracing is done by logging trace points, written to the Trace Log. A request is traced if the following conditions are true:

- The global trace flag is enabled via the CLI:
`delivery protocol http trace enable`
- The HTTP request includes the **X-NKN-Trace** header; for example if using Wget:
`wget --header "X-NKN-Trace:" <URL>`

The HTTP modules detect the above conditions and set the flag "HRF_TRACE_REQUEST" as well as other flags that direct each relevant module to log meaningful trace points.

To trace the path of a specific video:

1. Set the delivery trace:
`delivery protocol http trace enable`
2. Using WGET (or other tool, cURL, and so forth) to request the problem video with this header added:
`X-NKN-Trace`

To see the trace points and for additional information, see ["Reading the Trace Log \(tracelog\)" on page 195](#).

Enabling Debug Operations

Media Flow Controller provides a system-level debugging utility as well as a trace facility; see ["Reading the Trace Log \(tracelog\)" on page 195](#) for details.



CAUTION: The system gives high priority to debugging output. For this reason, debugging commands should be turned on only for troubleshooting specific problems or during troubleshooting sessions with technical support personnel. Excessive debugging output can render the system inoperable.

1. Generate debugging information for all system functions.
`debug generate dump`
2. View a list of debug-dump files.
`show files debug-dump`
3. View a summary of a specific debug-dump.
`show files debug-dump <filename_of_debug_dump>`
4. E-mail a generated debug-dump file to the list of recipients configured to receive **info** event notices (see [“Configuring Media Flow Controller Fault Notifications \(CLI\)” on page 208](#) for task details).
`file debug-dump email <filename_of_debug_dump>`
5. Upload a specific debug-dump file. The uploaded file is a gnu-zipped .tar file (.tgz) and can be unzipped with this command on Linux: `gunzip -c <filename>.tgz | tar tf -` or with WinZip on a Windows system.
`file debug-dump upload <filename_of_debug_dump> <URL>`



NOTE: Only FTP and TFTP URLs, as well as SCP pseudo-URLs are supported for the destination. See [“Terminology” on page 31](#) for the **scp** format.

Troubleshooting Media Flow Controller Invalid Licenses

Problem

Media Flow Controller licenses may be invalid. You may have older licenses that used a different scheme.

Solution

Release 2.0 and earlier Media Flow Controller licenses are tied to the MAC address of eth0. Typically you must manually set the eth0 interface at installation with one of the **Ethernet naming** options during installation. If not specified, the default assignment of ports is the order in which the drivers are loaded and if eth0 is not assigned the MAC address associated with your licenses at that time, your licenses will be invalid. If that happens, you should re-install Media Flow Controller and use the **Ethernet naming** options to name the correct interface eth0. However, if re-installation is not an option, you can use the management interface command to name the correct interface as eth0.



CAUTION: While this command does name the specified interface eth0, it also renames all other interfaces, which can cause other problems if other interfaces are configured for specific reasons.

To use the management interface command to name eth0:

1. Determine what MAC address your licenses are tied to:
`show license`
2. Determine which interface has the MAC address to which your licenses are tied (the last two octets of the **HWaddr** are what you need):


```
show interface
```

3. Type the following command to name that interface as the eth0 management interface:

```
management interface <MAC_address>
```

Troubleshooting namespace match uri Configuration

Problem

Requests are being misdirected. You may have configured the namespace **match uri <uri-prefix>** incorrectly or you may need to set **precedence**.

Solution

Proper namespace configuration is required for smooth Media Flow Controller functioning. Namespace is how Media Flow Controller knows what to deliver and where to fetch it, if needed. An important element is the **match** criteria for the namespace. A **match uri <uri-prefix>** can be very specific; for example, /vod/path1/path2; or very un-specific; for example, / (slash). An un-specific **<uri-prefix>** can be thought of as a super-set. In the case of a **<uri-prefix>** of simply / (slash) all video requests are going to map to that namespace because all video requests are going to have a / (slash) in them. If a super-set **<uri-prefix>** is desired, it is important to also set a precedence for that namespace so Media Flow Controller knows to look at other namespaces first. For more information, see ["Using namespace match <criteria> precedence" on page 145](#).

Troubleshooting namespace domain Configuration

Problem

Incoming requests are being rejected. You may have configured the **namespace domain** incorrectly.

Solution

When configuring **namespace domain**, make sure the domain you enter matches whatever you expect as HOST header in requests; you may append a port number as well if needed (and used in HOST header). If you append a port number that is not the default and is not in the HOST header, the request will fail. See ["Using namespace domain <FQDN:Port>" on page 144](#) for more details.

Troubleshooting File Not Getting Cached

Problem

File fetched from origin is not getting cached in disk. The **Cache-Control Max-Age** header value could be set too low.

Solution

If you observe a file fetched from origin but not getting cached in disk, the problem could be that the HTTP **Cache-Control Max-Age** header is set to a value less than 60 seconds. In that case, that file is never cached in disk.

You can modify the header value to be greater than 60 seconds using the **delivery protocol** commands. For details, see [“Configuring Media Flow Controller Delivery Protocols \(CLI\)” on page 102](#).

Troubleshooting Cache Promotion Not Happening

Problem

Objects in cache are not getting promoted correctly. You might have disabled cache promotion while debugging.

Solution

- Verify that cache promotion has not been disabled. Use the **show analytics** command.
- If cache promotion has been disabled, enable it by using the **analytics cache-promotion enable** command.

Troubleshooting Incoming Requests' URL Length

Problem

Incoming requests are being rejected. The URL length might have exceeded the maximum allowed length.

Solution

The default acceptable URL (domain + URI + Query Params + Headers) length, in characters/bytes, for incoming requests is **16384** bytes; maximum allowed value is **32768**. Incoming requests with lengths exceeding the set value are rejected. You can modify this with the **delivery protocol** argument **req-size incoming maximum**. See [delivery](#) for CLI details.

There are 2 thresholds:

- Max URL length—This is the length of the URL itself. A URL consists of (domain + (URI + query params)). The size of a URI cannot exceed 512 bytes. Any URI beyond this size is treated as non-cacheable. There is no CLI to configure this limit at present.
- Max Request length—This is the total size of a request which, as described, must not exceed the configured size.

Troubleshooting Accesslog SFTP

Problem

Sending the accesslog via SFTP.

Solution

Access log **on-the-hour** rotation configuration takes precedence over accesslog **rotate time-interval** configuration. You must disable the **on-the-hour** rotation configuration if you want to upload accesslog more frequently.

To disable accesslog **on-the-hour** rotation:

```
(config) # accesslog on-the-hour disable
```

To configure the accesslog copy for SFTP:

```
(config) # accesslog copy <sftp://user@host:path>
```

To setup the SSH keys:

1. Take the RSA key from the target host that you want to push the logs to; for example **/etc/ssh/ssh_host_rsa_key.pub**, and add it to Media Flow Controller:

```
(config) # ssh client global known-host "<IP> ssh-rsa ..."
```

2. Generate the public/private keys in Media Flow Controller:

```
(config) # ssh client user admin identify rsa2 generate
```

3. Take Media Flow Controller's public key and put in the target host's authorized keys:

```
(config) # show ssh client
```

Take the public key (Media Flow Controller); for example:

```
User admin:
RSAv2 Public key:
ssh-rsa ....
```

Put it here (Client machine); for example **/user_name/.ssh/authorized_keys**.

4. To verify, do a manual upload:

```
(config) # upload accesslog current sftp://user@host:path (i.e. sftp://
root@192.168.1.10:/tmp)
```

The first time you do it you'll see something like this:

```
(config) # upload accesslog current sftp://root@192.168.1.10:/tmp
```

```
sftp> cd /tmp
```

```
sftp> put access2.log access2.log.tmp
```

```
Uploading access2.log to /tmp/access2.log.tmp
```

```
sftp> -rm access2.log
```

```
Couldn't stat remote file: No such file or directory Removing /tmp/
access2.log Couldn't delete file: No such file or directory
```

```
sftp> rename access2.log.tmp access2.log exit
```

```
(config) #
```

Troubleshooting Lost Admin Password

Problem

Administrator's password is lost.

Solution

The procedure for resetting a lost Admin password for Media Flow Controller is based on the conventional procedure for resetting a lost Linux password, with a few important differences. The most important difference is that Media Flow Controller uses a database to store passwords instead of the standard Linux **/etc/passwd** file. So the procedure described below uses a shell script to reset the password database.



NOTE: The procedures described in this section assume broad familiarity with Linux system administration.

About the Media Flow Controller Boot Process

Media Flow Controller uses a standard Linux boot process shown in [Figure 10](#). To reset the admin password, you must interrupt the boot process and force Media Flow Controller into Linux single user mode. Then you run a shell script to reset the password database.

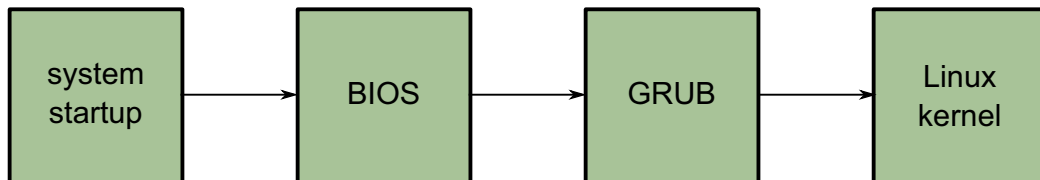


Figure 10 Media Flow Controller Boot Process

The procedures below are based on the GRUB bootloader and the modifying the Linux kernel command line.

Note: This procedure assumes that the boot media priority has not been altered from the system default (hard drive).

Password Database Reset Procedure

The procedure below resets the password database for Media Flow Controller.

1. Force a reboot:
 - If the CLI is still running:
 - a. Enter enable mode:
MFC> **enable**
 - b. Reboot Media Flow Controller:
MFC> **reload**

If CLI is not running, then you probably cannot log into the **admin** account. In this case, it is necessary to perform a hardware reboot.
 - 2. Enter the GRUB bootloader command-line interface by pressing and releasing the <SPACE> key during the initial boot sequence after GRUB starts displaying a series of periods (...).
You should see the following GRUB bootloader menu which describes two operating system images that can be booted:


```

          -----
          0: mFc-2.1.0 180_13291_229
          1: mFc-2.1.0 180_13291_229
          -----
          
```
 - 3. Select an operating system image with the arrow keys. In most cases, the first image can be used.
 - 4. Press the **e** key to choose the selected operating system image.
You should see a series of operating system kernel command lines:

```

-----
0: root (hd0,1)
1: kernel /vmlinuz ro root=LABEL=ROOT_1 rootdelay=5 img_id=1 quiet
  loglevel=4 panic=10 console=tty0 console=ttyS0,9600n8
2: initrd /initrd.img
-----

```

5. Select line 1 with the arrow keys.
6. Press the **e** key to begin editing the kernel command-line options.
You should see following kernel command line for editing:

Highlighted entry is 1:

```
[ Minimal BASH-like line editing is supported. For the first word, TAB
  lists possible command completions. Anywhere else TAB lists the
  possible completions of a device/filename. ESC at any time cancels.
  ENTER at any time accepts your changes.]
```

```
<1 quiet loglevel=4 panic=10 console=tty0 console=ttyS0,9600n8
```

7. Edit the kernel command line.
 - a. Select line 1 by using the arrow keys (i.e. press the down arrow key once). The message at the bottom should now indicate:
Highlighted entry is 1:
 - b. Append the word **single** to the end of the command line.
This change will be temporary and will be reset automatically at the next reboot.
 - c. Press the **<ENTER>** key to complete the changes to the command line.
 - d. Press the **b** key to complete the boot sequence into single-user mode.
8. Reset the password database.

Note: Typical Linux systems store their passwords in the **/etc/passwd** file. Media Flow Controller uses a database to maintain passwords. The password reset script **/sbin/resetpw.sh** performs all the necessary changes to this database.

- a. Run the password reset script:

```
# /sbin/resetpw.sh
```

If successful, the **admin** password will now be reset to the empty password. If any corruption in of configuration state is detected, an attempt to save existing configuration databases is made and the system's configuration state is reset to initial values (i.e. the same state just after manufacturing the system). In this case, the **admin** password will be set to an empty password. The command will notify the user of any locations of saved configuration state.

- b. Reboot:

```
# reboot
```

The changes made to the kernel command-line previously will be reset automatically and the system will reboot normally with a reset password database.

Troubleshooting No Web Interface Access

Problem

The Web interface is not connecting.

Solution

Access to the Web interface does rely on several factors:

- You must have an IP address set for the management interface, typically eth0, though this is not a requirement.
- You must have a default gateway (Internet or network access) set, unless you are plugged directly into the console.
- The IP address you enter on the Location bar of your browser must specify port 8080 (add “:8080” after the IP address).
- The Web interface must be enabled (**web enable**); enabled to listen (**web httpd listen enable**); and assigned to eth0 (**web httpd listen interface eth0**). These are all default settings, but could need resetting if the Web interface is not connecting.

Troubleshooting Accesslog Rotation Intervals

Problem

The accesslog is not rotating as expected.

Solution

Media Flow Controller allows an administrator to control how frequently access logs are rotated. **Accesslog** has two configurations that control rotation frequency:

- **accesslog on-the-hour** sets hourly log rotation.
- **accesslog rotate time-interval** sets log rotation at a specific time interval.

The **on-the-hour** rotation configuration has a higher precedence than **rotate time-interval** rotation configuration. To upload accesslog files more frequently, it is necessary to first disable the **on-the-hour** rotation configuration.

To disable **on-the-hour** rotation:

```
(config) # accesslog on-the-hour disable
```

Then configure the **time-interval** rotation for 2 hours:

```
(config) # accesslog rotate 2
```

CHAPTER 11

Configuring Media Flow Controller (Web Interface)

- [About the Media Flow Controller Web Interface](#)
- [Logging In to Media Flow Controller for the First Time \(Web Interface\)](#)
- [Configuring Media Flow Controller for the First Time \(Web Interface EZconfig\)](#)
- [Monitoring Media Flow Controller Statistics \(Web Interface\)](#)
- [System Configuration Overview \(Web Interface\)](#)
- [Configuring Interfaces, Default Gateway, Static Routes, DNS and Domain Names, Hostname, and Banners \(Web Interface\)](#)
- [Configuring Static Hosts and ARP \(Web Interface\)](#)
- [Configuring Date, Time, and NTP \(Web Interface\)](#)
- [Configuring RADIUS, TACACS+, and SSH \(Web Interface\)](#)
- [Configuring Users and AAA \(Web Interface\)](#)
- [Administering Media Flow Controller Overview](#)
- [Managing Configuration Files \(Web Interface\)](#)
- [Installing Licenses \(Web Interface\)](#)
- [Upgrading the System \(Web Interface\)](#)
- [Rebooting the System \(Web Interface\)](#)
- [Configuring the Web Interface \(Web Interface\)](#)
- [Service Configurations Overview](#)
- [Configuring Network Connections \(Web Interface\)](#)
- [Configuring Delivery Protocols \(Web Interface\)](#)
- [Configuring Virtual Players \(Web Interface\)](#)
- [Configuring NameSpaces \(Web Interface\)](#)
- [Managing the Media-Cache \(Web Interface\)](#)
- [Configuring Service Logging \(Web Interface\)](#)
- [Viewing Logs Overview](#)
- [Viewing the Dashboard Overview](#)
- [Viewing Reports \(Interface Statistics\)](#)
- [Using Media Flow Controller Media Flow Publisher](#)

About the Media Flow Controller Web Interface

Before you configure Media Flow Controller for the first time, see [“Before You Configure Media Flow Controller” on page 82](#).

The Media Flow Controller Web interface, also referred to as the Management Console, provides a subset of the command line interface management functions; however, you can make many configurations with the Web interface. See [Figure 11 on page 232](#).

JUNIPER NETWORKS

Media Flow Controller Management Console Host: test-vos-cl66
(not logged in)

Please enter your username and password, then click "Login".

Account:

Password:

Login

© 2008-2010 [Juniper Networks, Inc.](#)

Figure 11 Media Flow Controller Login Page

To configure Media Flow Controller using the Media Flow Controller Management Console Web Interface, the Web UI and HTTP access to it must be enabled on the management interface, eth0. These settings are enabled by default; however, if you need to re-enable them, you must use the CLI **web** commands.

Connecting and Logging In

You can connect to the Web interface using the IP address of your Media Flow Controller in a browser on port 8080. The Media Flow Controller opens to a login page.

Each user account has at least one privilege level that determines what actions they can take in the Web interface Management Console:

- Administrator (**admin**)—Full privileges. Can enter **Enable** mode and **Config** mode. By default, can log in as **admin** without a password.
- Monitor (**monitor**)—Can view all configurations but cannot change any configurations. Can view Dashboard and Reports, but cannot view Logs. By default, can log in as **monitor** without a password.
- Unprivileged (**unpriv**)—Can view all configurations but cannot change any configurations. Can view Dashboard and Reports, but cannot view Logs. Cannot log in as **unpriv**.

Related Topics

- [“About Media Flow Controller” on page 37](#)
- [“Media Flow Controller Environment” on page 38](#)
- [“Media Flow Controller Minimum System Requirements” on page 38](#)
- [“Understanding Media Flow Controller” on page 39](#)

Logging In to Media Flow Controller for the First Time (Web Interface)

Before you log in to Media Flow Controller for the first time, see [“Before You Configure Media Flow Controller” on page 82](#).

To log in to the Media Flow Controller Web interface for the first time, you need the IP address assigned to the eth0 management interface.

1. Open a browser and enter the Media Flow Controller management IP address including the management port, 8080. For example:

```
http://192.168.1.100:8080
```

2. Log in with these default credentials (there is no default password).

User: **admin**

The **Monitoring Summary** page is displayed.

Configuring Media Flow Controller for the First Time (Web Interface EZconfig)

The **EZconfig** tab allows you to easily set up Media Flow Controller for the first time.

- [“Setting the System Hostname \(EZconfig\)” on page 234](#)
- [“Setting Network Parameters \(EZconfig\)” on page 234](#)
- [“Creating a Virtual Player \(EZconfig\)” on page 235](#)
- [“Adding a Namespace \(EZconfig\)” on page 236](#)
- [“Enabling Interfaces \(EZconfig\)” on page 237](#)

Setting the System Hostname (EZconfig)

To set the system hostname using EZconfig:

1. Click the **EZconfig** tab.
The **EZconfig** page is displayed; see [Figure 12](#) for graphic.

| System Hostname | |
|--------------------------------------|----------------------|
| Host Name (FQDN) | <input type="text"/> |
| DNS IP | <input type="text"/> |
| <input type="button" value="Apply"/> | |

Figure 12 EZ Config Page System Hostname Area

2. Enter information in the text boxes:
 - **Host Name (FQDN)**—Enter a FQDN hostname for the system.
 - **DNS IP**—Enter an IP address for your Domain Name System (DNS) server.
 Click **Apply**.
3. Click **Save** at the top of the page to make changes persistent.

Setting Network Parameters (EZconfig)

To set global network connection options using EZconfig:

1. Click the **EZconfig** tab.
The **EZconfig** page is displayed; see [Figure 13](#) for graphic.

| Network Parameters | |
|--------------------------------------|-----------------------------------|
| Max-Bandwidth per Session(kbps) | <input type="text" value="0"/> |
| Network Max Connections | <input type="text" value="4900"/> |
| Assured Flow Rate (kbps) | <input type="text" value="0"/> |
| <input type="button" value="Apply"/> | |

Figure 13 EZ Config Page Network Parameters Area

2. Enter information in the text boxes:
 - **Max-Bandwidth per Session(kbps)**—Limit the allowed bandwidth for any one session. The actual session bandwidth is between the configured **Assured Flow Rate** and this value. Even if there is available bandwidth in the link, Media Flow Controller does not allocate more than this value for a session. When there is a full download, Media Flow Controller tries to allocate this value to the session. Default is **0** (unbounded) with the Media Flow Controller license, **200** kbps without it. You must have the Media Flow Controller license installed to change the default.
 - **Network Max Connections**—Limit the allowed number of concurrent sessions in Media Flow Controller. Default is **64000**; maximum allowed is **250,000**. You must

have the Media Flow Controller license installed to change the default (**10** without the license).

- **Assured Flow Rate (kbps)**—Set the assured flow rate (AFR) for any connection. AFR is the minimum rate at which a connection is allowed to exist in the system. Connections usually get a bandwidth between this and the **Max-Bandwidth per Session** setting. Default is **0** (zero), which means AFR is disabled so Media Flow Controller delivers at a best-effort but does not assure a particular bit rate per video.

Click **Apply**.

3. Click **Save** at the top of the page to make changes persistent.

Creating a Virtual Player (EZconfig)

To create an un-configured virtual player using EZconfig:

1. Click the **EZconfig** tab.
The **EZconfig** page is displayed, scroll down to **Virtual Player**; see [Figure 14](#) for graphic.

Figure 14 EZ Config Page Virtual Player Area

2. Enter information in the text box:
 - **Virtual Player Name**—Enter a name for the new virtual player.
 - **Virtual Player Type:**
 - **generic**—For caching most Web video content. Options: **assured-flow**, **connection max-bandwidth**, **fast-start**, **full-download**, **hash verification**, and **seek**.
 - **break**—For Break® video delivery (no **full-download** option).
 - **qss-streamlet**—Fine-grained list of delivery rate-maps for assured flow.
 - **yahoo**—For Yahoo® video delivery, includes hash digests and healthcheck probes.
 - **youtube**—For YouTube video delivery
 - **smoothstream-pub**—For Microsoft Smooth Stream function.
 - **flashstream-pub**—For adaptive HTTP streaming support to the Flash platform.

This steps merely creates the virtual player with default values, in many cases that may be Ok for your needs. See [“Configuring Virtual Players \(Web Interface\)” on page 289](#) for more configuration options.

Click **Add**.

3. Click **Save** at the top of the page to make changes persistent.

Adding a Namespace (EZconfig)

To create a namespace using EZconfig:

1. Click the **EZconfig** tab.
The **EZconfig** page is displayed, scroll down to **Add Namespace**; see [Figure 15](#) for graphic.

Figure 15 EZ Config Page Add Namespace Area

2. Enter information in the text boxes:
 - **Namespace**—Give the namespace a relevant name.
 - **URI-Prefix** (required)—The uri-prefix refines what requests Media Flow Controller accepts; see [“uri-prefix” on page 34](#) for usage details.
 - **Domain**—Specify a domain or use * (asterisk) to indicate any domain.
 - **HTTP Origin**— Use HTTP for origin fetch of content not in cache; specify the **Hostname/IP Address** of the origin server.
 - **NFS Origin**—Use NFS for origin fetch of content not in cache; specify the **Hostname/IP Address** of the origin server.
 - **Virtual Player**—The namespace uses this virtual player instead of the default settings of **Network Parameters**.

See [“Configuring NameSpaces \(Web Interface\)” on page 301](#) for more configuration options.

Click **Add**.

3. Click **Save** at the top of the page to make changes persistent.

Enabling Interfaces (EZconfig)

To enable interfaces using EZconfig:

1. Click the **EZconfig** tab.
The **EZconfig** page is displayed, scroll down to **Enable Interfaces**; see [Figure 15](#) for graphic.

| Enable Interfaces | | | | | |
|-------------------|------------|--|---------|--|--|
| eth0 | IP Address | <input type="text" value="172.19.172.66"/> | Netmask | <input type="text" value="255.255.255.0"/> | Enable <input checked="" type="checkbox"/> |
| sit0 | IP Address | <input type="text"/> | Netmask | <input type="text"/> | Enable <input checked="" type="checkbox"/> |

Figure 16 EZ Config Page Enable Interfaces Area

2. Specify IP addresses and netmasks for detected Ethernet interfaces; also, enable or disable interfaces.
Click **Apply**.
3. Click **Save** at the top of the page to make changes persistent.

Restarting Services (EZconfig)

Several services require restart after making changes. Choose a Service Name and click **Restart**.

Tip! When any changes are made to delivery protocols or network settings, you must restart the **mod-delivery** service; use **mod-ftp** for changes made to FTP settings, and **mod-log** for changes to logging settings. The **mod-oom** service (offline origin fetch manager) is provided for debugging purposes only.

| Service Restart | |
|-----------------|---|
| Service Name | <input type="text" value="mod-delivery"/> |

Figure 17 EZ Config Page Service Restart Area

Monitoring Media Flow Controller Statistics (Web Interface)

The **Monitoring** tab gives you quick access to statistics and information about the current system, including bandwidth usage, namespace usage, CPU load, and more. [Figure 18](#) shows the **Monitoring** menu.

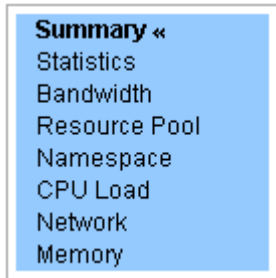


Figure 18 Monitoring tab left navigation menu

Viewing Media Flow Controller Summary

Purpose

View statistics and graphs of the managed Media Flow Controllers; and this information:

- **Summary**
 - Date and Time
 - Hostname
 - Uptime
 - Version
 - Model
 - Host ID (system serial string on the motherboard: your licenses are tied to the Host ID)
 - System Memory (used, free, and total)
 - RAM Cache Size
 - Number of CPUs/Cores
 - CPU load averages.
- **Interface Statistics**—The current TX (transmissions out) and RX (transmissions received) system statistics.

Action

After you log in to Media Flow Controller, select the **Monitoring** tab. The **Summary** page is displayed.

Viewing Media Flow Controller Statistics

Purpose

View counters for:

- Cache Hierarchy
- HTTP Origin
- Virtual Player

- Connections
- HTTP Delivery
- HTTP Tunnel
- Real-Time Streaming Delivery

View statistics **Current Values** for:

- Current Bandwidth (MB/Sec)
- Average Cache Bandwidth (MB/Sec)
- Average Disk Bandwidth (MB/Sec)
- Average Origin Bandwidth (MB/Sec)
- Average HTTP Transaction Rate (per Sec)
- Current Cache Bandwidth (MB/Sec)
- Current Disk Bandwidth (MB/Sec)
- Current Origin Bandwidth (MB/Sec)

View alarm **Current Values**, **Alarm Thresholds**, and **Clear Thresholds** for:

- Free Filesystem (%)
- Average CPU Utilization (%)
- Paging

Action

- From the left navigation pane in the **Monitoring** tab, click **Statistics**. The **Statistics Summary** page is displayed.
- Click **Reset Counter** at the top of the page. All statistics counters are restarted.

Viewing Media Flow Controller Bandwidth Usage

Purpose

Check **Cache Bandwidth**, **Disk Bandwidth**, and **Origin Bandwidth** usage for the last hour. See [Figure 19](#) for graphic.

Action

- From the left navigation pane in the **Monitoring** tab, click **Bandwidth**. The **Bandwidth Usage (Last Hour)** page is displayed.
- Click **Pause** and **Resume** buttons to stop/start graph charting.

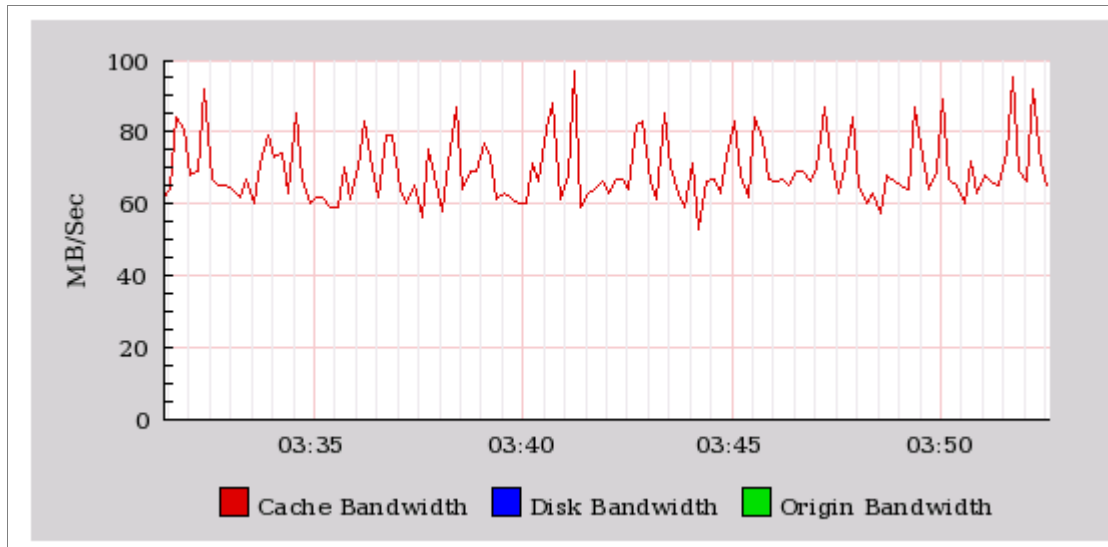


Figure 19 Monitoring > Bandwidth Usage (Last Hour) Chart Example

Viewing Media Flow Controller Namespace Counters

Purpose

List the current configured namespaces and the current number of GET requests for each. See [Figure 20](#) for graphic.

Action

- From the left navigation pane in the **Monitoring** tab, click **Namespace**. The **Namespace Counters** page is displayed.

| Namespace Counters | | | | | | |
|---------------------------|--------------------|---------------|------------------|------------------|------------------|------------------|
| Namespace | Number of Requests | Resource pool | HTTP | | RTSP | |
| | | | Current Sessions | Bandwidth (Mbps) | Current Sessions | Bandwidth (Mbps) |
| mfc_probe | 0 | global_pool | 0 | 0.000 | 0 | 0.000 |
| newTest | 0 | newP | 0 | 0.000 | 0 | 0.000 |

Figure 20 Monitoring > Namespace Counters Page Detail

Viewing Media Flow Controller CPU Load

Purpose

View different graphs of the CPU load in the last hour. See [Figure 21](#) for graphic.

Action

- From the left navigation pane in the **Monitoring** tab, click **CPU Load**. The **CPU Load (Last Hour)** page is displayed.

- Choose from the drop-down menu to view **Aggregated** (default), **Per CPU**, or **Per CPU Stacked** graph.
- Use **Pause** and **Resume** buttons to stop/start graph charting, use **Clear Data** to start new graphing.

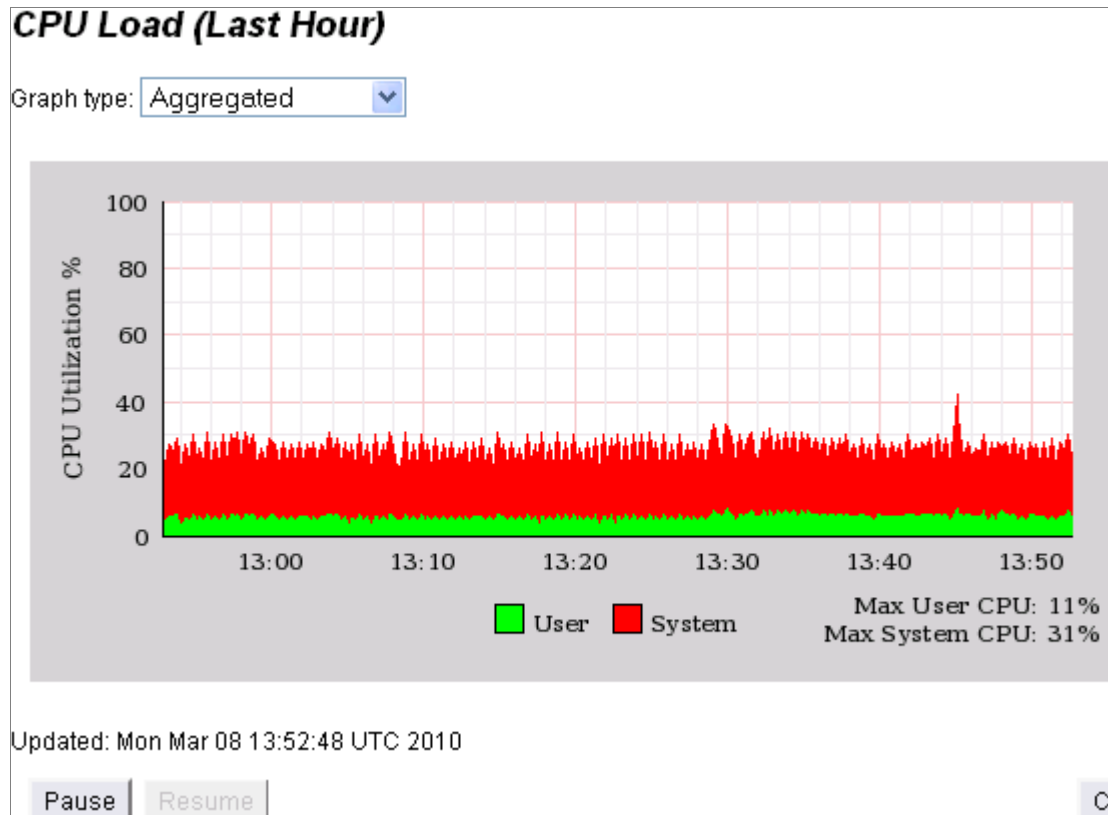


Figure 21 Monitoring > CPU Load Page Detail

Viewing Network Usage

Purpose

A last-hour graph of **Network Usage** including RX and TX information on all data ports. See [Figure 22](#) for graphic. Current statistics include:

- Bytes, packets, discards, errors, and overruns for RX and TX
- RX mcast packets
- RX frame
- TX carrier
- TX collisions

Action

- From the left navigation pane in the **Monitoring** tab, click **Network**. The **Network Usage (Last Hour)** page is displayed.
- Use **Pause** and **Resume** buttons to stop/start graph charting.

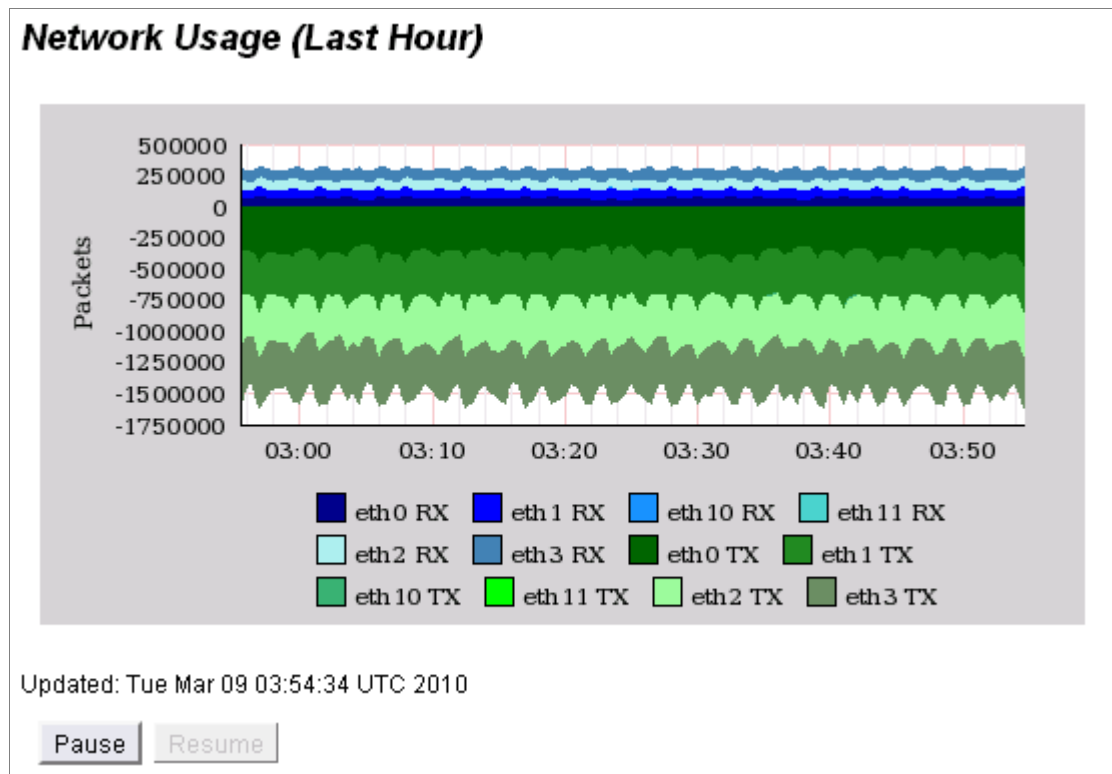


Figure 22 Monitoring > Network Usage (Last Hour) Page Detail

Viewing Memory Utilization

Purpose

A last-day graph of **Memory Utilization** plus a pie chart of **Current Memory Statistics** including statistics of Physical and Swap memory (Total, Used, and Free). See [Figure 23](#) for graphic.

Action

- From the left navigation pane in the **Monitoring** tab, click **Memory**. The **Memory Utilization (Last Day)** page is displayed.
- Use **Pause** and **Resume** buttons to stop/start graph charting.

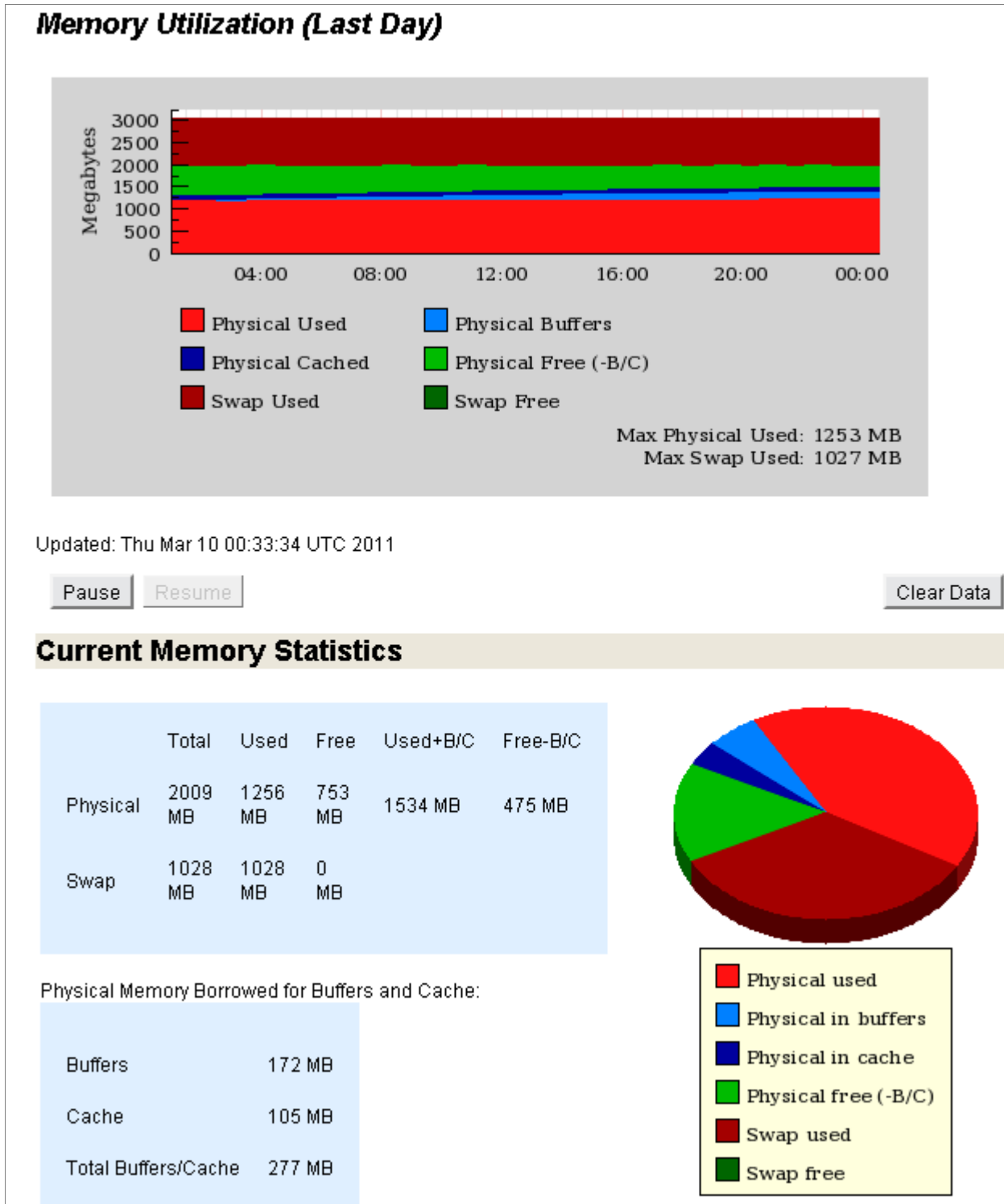


Figure 23 Monitoring > Memory Utilization and Current Memory Statistics Page Detail

System Configuration Overview (Web Interface)

You can configure many system settings for the Juniper Networks Media Flow Controller by using the **System Config** tab. [Figure 24](#) shows the **System Config** menu.

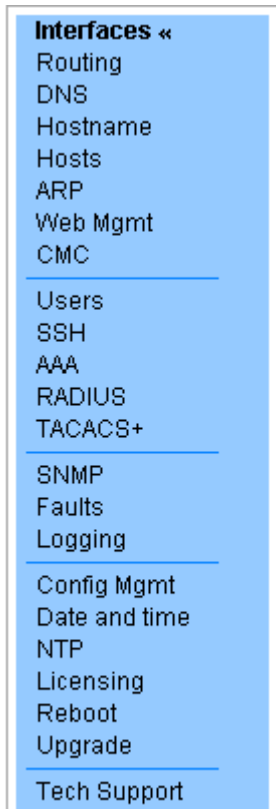


Figure 24 System Config tab left navigation menu

To configure system settings:

- Configure the interfaces.
See [“Configuring Interfaces, Default Gateway, Static Routes, DNS and Domain Names, Hostname, and Banners \(Web Interface\)”](#) on page 245
- Configure routing parameters.
See [“Configuring Static Hosts and ARP \(Web Interface\)”](#) on page 253
- Configure system date, time, and NTP servers.
See [“Configuring Date, Time, and NTP \(Web Interface\)”](#) on page 256
- Configure authentication and authorization options, and users.
See [“Configuring Users and AAA \(Web Interface\)”](#) on page 265
- Configure fault notifications, and logging options.
See [“Configuring Faults and Logging \(Web Interface\)”](#) on page 269

Configuring Interfaces, Default Gateway, Static Routes, DNS and Domain Names, Hostname, and Banners (Web Interface)

Before you configure Media Flow Controller interfaces, see [“Before You Configure Media Flow Controller” on page 82.](#)

Tip! You may want to change the Web default logout time (**900** = 15 minutes); to do this:

- Click the **System Config** tab.
The **Network Interfaces** page is displayed.

Configuring Interfaces (Web Interface)

We recommend using eth0 for management, eth1 for origin fetch, and the other interfaces for traffic. See [“Example: Media Flow Controller Interface Configuration” on page 90.](#) for details. You can also configure the DHCP primary interface and add interface aliases on this page.

To view and set network interfaces:

- From the left navigation pane in the **System Config** tab, select **Interfaces**.
The **Network Interfaces** page is displayed. See [Figure 25](#).

Network Interfaces

eth0 state

| | | | |
|-------------------|-------------------|----------------|-------------------|
| Status | Admin up, link up | Speed | 1000Mb/s (auto) |
| IP address | 172.19.172.67 | Duplex | full (auto) |
| Netmask | 255.255.255.0 | MTU | 1500 |
| Type | ethernet | HW addr | 00:0C:29:BA:64:8D |

eth0 configuration

Enabled

Speed Auto

Obtain IP Address Automatically (DHCP)

Duplex Auto

Specify IP Address Manually

MTU 1500

IP address 172.19.172.67
 Netmask 255.255.255.0

Comment:

Apply
Cancel

Figure 25 Network Interfaces Page Detail (eth0 state and eth0 configuration)

eth0 state

View the state of the eth0 interface, see [Figure 25](#).

To view the **eth0 state** area:

Status (**Admin up**: The interface is enabled, **Link up**: The interface has a current connection), **IP address**, **Netmask**, **Type**, **Speed**, **Duplex**, **MTU**, **HWaddr** (hardware address), and a **Comment** (if configured) for each discovered interface.

eth0 configuration

The eth0 interface is used to manage Media Flow Controller. See [Figure 25](#).



CAUTION: For VXA Series Media Flow Engine appliances, do not ever change the Ethernet name mappings; all interface assignments are handled automatically during manufacturing.

To configure the eth0 interface on the **Network Interfaces** page:

1. Enter information in the text box:
 - **Enabled**—Select to enable the interface for activity, default; or de-select the checkbox to disable the interface, respectively.
 - **Obtain IP Address Automatically (DHCP)**—Allow DHCP to assign the IP address for this Media Flow Controller. OR
 - **Specify IP Address Manually**—Enter the **IP address** and **Netmask** you want for this Media Flow Controller.
 - **Speed** and **Duplex**—Choose **Auto** (default) for the Speed and Duplex to be set automatically based on hardware. Or you can set these options to alternate values in the drop-down menu. We highly recommend that **Speed** and **Duplex** not be changed from the auto-configured defaults.
 - **MTU**—The Maximum Transmission Unit (MTU) sets the largest number of bytes a frame can carry. Default is 1500.
 - **Comment**—A description for the interface.
2. Click **Apply** to immediately apply changes; **Cancel** to revert to the existing configuration.
3. Click **Save** at the top of the page to make changes persistent.

Additional interface **state** areas are displayed for each interface discovered by the system. To configure additional interfaces, click **Configure Interface** in the **state** area for each to open a **configuration** area identical to the **eth0 configuration** area.

DHCP Primary Interface

Set DHCP (dynamic host configuration protocol) primary interface and verify the current primary DHCP interface. See [Figure 26](#) for a graphic.



DHCP Primary Interface

Configured primary

Acting primary (none)

Figure 26 Network Interfaces Page Detail (DHCP Primary Interface)

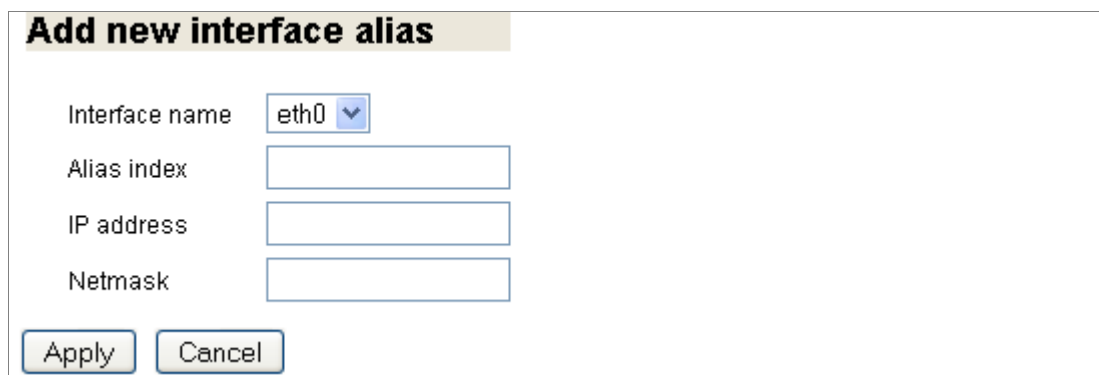
DHCP allows new network devices to be automatically supplied with an IP address and other information, depending on the setup of the DHCP server. Media Flow Controller has no primary DHCP interface by default. Setting a primary interface ensures that DHCP messages arrive only on that interface.

To set the DHCP Primary interface:

1. Choose a **Configured primary** interface from the drop-down list.
2. Click **Apply** to immediately apply changes; **Cancel** to revert to existing configuration.
3. Click **Save** at the top of the page to make changes persistent.

Add New Interface Alias

An interface alias lets you assign multiple IP addresses to the same interface. See [Figure 27](#) for a graphic.



Add new interface alias

Interface name

Alias index

IP address

Netmask

Figure 27 Network Interfaces Page Detail (Add new interface alias)

To add a new interface alias:

1. Choose an existing interface from the drop-down list.
2. Enter information in the text box:
 - **Alias index**—A name for the interface alias. This creates a pseudo interface; its address appears in the output of **show interface** under the primary interface's data
 - IP address
 - **Netmask**
3. Click **Apply** to immediately apply changes; **Cancel** to revert to existing configuration. If you click **Apply**, **state** and **configuration** areas for the new alias are displayed. In the configuration area, you can change disable the interface (interfaces are enabled by default), change the **IP address** and **Netmask**, and add a **Comment**.
4. Click **Save** at the top of the page to make changes persistent.

Setting the Default Gateway and Static Routes (Web Interface)

Set IP routing parameters, including **Default Gateway** and **Static Routes**. The default gateway provides an entry and exit point between the internal network and all external networks; requests with destinations outside of the internal network's routing table are sent to the default gateway for routing.

Static routes, configured paths to a subnet, are used to route data from one subnet to another.

To set network IP routing parameters:

- From the left navigation pane in the System Config tab, select **Routing**. The **IP Routing** page is displayed.

Default Gateway

Set the default gateway as the main access point to external networks, including the Internet.

To set the default gateway:

1. Enter an IP address in the Default gateway text box and click **Set Default Gateway** to immediately apply changes.
2. Click **Save** at the top of the page to make them persistent across reboots. See [Figure 28](#).

The screenshot shows the 'IP Routing' configuration page. The 'Default Gateway' section is highlighted with a light brown background. Below this, there is a text input field labeled 'Default gateway' containing the IP address '172.19.152.1'. Below the input field is a button labeled 'Set Default Gateway'.

Figure 28 IP Routing Page, Default Gateway

Static and Dynamic Routes

A static route is a hard coded (manually defined) path that specifies the route to a certain subnet using a certain path. See [Figure 29](#).

View all configured static and dynamic routes:

- **Destination**—The subnet/path for this static route.
- **Mask**—The netmask for this route.
- **Gateway**—The configured gateway (path to the Internet) for this static route.
- **Interface**—The port configured for this static route.
- **Active**—Whether or not this route is being used currently.
- **Static**—Whether or not this route is static (hard coded).

Select a route and click **Removed Selected** to immediately apply changes; click **Save** at the top of the page to make them persistent across reboots.

| Static and Dynamic Routes | | | | | | |
|---------------------------|--------------|---------------|--------------|-----------|--------|--------|
| | Destination | Mask | Gateway | Interface | Active | Static |
| | 172.19.172.0 | 255.255.255.0 | 0.0.0.0 | eth0 | yes | no |
| <input type="checkbox"/> | default | 0.0.0.0 | 172.19.152.1 | * | no | yes |

Figure 29 IP Routing Page, Static and Dynamic Routes

Add Static Route

Static routes set a path in the routing table for a particular destination. See [Figure 30](#).

To set a static route:

- Enter this information to the text boxes:
 - Destination**—A network prefix for where you want a static route to.
 - Netmask**—The netmask for the configured network prefix.
 - Gateway**—IP address of the configured default gateway.
 - Interface**—Select a configured interface from the drop-down list.
- Click **Add Route** to immediately apply changes.
- Click **Save** at the top of the page to make them persistent across reboots. See [Figure 30](#).

| Add Static Route | |
|------------------|--------------------------------|
| Destination | <input type="text"/> |
| Netmask | <input type="text"/> |
| Gateway | <input type="text"/> |
| Interface | <input type="text" value="v"/> |

Figure 30 IP Routing Page, Add Static Route

Configuring DNS and Domain Names (Web Interface)

View, add, modify, or remove **Static and Dynamic Name Servers** and **Domain Names** from your Domain Name System (DNS).

To configured DNS and domain names:

- From the left navigation pane in the **System Config** tab, select **DNS**. The **DNS** page is displayed.

Static and Dynamic Name Servers

View all configured static and dynamic name servers. See [Figure 31](#).

- **IP Address**—Of the configured name server.
- **Active**—Whether or not this name server is being used currently.
- **Source**—“Configured” means it was manually added; “Dynamic” means it came from a name server.

| DNS | | |
|---------------------------------|--------|------------|
| Static and Dynamic Name Servers | | |
| IP Address | Active | Source |
| 172.19.152.9 | yes | configured |

Figure 31 DNS Page Detail, Static and Dynamic Name Servers

Add or Modify Name Servers

To add or modify name servers from the **DNS** page:

1. You can set up to three dynamic name servers; for each option, enter an IP address.
 - **Primary DNS IP address**—This name server is tried first.
 - **Secondary DNS IP address**—This name server is tried second.
 - **Tertiary DNS IP address**—This name server is tried last.
2. Click **Apply** to immediately apply changes; **Cancel** to revert to existing configuration.
3. Click **Save** at the top of the page to make changes persistent. See [Figure 32](#).

| Add or Modify Name Servers | |
|--|---|
| Primary DNS IP address | <input type="text" value="172.19.152.9"/> |
| Secondary DNS IP address | <input type="text"/> |
| Tertiary DNS IP address | <input type="text"/> |
| <input type="button" value="Apply"/> <input type="button" value="Cancel"/> | |

Figure 32 DNS Page Detail, Add or Modify Name Servers

Static and Dynamic Domain Names

View all configured static and dynamic domain names. See [Figure 33](#).

- **Domain**—The configured name for that domain.
- **Active**—Whether or not this domain name is being used currently.
- **Source**—“Configured” means it was manually added; “Dynamic” means it came from a name server.

| Static and Dynamic Domain Names | | |
|---------------------------------|--------|------------|
| Domain | Active | Source |
| juniper.local | yes | configured |

Figure 33 DNS Page Detail, Static and Dynamic Domain Names

Configured Domain Names

To delete a domain name from the **DNS** page:

1. Select a configured domain name from the list and click **Remove Selected**.
2. Click **Save** at the top of the page to make changes persistent. See [Figure 34](#).

Configured Domain Names

juniper.local

Figure 34 DNS Page Detail, Configured Domain Names

Add New Domain Name

To add a domain name from the **DNS** page:

1. Enter a **Domain Name** and click **Add Domain Name**; it can be removed via the list of **Configured Domain Names** described above.
2. Click **Save** at the top of the page to make changes persistent. See [Figure 35](#).

Add New Domain Name

Domain Name

Figure 35 DNS Page Detail, Add New Domain Name

Setting Hostnames and Banners (Web Interface)

View or change the **System Hostname** and the **DHCP Hostname** (Dynamic Host Configuration Protocol), and set Banners. You can set a **MOTD** (message of the day), a **Login Remote**, and a **Login Local** banner.

To configured the system hostname, the DHCP hostname and options, and banners:

- From the left navigation pane in the **System Config** tab, select **Hostname**. The **Hostname and Banners** page is displayed.

System Hostname

To configure a hostname for the system on the **Hostnames and Banners** page:

1. Enter a name in the **Host Name** text box and click **Apply** to immediately apply changes; **Cancel** to revert to existing configuration.
2. Click **Save** at the top of the page to make changes persistent. See [Figure 36](#).

The screenshot shows the 'Hostname and Banners' configuration page. Under the 'System Hostname' section, there is a text input field labeled 'Host Name' containing the value 'test-vos-cl66'. Below the input field are two buttons: 'Apply' and 'Cancel'.

Figure 36 Hostname and Banners Page Detail, System Hostname

DHCP Hostname

Dynamic Host Configuration Protocol settings. See [Figure 37](#). An interface is only eligible to be the DHCP primary if it is "admin up", has DHCP enabled, and has gotten a DHCP lease. Ineligibility does not prevent an interface from being configured as the DHCP primary interface, but prevents it from actually being used as such. If the configured primary interface is eligible, it is chosen as the acting primary; otherwise, one of the eligible interfaces is chosen (the first, in alphabetical order)

The screenshot shows the 'DHCP Hostname' configuration section. It features a checkbox labeled 'Send hostname with DHCP client request' which is currently unchecked. Below this are two radio button options: 'Use system hostname (currently test-vos-cl66)' which is selected, and 'Alternate hostname for DHCP:' which is unselected. A text input field is provided for the alternate hostname. At the bottom are 'Apply' and 'Cancel' buttons.

Figure 37 Hostname and Banners Page Detail, DHCP Hostname

To configure a hostname for the Dynamic Host Configuration Protocol on the **Hostnames and Banners** page:

1. Select or leave de-selected the **Send hostname with DHCP client request** checkbox. By default, no hostname is sent during DHCP negotiation. The server may use and honor the hostname supplied by the client. By default, the client sends the system's hostname. This may be overridden by entering an **Alternate hostname for DHCP**. This is a global configuration command that affects all DHCP interfaces. Changing this setting forces a renewal of DHCP on all interfaces
2. Select a DHCP hostname to be sent with DHCP client request (if selected), either:
 - **Use system hostname**—Default.

- **Alternate hostname for the DHCP**— Only applies when **Send hostname with DHCP client request** is selected. The hostname may be unqualified or fully qualified. This is a global configuration command that affects all DHCP interfaces. Changing this setting forces a renewal of DHCP on all interfaces.
3. Click **Save** at the top of the page to make changes persistent.

Banners

You can set a **MOTD** (message of the day), a **Login Remote**, and a **Login Local** banner. See [Figure 38](#); only **MOTD** banner area is shown; the **Login Remote** and **Login Local** banner areas are identical to the **MTOD** banner area.

To set a banner for the system on the **Hostnames and Banners** page:

1. Enter a name in the **Host Name** text box and click **Apply** to immediately apply changes; **Cancel** to revert to existing configuration.
2. Click **Save** at the top of the page to make changes persistent.

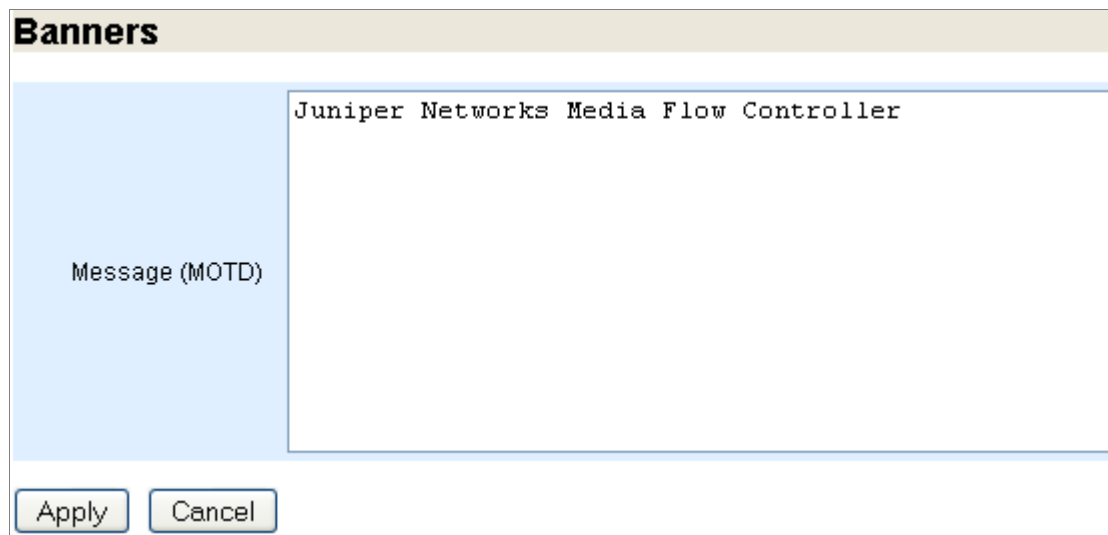


Figure 38 Hostname and Banners Page Detail, Banners (more Banners options at bottom of page)

Configuring Static Hosts and ARP (Web Interface)

Before you configure Media Flow Controller static hosts and ARP, see [“Before You Configure Media Flow Controller” on page 82](#).

Configuring Static Hosts

Set static host entries; a static host is not subject to IP address changes via DNS (dynamic name server).

To set static hosts:

- From the left navigation pane in the **System Config** tab, select **Hosts**. The **Static Hosts** page is displayed.

Static Host Entries

View configured static host entries, including **IP address** and **Hostname**. See [Figure 39](#).

To delete a static host entry on the **Static Hosts** page:

1. Click **Remove Selected** to delete any static host entries you have created.
2. Click **Save** at the top of the page to make changes persistent.

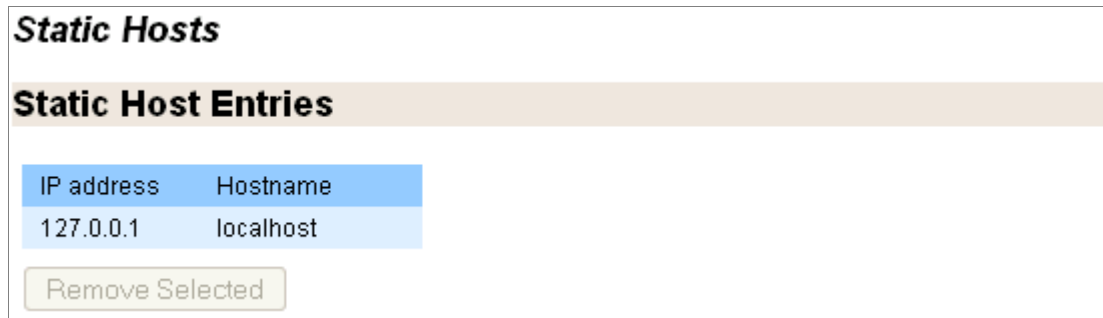


Figure 39 Static Hosts Page Detail, Static Host Entries

Add New Host

You must know the **IP address** and **Hostname** to enter a static host mapping. See [Figure 40](#).

To add a new host on the **Static Hosts** page:

1. Enter an **IP address** and **Hostname**. Click **Add Entry** to immediately apply changes.
2. Click **Save** at the top of the page to make changes persistent.



Figure 40 Static Hosts Page Detail, Add New Host

Configuring ARP (Web Interface)

Manage Address Resolution Protocol (ARP) entries.

To manage ARP entries on the **Static Hosts** page:

- From the left navigation pane in the **System Config** tab, select **ARP**. The **Address Resolution** page is displayed.

Static and Dynamic ARP Entries

View and remove **Static and Dynamic ARP Entries**. See [Figure 41](#).

| Static and Dynamic ARP Entries | | | | |
|--------------------------------|-------------------|-----------|--------|--------|
| IP address | MAC address | Interface | Active | Static |
| 10.5.1.101 | 00:15:17:7D:24:D7 | eth3 | yes | no |
| 10.3.1.101 | 00:15:17:80:13:CB | eth1 | yes | no |
| 172.19.172.52 | 00:1E:C9:AF:38:8D | eth10 | yes | no |
| 10.4.1.101 | 00:15:17:7D:24:D6 | eth2 | yes | no |
| 172.19.172.15 | 00:22:6B:75:C6:00 | eth10 | yes | no |
| 10.1.1.101 | 00:15:17:80:13:CA | eth0 | yes | no |
| 172.19.172.154 | 00:21:70:9F:26:89 | eth10 | yes | no |

Remove Selected

Figure 41 Address Resolution Page Detail, Static and Dynamic ARP Entries

To view and remove static and dynamic ARP entry status from the **Address Resolution** page:

- View this information:
 - IP address**—The configured IP address for this entry.
 - MAC address**—The physical address of this entry.
 - Interface**—The port configured for this entry.
 - Active**—Whether or not this entry is being used currently.
 - Static**—Whether or not this entry comes from DNS.
- Click **Remove Selected** to delete an entry.
- Click **Save** at the top of the page to make changes persistent.

Add Static Entry

Add entries to the ARP cache as a static entry. See [Figure 42](#).

| Add Static Entry | |
|------------------|----------------------|
| IP address | <input type="text"/> |
| MAC address | <input type="text"/> |
| Add Entry | |

Figure 42 Address Resolution Page Detail, Add Static Entry

To add a static ARP entry from the **Address Resolution** page:

- Enter the **IP address** and **MAC address** of the system you want for ARP.
- Click **Add Entry** to immediately apply changes; **Cancel** to revert to existing configuration.
- Click **Save** at the top of the page to make changes persistent.

Clear Dynamic ARP Cache

Click **Clear** to empty the ARP cache. See [Figure 43](#).

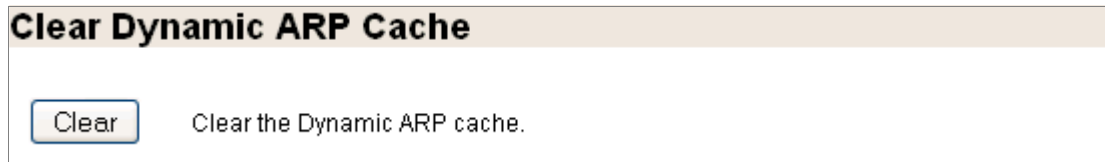


Figure 43 Address Resolution Page Detail, Clear Dynamic ARP Cache

Configuring Date, Time, and NTP (Web Interface)

Before you configure Media Flow Controller date, time, and NTP, see [“Before You Configure Media Flow Controller” on page 82](#).

Configuring the System Date and Time (Web Interface)

Proper time configuration is required for correct caching (validating content, and so forth).

To configure the system date and time:

- From the left navigation pane in the **System Config** tab, select **Date and time**. The **Date and Time** page is displayed. See [Figure 44](#).

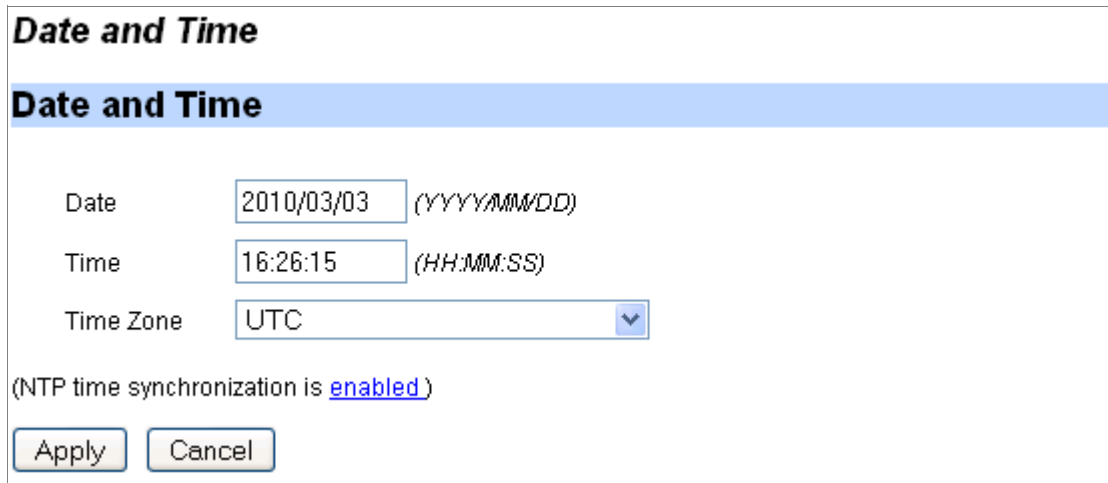


Figure 44 System Config > Date and Time Page Detail (Date and Time)

To configure the system date, time, and timezone:

- Enter a this information:
 - Date
 - Time
- Choose a **Time Zone** from the drop-down list.
- (Optional) Click the enabled link to jump to the **NTP** page.
- Click **Apply** to set the date, time, and time zone; **Cancel** to revert to existing configuration.
- Click **Save** at the top of the page to make changes persistent across reboots.

Configuring NTP (Web Interface)

Configure Network Time Protocol (NTP) options.

To configure NTP servers:

- From the left navigation pane in the **System Config** tab, select **NTP**. The **NTP** page is displayed. See [Figure 45](#).

NTP

NTP Setup

Enable NTP Time Synchronization

Clock is unsynchronized.

NTP Servers

| Server | Status | Stratum | Offset (ms) | Reference Clock | Poll Interval (sec.) | Last response (sec.) | NTP version |
|-----------------|--------|---------|-------------|-----------------|----------------------|----------------------|-------------|
| No NTP servers. | | | | | | | |

Add New NTP Server

Server IP

Version

Enabled

Figure 45 NTP Page

NTP Setup

Set NTP synchronization. See [Figure 46](#).

Figure 46 NTP Page

To enable/disable NTP time synchronization:

1. Select the checkbox to enable NTP; de-select it to disable NTP.
2. Click **Apply** to complete enabling NTP, **Cancel** to revert to existing configuration.
3. Click **Save** at the top of the page to make changes persistent across reboots.

NTP servers

Manage existing NTP servers. See [Figure 47](#).

| Server | Status | Stratum | Offset (ms) | Reference Clock | Poll Interval (sec.) | Last response (sec.) | NTP version |
|-----------------|--------|---------|-------------|-----------------|----------------------|----------------------|-------------|
| No NTP servers. | | | | | | | |

Figure 47 NTP Page

To manage NTP servers.

1. For each configured NTP server view this information:
 - **Server**—IP address of the NTP server.
 - **Status**—Whether or not the server is currently in use.
 - **Stratum**—The hierarchical system this NTP server uses.
 - **Offset (ms)**—Whether or not an offset (a degree in milliseconds of difference) of the server's time is configured.
 - **Reference Clock**—How the NTP server is finding its reference (base) clock; INIT means this is configured in the /etc/init.d directory.
 - **Poll Interval (sec.)**—How often the server exchanges information with its configured reference clocks.
 - **Last response (sec.)**—When the last response from the reference clock was received.
 - **NTP version**—The configured version of NTP.

2. Select a server and **Remove Selected Server**, **Enable Server**, or **Disable Server**.
3. Click **Save** at the top of the page to make changes persistent across reboots.

Add New NTP Server

Add additional NTP servers for redundancy. See [Figure 48](#).

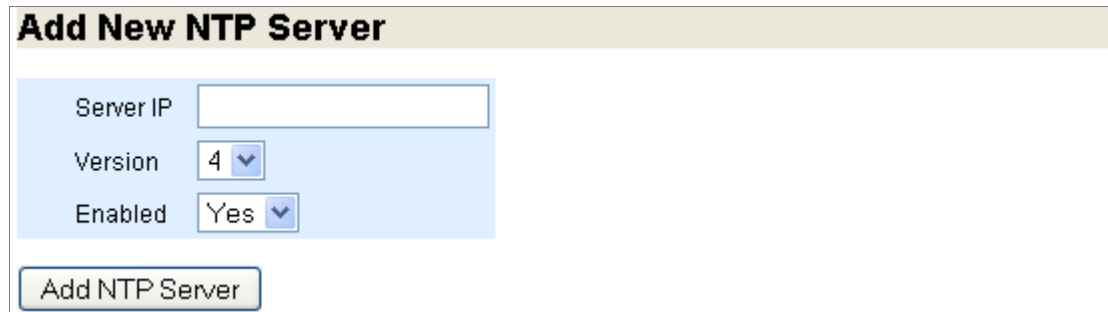


Figure 48 NTP Page Detail, Add NTP Server

To add a new NTP server:

1. Enter a **Server IP** address. Select a **Version** from the drop-down list and **Yes** or **No** from the **Enabled** drop-down list to enable or disable the NTP server. Click **Add NTP Server**.
2. Click **Save** at the top of the page to make changes persistent across reboots.

Configuring RADIUS, TACACS+, and SSH (Web Interface)

Before you configure Media Flow Controller for the first time, see [“Before You Configure Media Flow Controller” on page 82](#).

RADIUS (Remote Authentication Dial In User Service) is a networking protocol that provides centralized access, authorization and accounting management for people or computers to connect and use a network service.

TACACS+ (Terminal Access Controller Access-Control System Plus) is a protocol that provides access control for routers, network access servers, and other network devices via one or more centralized servers. TACACS+ provides separate authentication, authorization and accounting services. TACACS+ servers are tried in the order they are configured.

Secure Shell or SSH is a protocol, using a client-server model, for exchanging data using a secure channel between two network devices. SSH uses public-key cryptography to authenticate the remote computer; the remote computer can also be authenticated (optional). SSH is used to log into remote machines and execute commands, but it also supports tunneling, and SFTP or SCP protocol file transfers. The standard SSH port is TCP port 22.

Configuring RADIUS (Web Interface)

RADIUS (Remote Authentication Dial In User Service) is a networking protocol that provides centralized access, authorization and accounting management for people or computers to connect and use a network service.

To configure RADIUS:

- From the left navigation pane in the **System Config** tab, select **RADIUS**. The **RADIUS** page is displayed.

Default RADIUS Settings

Configure **Default RADIUS Settings**. See [Figure 49](#).

Figure 49 RADIUS Page Detail, Default RADIUS Settings

To configure default RADIUS settings:

1. Enter this information to the text boxes:
 - **Key**—A shared secret text string. If no **key** is set, the user is prompted for the key.
 - **Timeout**—Timeout for retransmitting a request to any RADIUS server. Range is **1-60**, default is **3**.
 - **Retransmit**—The number of times the client attempts to authenticate with any RADIUS server. Range is **0-5**, default is **1**.
 - **Login-lat-group**—The string that identifies the groups that the user is authorized to use when Login-service is defined as LAT (local area transport). If none is set, the user is prompted for the string.
2. Click **Apply** to immediately apply changes; **Cancel** to reset previous values.
3. Click **Save** at the top of the page to make changes persistent across reboots.

RADIUS Servers

In the server list: **Remove, Enable or Disable RADIUS Servers**. See [Figure 50](#).

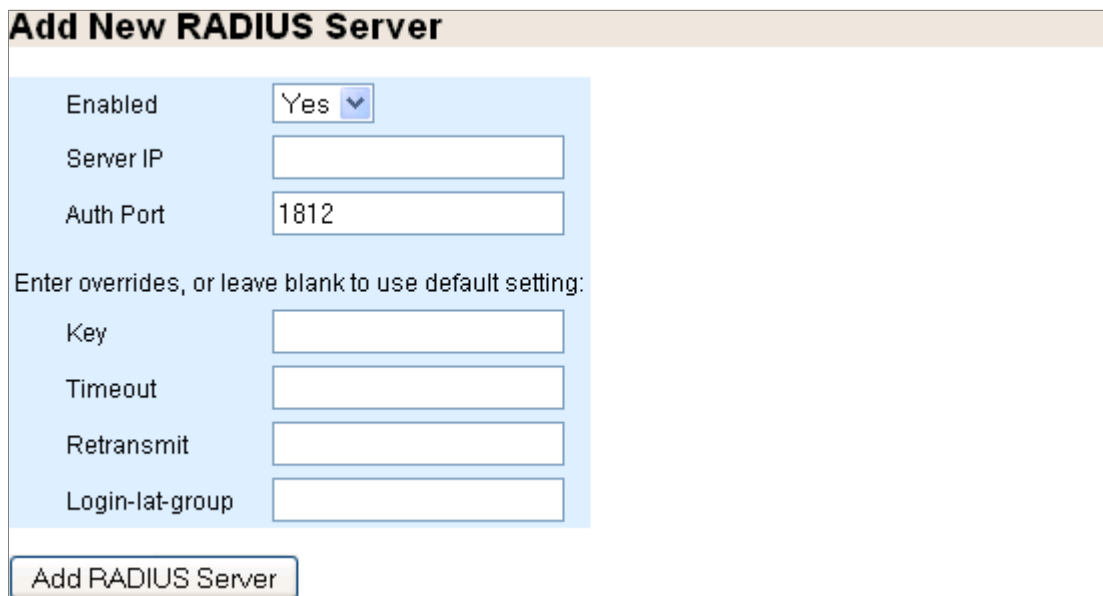
Figure 50 RADIUS Page Detail, RADIUS Servers

1. View this information on configured RADIUS servers:
 - **Server**—The configured IP address for this RADIUS server.
 - **Auth-Port**—The configured port for authentication requests to this server.

- **Key**—The configured shared secret text string. If empty, the user is prompted for the key.
 - **Timeout**—The configured timeout for retransmitting a request.
 - **Retransmit**—The configured number of times a client may attempt to authenticate.
 - **Login-lat-group**—The configured string that identifies the groups that the user is authorized to use when Login-service is defined as LAT (local area transport).
 - **Enabled**—Whether or not this RADIUS server is enabled. Disabling a server makes it inactive but does not delete it from the system.
2. Select a server and **Remove Selected Server**, **Enable Server**, or **Disable Server**.
 3. Click **Save** at the top of the page to make changes persistent across reboots.

Add New RADIUS Server

Add a new RADIUS server. See [Figure 51](#).



The screenshot shows a web form titled "Add New RADIUS Server". The form is set against a light blue background. At the top, there is a dropdown menu for "Enabled" with "Yes" selected. Below this are three text input fields: "Server IP" (empty), "Auth Port" (containing "1812"), and "Key" (empty). A section titled "Enter overrides, or leave blank to use default setting:" contains four more text input fields: "Timeout" (empty), "Retransmit" (empty), and "Login-lat-group" (empty). At the bottom of the form is a button labeled "Add RADIUS Server".

Figure 51 RADIUS Page Detail, Add New RADIUS Server

To add and enable a new RADIUS server:

1. Select **Enabled**—The server must be enabled to do authentication.
2. Enter this information to the text boxes:
 - **Server IP**—IP address for the server.
 - **Auth Port**—The port for authentication requests; default is **1812**. You can use the same IP address in more than one **host** as long as the **auth-port** is different.
3. (Optional) Enter overrides for **Key**, **Timeout**, **Retransmit**, and **Login-lat-group** values for this RADIUS server from the default RADIUS settings you made above.
4. Click **Add RADIUS Server** to complete operation.
5. Click **Save** at the top of the page to make changes persistent across reboots.

Configuring TACACS+ (Web Interface)

Configure TACACS+ authentication options.

To configure TACACS+:

- From the left navigation pane in the **System Config** tab, select **TACACS+**. The **TACACS+** page is displayed.

Default TACACS+ Settings

Configure **Default TACACS+ Settings**. See [Figure 52](#).

TACACS+

Default TACACS+ Settings

Key

Timeout

Retransmit

Figure 52 TACACS+ Page Detail, Default TACACS+ Settings

To configure default TACACS+:

- Enter this information to the text boxes:
 - Key**—A shared secret text string. If no **key** is set, the user is prompted for the key.
 - Timeout**—Timeout for retransmitting a request. Range is **1-60**, default is **3**.
 - Retransmit**—The number of times the client attempts to authenticate with any TACACS+ server. Range is **0-5**, default is **1**.
- Click **Apply** to immediately apply changes; **Cancel** to reset previous values.
- Click **Save** at the top of the page to make changes persistent across reboots.

TACACS+ Servers

In the server list: **Remove**, **Enable** or **Disable TACACS+ Servers**. See [Figure 52](#).

TACACS+ Servers

| Server | Auth-Port | Auth-Type | Key | Timeout | Retransmit | Enabled |
|---------------------|-----------|-----------|-----|---------|------------|---------|
| No TACACS+ servers. | | | | | | |

Figure 53 TACACS+ Page Detail, TACACS+ Servers

- View this information on configured TACACS+ servers:
 - Server**—The configured IP address for this TACACS+ server.
 - Auth-Port**—The configured port for authentication requests to this server.

- **Auth-Type**—The configured type of authentication this TACACS+ server will use.
 - **Key**— The configured shared secret string. If empty, the user is prompted for the key.
 - **Timeout**—The configured timeout for retransmitting a request.
 - **Retransmit**—The configured number of times a client may attempt to authenticate.
 - **Enabled**—Whether or not this TACACS+ server is enabled. Disabling a server makes it inactive but does not delete it from the system.
2. Select a server and **Remove Selected Server**, **Enable Server**, or **Disable Server**.
 3. Click **Save** at the top of the page to make changes persistent across reboots.

Add New TACACS+ Server

Add a new TACACS+ server. See [Figure 52](#).

Figure 54 TACACS+ Page Detail, Add New TACACS+ Server

To add and enable a new TACACS+ server:

1. Select **Enabled**—The server must be enabled to do authentication.
2. Enter this information to the text boxes:
 - **Server IP**—IP address for the server.
 - **Auth Port**—The port for authentication requests; default is **49**. You can use the same IP address in more than one host as long as the **auth-port** is different.
 - **Auth Type**—Which type of authentication this TACACS+ server will use; both authentication types transmit the username and password in un-encrypted text and are acceptable when passwords are stored in an external database. Choose either:
 - **ascii**—American Standard Code for Information Interchange.
 - **pap**—Password authentication protocol (default).
3. (Optional) Enter overrides for **Key**, **Timeout**, **Retransmit**, and **Login-lat-group** values for this TACACS+ server from the default TACACS+ settings you made above.
4. Click **Add TACACS+ Server** to complete operation.
5. Click **Save** at the top of the page to make changes persistent across reboots.

Configuring SSH (Web Interface)

Configure Secure Sockets Shell (SSH) transmissions. SSH is protocol to secure connections through an encryption known to the server and the client. The encryption operates via Host Keys, a secret string configured on the server and communicated to an authenticated client. See [ssh](#) for CLI details.

To configure SSH:

- From the left navigation pane in the **System Config** tab, select **SSH**. The **SSH** page is displayed. See [Figure 55](#).

SSH

SSH Server

Host Keys

| Key Type | Finger Print |
|----------|---|
| RSA1 | 7b:11:ec:69:ad:ed:f7:86:09:6f:03:46:dc:58:59:cc |
| RSA2 | 82:35:42:ed:e1:16:87:0d:9f:76:32:c8:ab:95:b7:55 |
| DSA2 | 05:32:67:66:31:cb:45:26:ad:89:1b:53:5a:30:06:47 |

Generate New Host Keys

Generate Host Keys

Figure 55 SSH Page

SSH Server

View information on the current SSH server Host Keys. See [Figure 55](#).

- Key Type**—Either RSA1 (inventors initials) or DSA2 (Digital Signature Algorithm, 2).
- Finger Print**—A human-readable string so you can check the key manually.

Generate New Host Keys

Generate a new identity (private and public keys) for the logged-in user. See [Figure 55](#).

Click **Generate**.

When the keys are generated, the private key is written to the logged-in user's .ssh directory in an appropriately named file (for example, id_dsa). This identity can be used when the user connects from the system to another host with **slogin**.

Configuring Users and AAA (Web Interface)

Before you configure Media Flow Controller for the first time, see [“Before You Configure Media Flow Controller” on page 82](#).

Before configuring AAA authentication, configure your authentication methods; RADIUS and TACACS+ are supported authentication methods.

Configuring Users (Web Interface)

View, remove, enable, disable, and add new users; plus change existing user passwords.

To manage user accounts:

- From the left navigation pane in the **System Config** tab, select **Users**. The **Users** page is displayed.

Active Users

View configured users information. See [Figure 56](#).

| Active Users | | | | |
|--------------|----------------------|-------|--------------|---------------|
| Username | Full Name | Line | Host | Idle |
| admin | System Administrator | pts/0 | 172.23.1.201 | 0d 0h 14m 14s |
| admin | System Administrator | web/2 | 172.23.1.201 | 0d 0h 0m 0s |

Figure 56 Users Page Detail, Active Users

To view user information:

- **Username**—Used for logins.
- **Full Name**—For the user, as configured.
- **Line**—How the user is connected, SSH or Web.
- **Host**—What system this user is configured on.
- **Idle (seconds)**—Time since the last command execution of this user.

User Accounts

View user account information. See [Figure 57](#).

| User Accounts | | | |
|--|----------------------|--------------|---------|
| User | Full Name | Capability | Enabled |
| admin | System Administrator | admin | yes |
| <input type="checkbox"/> cmcrendv | CMC Rendezvous User | cmcrendv | yes |
| <input type="checkbox"/> ftp | FTP User | ERROR-NO-MAP | yes |
| <input type="checkbox"/> mfc_probe_ftpuser | | ftpuser | yes |
| <input type="checkbox"/> monitor | System Monitor | monitor | yes |

Figure 57 Users Page Detail, User Accounts

For each configured user account:

- View user account information:
 - User**—Used for logins.
 - Full Name**—For the user, as configured.
 - Capability**—The privilege level assigned this user. There are three pre-defined capabilities:
 - admin**—Full privileges (default); in **Enable** mode all **EXEC** commands are available.
 - monitor**—Privileges for reading configuration data (not logs) and performing all actions, but not for changing any configuration.
 - unpriv**—Unprivileged.
 - Enabled**—Whether or not this user account is enabled. User accounts are enabled by default. Disabling an account makes it inactive (logins are disallowed) but does not delete it from the system.
- Select a user account and **Remove Selected User**, **Enable User**, or **Disable User**.
- Click **Save** at the top of the page to make changes persistent across reboots.

Add New User

Add a new user to Media Flow Controller. See [Figure 58](#).

Figure 58 Users Page Detail, Add New User

To add a new user:

1. Enter this information to the text boxes:
 - **User**—Login name.
 - **Full Name**—Displays in User Accounts list.
 - **Capability**— There are three pre-defined capabilities:
 - **admin**—Full privileges (default); in **Enable** mode all **EXEC** commands are available.
 - **monitor**—Privileges for reading configuration data (not logs) and performing all actions, but not for changing any configuration.
 - **unpriv**—Unprivileged.
2. Click **Add User** to complete operation.
3. Click **Save** at the top of the page to make changes persistent across reboots.

Change Password

Change or set a password for each configured user. See [Figure 59](#).

Figure 59 Users Page Detail, admin Password (additional Password areas are displayed for each user)

1. For each user, **admin**, **cmcrendv** (Not Supported), and **monitor** are default users, set or change the password. May be left empty if no password is required.
2. Click **Save** at the top of the page to make changes persistent across reboots.

Configuring AAA (Web Interface)

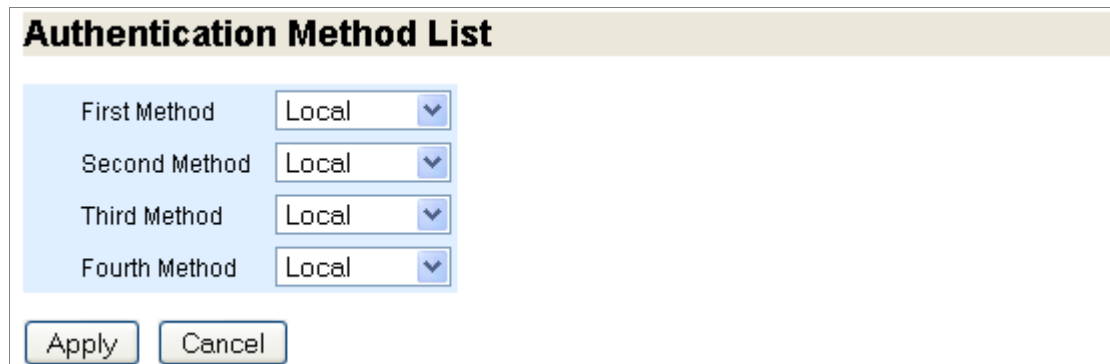
Configure AAA (authentication, authorization and accounting) settings; accounting options are not supported at this time. RADIUS or TACACS+ authentication must be configured before these options can be specified with this command.

To configure AAA authentication:

- From the left navigation pane in the **System Config** tab, select **AAA**. The **AAA Authentication** page is displayed.

Authentication Method List

Set the list of acceptable authentication methods for system logins. See [Figure 60](#).



Authentication Method List

| | |
|---------------|-------|
| First Method | Local |
| Second Method | Local |
| Third Method | Local |
| Fourth Method | Local |

Apply Cancel


Figure 60 AAA Page Detail, Authentication Method List

To set authentication method order:

- Choose from the drop-down list for First Method, Second Method, and Third Method, authentication methods. The order in which the methods are specified is the order in which they are attempted. The authentication methods must be configured.
- Click **Apply** to complete operation; **Cancel** to revert to existing configuration.
- Click **Save** at the top of the page to make changes persistent across reboots.

Authorization

Set authorization options. See [Figure 61](#).



Authorization

| | |
|------------------|--------------|
| Map Order | remote-first |
| Map Default User | admin |

Apply Cancel

Figure 61 AAA Page Detail, Authorization

To set authorization:

1. Choose a **Map Order**— How the remote user mapping behaves when authenticating users via RADIUS or TACACS+. If the authenticated user name is valid locally, no mapping is performed. Options:
 - **remote-first**— If a local-user mapping attribute is returned and is a valid local user name, map the authenticated user to the local user specified in the attribute. Otherwise, if the attribute is not present or not valid locally, use the user specified as the **default-user**.
 - **remote-only** — Only try to map a remote authenticated user if the authentication server sends a local-user mapping attribute; otherwise, no further mapping is tried.
 - **local-only** — All remote users are mapped to the user specified by **Map Default User**. Any vendor attributes received by an authentication server are ignored.
2. Choose a **Map Default User**—What local account a non-local user authenticated via RADIUS or TACACS+ is logged on as; you must select a local and enabled user. This mapping is used depending on the setting of **Map Order**. Click **Apply** to complete operation; **Cancel** to revert to existing configuration.
3. Click **Save** at the top of the page to make changes persistent across reboots.

Configuring Faults and Logging (Web Interface)

Before you configure Media Flow Controller for the first time, see [“Before You Configure Media Flow Controller” on page 82](#).

Configuring Fault Reporting (Web Interface)

Configure Fault Reporting (event notification) options. See [email](#) for CLI details.

To configure Fault Reporting:

- From the left navigation pane in the **System Config** tab, select **Faults**. The **Faults** page is displayed.

Fault Reporting

Set SMTP server, Domain name overrides, Return address, and other options. See [Figure 62](#).

| Fault Reporting | |
|----------------------------------|---------------------------------------|
| SMTP server | <input type="text"/> |
| Domain name override | <input type="text"/> |
| Return address | do-not-reply |
| Include hostname in return addr | <input checked="" type="checkbox"/> |
| Enable autosupport notifications | <input checked="" type="checkbox"/> |
| Enable SMTP authentication | <input type="checkbox"/> |
| Current Reply Address | do-not-reply@CMBU-CLI30.juniper.local |

Apply Cancel

Figure 62 Faults Page Detail, Fault Reporting

To set fault reporting:

- Enter this information to the text boxes:
 - SMTP server**—Use a **hostname** or **IP address** to set the mail relay (**mailhub** in the CLI) to use to send notification e-mails.
 - Domain name override**—Use a **hostname** or **IP address** to set the domain name from which e-mails are to appear to come (provided that the return address is not already fully-qualified). This is used in conjunction with the system hostname to form the full name of the host from which the e-mail appears to come. The rules are as follows:
 - If an e-mail domain is specified using this command, it is always used. If the **hostname** has any dots in it, everything to the right of the first dot is stripped and the e-mail domain is appended.
 - Otherwise, if the **hostname** has dots in it, it is used as is.
If not set, the currently-active system domain name is used. This can come either from the resolver configuration, or from state dynamically instantiated by DHCP.
 - Return address**—Set the username or fully-qualified return address from which e-mail notifications are sent. If the string provided contains an at (@) sign, it is considered fully-qualified and is used as-is. Otherwise, it is considered just the username, and Media Flow Controller appends **@<hostname>.<domain>**. The default is **do-not-reply**, but this can be changed to **admin** or as desired in case something along the line does not like fictitious addresses.
- Select or de-select these options:
 - Include hostname in return addr**—Include (or do not include by un-checking) the hostname in the return address for e-mail notifications. This only takes effect if the return address does not contain an at (@) sign.

- **Enable autosupport notifications**—Enable or disable (by un-checking) the sending of e-mail to vendor autosupport when certain failures occur.
 - **Enable SMTP authentication**—Enable (by checking) or disable SMTP authentication for fault reporting. If enabled, also enter information to these text boxes:
 - **Username**—Set a username for SMTP authentication of e-mails.
 - **Password**—Set a password for SMTP authentication of e-mails; if no password is set, the user is prompted for the password. As of Release 2.0.7 the only authentication method supported is "LOGIN", which sends the password in the clear (base64); so users should be aware that this involves some security risk.
3. Click **Apply** to complete Fault notification configuration, **Cancel** to revert to existing configuration.
 4. Click **Save** at the top of the page to make changes persistent across reboots.

Notify Recipients

View and delete recipients for stats alarms. See [Figure 63](#). See also [email class Options](#).

| Notify Recipients | | | |
|--|--------|-------|----------|
| Email | Detail | Infos | Failures |
| No Recipients. | | | |
| <input type="button" value="Remove Recipients"/> | | | |

Figure 63 Faults Page Detail, Notify Recipients

1. View this information on configured notify recipients:
 - **Email**—Address of configured recipient.
 - **Detail**—Whether or not this recipient is sent detailed or summarized e-mails. Each e-mail potentially has both a detailed and summarized form, where the detailed form has a superset of the information.
 - **Infos**—Whether or not this recipient receives **Info** class e-mails.
 - **Failures**—Whether or not this recipient receives **Failure** class e-mails.
2. To delete a recipient, select a notify recipient and **Remove Recipients**.
3. Click **Save** at the top of the page to make changes persistent across reboots.

Add New Notify Recipients

Add recipients for e-mail notifications of fault events. See [Figure 64](#).

Add New Notify Recipients

Email addresses for event notification:

Email Address

Get Detail

Get Infos

Get Failures

Figure 64 Faults Page Detail, Add New Notify Recipients

To add a notify recipient:

1. Enter this information to the text boxes:
 - **Email Address**—Address of recipient.
2. Select or de-select these options:
 - **Get Detail**—Choose to send this recipient detailed or summarized e-mails.
 - **Infos**—Choose to send this recipient **Info** class e-mails.
 - **Failures**—Choose to send this recipient **Failure** class e-mails.
3. Click **Add Recipient** to complete adding the new notify recipient.
4. Click **Save** at the top of the page to make changes persistent across reboots.

Configuring System Logging (Web Interface)

Configure logging options. See [“Configuring Media Flow Controller System Log” on page 198](#) for more information.

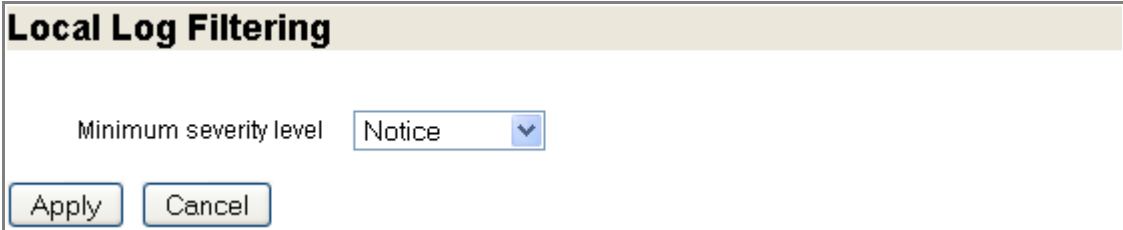
These settings configure the system log (syslog) that records all system activity such as user logins, configuration changes, and system condition changes. It does not record service activity or errors. The Media Flow Controller errorlog records service related errors but is mostly useful for debugging by Juniper Networks Support. Media Flow Controller provides several service-specific logs, detailed in [“Chapter 10, “Troubleshooting Media Flow Controller.”](#)

To configure logging options:

- From the left navigation pane in the **System Config** tab, select **Logging**. The **Logging** page is displayed.

Local Log Filtering

Set log filters. See [Figure 65](#).



Local Log Filtering

Minimum severity level

Figure 65 Logging Page Detail, Local Log Filtering

To set log filtering:

1. Choose a **Minimum severity level**:
 - **None**—Media Flow Controller does not log anything from this class.
 - **Emerg**—System is unusable or cannot recover.
 - **Alert**—Action must be taken immediately for functioning to continue.
 - **Critical**—An unexpected error-causing condition or response for unknown reasons.
 - **Error**—Error conditions.
 - **Warning**—An anomalous condition that can be ignored and functioning continue, but may affect operations.
 - **Notice**—Normal but significant condition or response that could affect operations (default).
 - **Info**—Normal but significant condition or response that does not affect operations.
 - **Debug**—Messages generated by the system debugging utility.
2. Click **Apply** to complete Log filtering configuration, **Cancel** to revert to existing configuration.
3. Click **Save** at the top of the page to make changes persistent across reboots.

Local Log Rotation

Set log rotation parameters; this is especially valuable if this Media Flow Controller will be managed by Central Management Console. See [Figure 66](#).

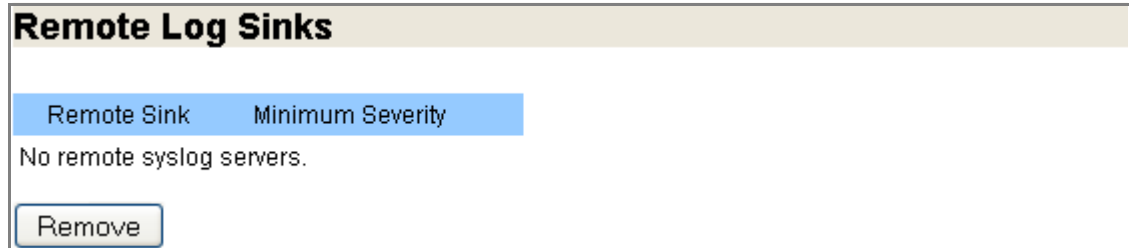
Figure 66 Logging Page Detail, Local Log Rotation

To set local log rotation options:

- Select or de-select these options:
 - Rotate every**—**Day** (at midnight), **Week** (first day, at midnight), or **Month** (first day, at midnight).
 - Rotate when log reaches**—A certain size. The file size is checked hourly, so if it passes the threshold in the middle of the hour it is not rotated right away.
 - Rotate when log reaches**—A percentage of storage (/var) space. The var size is checked hourly, so if it passes the threshold in the middle of the hour it is not rotated right away.
- Enter this information:
 - Keep at most <n> log files**—How many logs to maintain on the system. If the number of log files exceeds this number (at rotation time, or when this setting is lowered), the system deletes as many as necessary, starting with the oldest, to bring it down to this number.
- Click **Apply** to complete Log rotation configuration, **Cancel** to revert to existing configuration, **Force Rotation** to immediately generate a syslog file and start logging over.
- Click **Save** at the top of the page to make changes persistent across reboots.

Remote Log Sinks

View and delete configured log sinks (remote servers receiving log messages from this system). See [Figure 67](#).



| Remote Sink | Minimum Severity |
|---------------------------|------------------|
| No remote syslog servers. | |

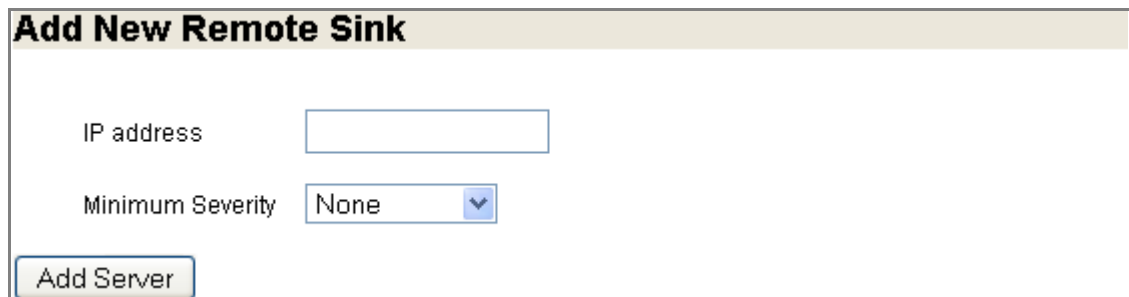
Remove

Figure 67 Logging Page Detail, Remote Log Sinks

1. View this information:
 - **Remote Sink**—Address of configured Remote Sink.
 - **Minimum Severity**—The configured log severity level for this Remote Sink.
2. To delete a log sink, select a log sink and click **Remove**.
3. Click **Save** at the top of the page to make changes persistent across reboots.

Add New Remote Sinks

Add new remote sinks (remote servers receiving log messages from this system). See [Figure 68](#).



Add New Remote Sink

IP address

Minimum Severity

Add Server

Figure 68 Logging Page Detail, Add New Remote Sink

To add a new remote sink:

1. Enter an **IP address**.
2. Choose a **Minimum Severity** level (described in [“Local Log Filtering” on page 273](#)).
3. Click **Apply** to add the new remote sink, **Cancel** to revert to existing configuration.
4. Click **Save** at the top of the page to make changes persistent across reboots.

Log Format

Configure a log format. See [Figure 69](#).



The screenshot shows a web interface for configuring the log format. The title is "Log Format". Below the title, there is a label "Log format" and a dropdown menu with "Standard" selected. At the bottom of the form, there are two buttons: "Apply" and "Cancel".

Figure 69 Logging Page Detail, Log Format

To set a log format:

1. Choose either **Standard** (default) or **WELF** (Web trends Enhanced Log Format). If you choose **WELF**, a **WELF firewall name** option is displayed; specify the firewall name that should be associated with each message logged in WELF format. If no firewall name is set, the hostname is used by default.
2. Click **Apply** to set the format, **Cancel** to revert to existing configuration.
3. Click **Save** at the top of the page to make changes persistent across reboots.

Administering Media Flow Controller Overview

You can perform standard Media Flow Controller administrative tasks using the Web interface, also known as the Management Console.

- [Managing Configuration Files \(Web Interface\)](#)
- [Installing Licenses \(Web Interface\)](#)
- [Restarting Services](#)
- [Upgrading the System \(Web Interface\)](#)
- [Rebooting the System \(Web Interface\)](#)
- [Configuring the Web Interface \(Web Interface\)](#)
- [Configuring the Web Interface Proxy \(Web Interface\)](#)

Managing Configuration Files (Web Interface)

The system can store one or more configuration files on persistent storage with one of the files is designated as **active**: the file that configuration is loaded from on boot, and to which configuration is saved upon a save request. Configuration changes are immediately applied to the running configuration, but are not made persistent until they are explicitly saved.

To manage configuration files:

- From the left navigation pane in the **System Config** tab, select **Config Mgmt**. The **Configurations** page is displayed.

Configuration Files

Choose a configuration and take action. See [Figure 70](#).

The screenshot shows the 'Configurations' page with a sub-section titled 'Configuration Files'. It features a table with two rows of configuration files. The first row is highlighted in light blue and contains a checkbox, the filename 'initial', and a link '(not saved: view running config)'. The second row is highlighted in light green and contains a checkbox and the filename 'initial.bak'. Below the table are three buttons: 'Delete', 'Switch To', and 'Download', each with a corresponding description of its function.

| Filename |
|--|
| <input type="checkbox"/> initial (not saved: view running config) |
| <input type="checkbox"/> initial.bak |

Delete the selected configuration(s).

Make the selected configuration active and apply it to the system. (Select only one)

Download the selected configuration as a binary file. (Select only one)

Figure 70 System Config > Configurations Page Detail, Configuration Files

To manage configuration files from the **Configurations** page:

- View this information for each available configuration file:
 - Filename**—Name of the configuration.
 - Active**—Whether or not the configuration is currently in use.
- Select a file using the checkboxes, and:
 - Click the filename link to open a printout of that configuration file.
 - Click the **view running config** link to open a printout of the running configuration.
 - Click **Delete** to remove from the system the selected configuration file.
 - Click **Switch-To** to make the selected configuration active.
 - Click **Download** to download the selected configuration as a binary file; you are given the option of opening the file or saving it.
- Click **Save** at the top of the page to make changes persistent across reboots.

Active Configuration

For the current, active configuration take various actions. See [Figure 71](#).

Active Configuration

Save Save the running configuration to the active configuration file.

Revert Discard the running configuration and apply the contents of the active configuration.

Reset Reset both the running and active saved configuration to factory defaults, preserving licenses, host keys, and configuration necessary for network connectivity (interfaces, routes, and ARP).

Save As Save the running configuration to a new file and make it active.

New filename:

Figure 71 System Config > Configurations Page Detail, Active Configuration

To manage the active configuration from the **Configurations** page:

- Use the action buttons:
 - Click **Save** to save the active configuration file
 - Click **Revert** to discard the running configuration and apply the active configuration.
 - Click **Reset** to reset both running and active configuration files to the factory defaults.
 - Click **Save As** to save the active configuration as a new file; enter a name in the **New filename** text box.
- Click **Save** at the top of the page to make changes persistent across reboots.

Upload Configuration File

Upload your configuration as local binary file or a local text file from the **Configurations** page. See [Figure 72](#).

Upload Configuration

Upload local binary file:
(To be saved as separate file with its original name)

Upload local text file (CLI commands):
(To be executed immediately in the running configuration)

Figure 72 System Config > Configurations Page Detail, Upload Configuration

Execute CLI Commands

Use this text box to enter CLI commands, each on a separate line, to be executed ad-hoc. End with the **write memory** command. When done, click **Execute CLI commands**.

Import Configuration

Retrieve a configuration from a remote system. See [Figure 73](#).

Figure 73 System Config > Configurations Page Detail, Import Configuration

To import a configuration:

- Enter this information to the text boxes:
 - Hostname or IP address**—The location of the desired configuration.
 - Remote Username**—Login username.
 - Remote Password**—Login password for the set **Remote Username**.
 - Remote Config Name**—Filename of the desired configuration.
 - New Config Name**—A new name for the imported configuration (optional).
 - Import shared data only**—Uncheck to import all nodes, even those not available on this system.
- Click **Import Configuration**.

Installing Licenses (Web Interface)

Use this page to view and remove installed licenses, and add new licenses.

To manage licensing:

- From the left navigation pane in the **System Config** tab, select **Licensing**. The **Licensing** page is displayed.

Licensing

Installed Licenses

| License | |
|--------------------------|--|
| <input type="checkbox"/> | LK2-MFD-413E-8B42-3EH2-4381-GLUL-R4D8-1GT1-KXRJ-BK52-FYLX-RLVC-WR64-3YBK-R |
| Feature | MFD |
| Valid | yes |
| Start date | 2009/12/25 |
| End date | 2010/09/30 |
| Tied to MAC addr | 00:0C:29:BA:64:8D |
| Active | yes |

Figure 74 System Config > License Page Detail, Installed Licenses

Installed Licenses

View and delete licenses. See [Figure 74](#).

To delete a license from the **Licensing** page:

1. Select a license and click **Remove**.
2. Click **Save** at the top of the page to make changes persistent across reboots.

Add New Licenses

Use this area to manually enter a license key. See [Figure 75](#).

Add New License(s)

Please enter one or more licenses, each on a separate line.

Figure 75 System Config > License Page Detail, Add New Licenses

To manually add licenses from the **Licensing** page:

1. Enter one or more licenses on separate lines into the text box and click **Add Licenses**.
2. Click **Save** at the top of the page to make changes persistent across reboots.

Restarting Services

Several services require restart after making changes.

When any changes are made to delivery protocols or network settings, you must restart the **mod-delivery** service; use **mod-ftp** for changes made to FTP settings, and **mod-log** for changes to logging settings. The **mod-oom** service (offline origin fetch manager) is provided for debugging purposes only.



The screenshot shows a web interface titled "Restart Services". It features a dropdown menu labeled "Service Name" with "mod-delivery" selected. Below the dropdown is a button labeled "Restart".

Figure 76 System Config > Restart Services Page

To restart services:

- From the left navigation pane in the **System Config** tab, select **Restart Services**. The **Restart Services** page is displayed.

Choose a **Service Name** and click **Restart**.

To restart services:

- From the **EZconfig** page .

Scroll to the Choose a **Service Name** and click **Restart**.

Upgrading the System (Web Interface)

Use this page to install upgrades.

To install upgrades:

- From the left navigation pane in the **System Config** tab, select **Upgrade**. The **Upgrades and Imaging** page is displayed.

Installed Images

Manage installed images. See [Figure 77](#).

Upgrades and Imaging

Installed Images

Partition 1 (currently booted) (to boot next)
mfc-2.1.0-qa 323_16505_247

Partition 2
mfc-2.1.0-qa 323_16505_247

Switch Boot Partition

Figure 77 System Config > Upgrade Page Detail, Installed Images

To manage installed images from the **Upgrades** page:

1. View the images installed on the two boot partitions.
2. Click **Switch Boot Partition** if the image you want to install is in the other partition.
3. Click **Save** at the top of the page to make changes persistent across reboots.

Install New Image

To set partition; use **Switch Boot Partition**, above, if needed. See [Figure 78](#).

Install New Image to Partition 2

Install from URL:

Install via scp (pseudo-URL format: scp://username@hostname/path/image.img):
URL:
Password:

Install from local file:
(Progress tracking begins after file is uploaded)

Installation options:

View image upgrade progress

Image validation:

To activate a newly-installed software image, please [reboot](#) the system.

Figure 78 System Config > Upgrade Page Detail, Install New Image

To install a new image from the **Upgrades** page:

1. Select the appropriate radio buttons and enter this information, respectively:
 - **Install from URL**—Enter the URL and file path of the install image.
 - **Install via SCP**—Enter the **URL** and file path of the install image and a **Password** allowing access.
 - **Install from local file**—Use the **Browse** button to locate the file on your local system.
2. Select installation options:
 - **View image upgrade process**— When this is checked, you get a progress bar and status messages of the upgrade process.
 - **Image validation**, choose from the drop-down list:
 - Validate if signature is present
 - Require signature and validate
 - Ignore signature
3. Click **Install Image**. A progress bar is displayed showing the image install.
4. To complete the upgrade, go to the **Reboot** page and reboot the system.
5. Click **Save** at the top of the page to make changes persistent across reboots.

Rebooting the System (Web Interface)

A reboot brings up the same configuration that was last active.

To reboot or shut down the appliance:

- From the left navigation pane in the **System Config** tab, select **Reboot**. The **System Reboot or Shutdown** page is displayed.

Reboot or Shutdown

Rebooting the system brings up the last active configuration. See [Figure 79](#).

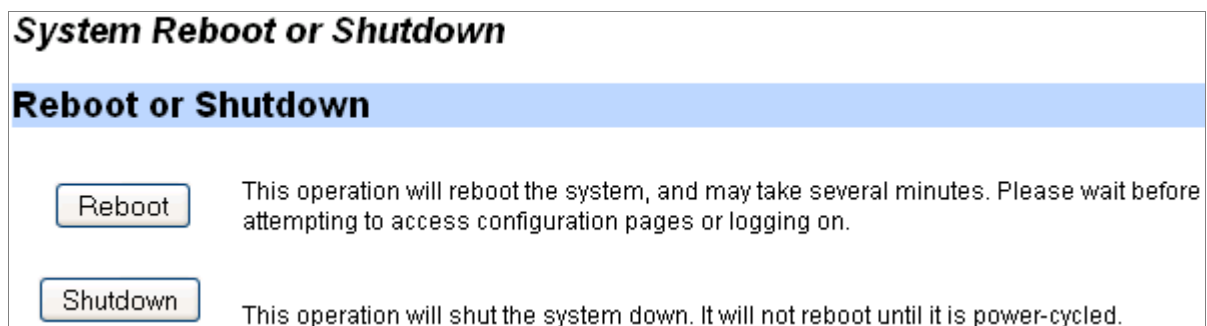


Figure 79 System Config > Reboot Page Detail

To reboot or shut down the system:

1. Click **Reboot** to bring up the last active configuration.
2. Click **Shutdown** to turn the system off.

Configuring the Web Interface (Web Interface)

Configure Media Flow Controller Web interface options. Before you configure the Media Flow Controller Web interface, see [“Before You Configure Media Flow Controller” on page 82](#).

To configure the Web interface Management Console:

- From the left navigation pane in the **System Config** tab, select **Web Mgmt**. The **Web Settings** page is displayed. See [Figure 80](#).

Web Settings

Web UI Configuration

| | |
|--------------------------|-------------------------------------|
| Enable Web Configuration | <input checked="" type="checkbox"/> |
| Auto Logout Timeout | 15.0 minutes |
| Enable HTTP | <input checked="" type="checkbox"/> |
| HTTP Port | 8080 |
| Redirect HTTP to HTTPS | <input type="checkbox"/> |
| Enable HTTPS | <input checked="" type="checkbox"/> |
| HTTPS Port | 443 |
| Web Session Renewal | 30.0 minutes |
| Web Session Timeout | 150.0 minutes |

Apply Cancel Generate New HTTPS Certificate

Figure 80 Web Settings Page Detail, Web UI Configuration

To set parameters for the Media Flow Controller Web interface:

- Enter this information to the text boxes:
 - Enable Web Configuration**—Allow configurations via the Web interface.
 - Auto Logout Timeout**—Control the length of user inactivity required before the Web interface automatically logs out a user.
 - Enable HTTP** and set an **HTTP Port**, de-select to disable HTTP.
 - Enable HTTPS** and set an **HTTPS Port**, de-select to disable HTTPS.
 - Web Session Renewal**—Control the length of time before Web session cookies are automatically regenerated.
 - Web Session Timeout**—Configure time after which a session expires.
- Click **Apply** to complete operation; **Cancel** to revert to existing configuration. You can also **Generate New HTTPS Certificate** by clicking that button.
- Click **Save** at the top of the page to make changes persistent across reboots.

Configuring the Web Interface Proxy (Web Interface)

Configure the Web interface proxy, if needed. See [Figure 81](#).

Web Proxy Configuration

| | |
|---------------------|--------------------------------------|
| Web Proxy address | <input type="text" value="0.0.0.0"/> |
| Web Proxy port | <input type="text" value="1080"/> |
| Authentication type | <input type="text" value="None"/> |
| Basic auth username | <input type="text"/> |
| Basic auth password | <input type="password"/> |

Figure 81 Web Settings Page Detail, Web Proxy Configuration

To set parameters for the Media Flow Controller Web interface when proxied:

- Enter this information to the text boxes:
 - Web Proxy address**—Specify a proxy to be used for any HTTP or FTP downloads.
 - Web Proxy port**—If no port is specified, the default is **1080**.
 - Authentication type**—Configure the type of authentication to be used with a Web proxy; either **None** or **Basic**.
 - Basic auth username**—If you are authenticating with Basic HTTP authentication, enter a username.
 - Basic auth password**—If you are authenticating with Basic HTTP authentication, enter a password for your **Basic auth username**.
- Click **Apply** to complete operation; **Cancel** to revert to existing configuration.
- Click **Save** at the top of the page to make changes persistent across reboots.

Service Configurations Overview

You can configure most Media Flow Controller service settings using the **Service Config** tab. [Figure 82](#) shows the **Service Config** menu.



Figure 82 Service Config tab left navigation menu

To configure system settings:

- Configure global network connection options.
See [“Configuring Network Connections \(Web Interface\)” on page 286](#)
- Configure global delivery protocol options.
See [“Configuring Delivery Protocols \(Web Interface\)” on page 287](#)
- Configure virtual players to assist with various client players.
See [“Configuring Virtual Players \(Web Interface\)” on page 289](#)
- Configure namespaces to determine how to handle incoming requests.
See [“Configuring NameSpaces \(Web Interface\)” on page 301](#)
- Configure media caching options.
See [“Managing the Media-Cache \(Web Interface\)” on page 308](#)
- Configure service logging options.
See [“Configuring Service Logging \(Web Interface\)” on page 309](#)

Configuring Network Connections (Web Interface)

Set global network connection parameters; they may be overridden by a defined virtual player.

To configure delivery network options:

- Click the **Service Config** tab
The **Network Configuration** page is displayed. See [Figure 83](#).

| Network Connection | |
|--------------------------------------|-----------------------------------|
| Connection Idle Timeout(secs) | <input type="text" value="60"/> |
| Connection Max-Bandwidth(kbps) | <input type="text" value="0"/> |
| Connection Concurrent Sessions | <input type="text" value="5000"/> |
| Connection Assured Flow Rate(kbps) | <input type="text" value="0"/> |
| <input type="button" value="Apply"/> | |

Figure 83 Delivery Network Page

Network connection options can be overridden by a virtual player. See [Figure 83](#).

To configure network connection options:

1. Enter this information to the text boxes:
 - **Connection Idle Timeout (sec)**—The socket idle time out in seconds; this is the time the network waits before closing the connection due to data inactivity. This applies to the client (Media Flow Controller inbound) and the origin server (Media Flow Controller outbound) connections.
 - **Connection Max-Bandwidth (kbps)**—Set the maximum bandwidth for a session. The actual session bandwidth is between the **network connection assured-flow-rate** and this value. Even if there is available bandwidth in the link, Media Flow Controller does not allocate more than this value for a session. When there is a full

download, Media Flow Controller tries to allocate this value to the session. You must have the Media Flow Controller license installed to change the default (**200** kbps without the license).

- **Connection Concurrent Sessions**—Set a limit on concurrent sessions in Media Flow Controller. Default is **64000**; maximum allowed is **250,000**. You must have the Media Flow Controller license installed to change the default (**10** without the license).
 - **Connection Assured Flow Rate (kbps)**—Set the assured flow rate (AFR) for any connection. AFR is the minimum rate at which a connection is allowed to exist in the system. Connections usually get a bandwidth between this and the **connection max-bandwidth** setting. Default is **0** (zero), means Media Flow Controller best effort. AFR is disabled by default; if needed, you can enable it with these configurations or with a **virtual-player** configuration. For more information, see [“Using Network Connection Assured Flow” on page 100](#).
2. Click **Apply**.
 3. Click **Save** at the top of the page to make changes persistent across reboots.

Configuring Delivery Protocols (Web Interface)

Set delivery options for delivering content to the player (end-user/consumer). If not specified, default actions take place in the delivery path. The listen interfaces are the ports on the Media Flow Controller that receive and deliver media. These ports typically have Internet access, and should be connected with highest-quality cables. After configured, Media Flow Controller accepts traffic on those interfaces only. Up to 10 interfaces can be specified.

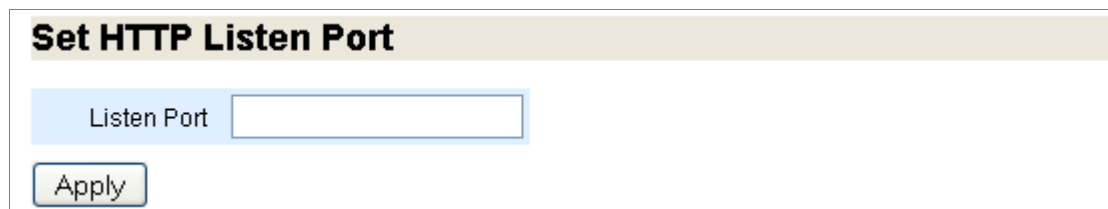
This command sets global attributes; use **namespace** options to set these attributes on a namespace basis; see [“Configuring NameSpaces \(Web Interface\)” on page 301](#).

To configure the HTTP delivery protocol:

- From the left navigation pane in the **Service Config** tab, select **Delivery Protocol**. The **Delivery Protocol Configuration** page is displayed.

Set HTTP Listen Port

The Listen Port is the interface explicitly assigned to receive certain traffic. See [Figure 84](#).



The screenshot shows a configuration window titled "Set HTTP Listen Port". Inside the window, there is a text input field labeled "Listen Port" with a light blue border. Below the input field is an "Apply" button.

Figure 84 Delivery Protocol Configuration Page Detail, Set HTTP Listen Port

To set the HTTP Listen Port from the **Delivery Protocol Configuration** page:

- Enter a port number in the **Listen Port** text box, if needed; default is 80.

Configure/Add Selected HTTP Listen Interfaces

Specify the set of interfaces on which the media delivery protocol listens for incoming requests. If not specified, the delivery protocol listens on all interfaces (default). See [Figure 84](#).

| Interface Name |
|-------------------------------|
| <input type="checkbox"/> eth0 |
| <input type="checkbox"/> lo |

Add

Figure 85 Delivery Protocol Configuration Page Detail, Configure/Add Selected HTTP Listen Interfaces

To add selected HTTP listen interfaces from the **Delivery Protocol Configuration** page:

1. Select the checkbox for the interface you want to add to listen for HTTP traffic. You can specify a list of space-separated interfaces such as **eth2 eth3 eth4**, or **eth10 eth11 eth12**, or **eth20 eth21 eth22**, and so on. Up to 10 can be specified. Click **Add**.
2. Click **Save** at the top of the page to make changes persistent across reboots.

HTTP Listen Interfaces

View or remove selected configured listen interfaces. See [Figure 84](#).

| Interface Name |
|-------------------------------------|
| Default (All configured interfaces) |

Remove Selected

Figure 86 Delivery Protocol Configuration Page Detail, HTTP Listen Interfaces

To add selected HTTP listen interfaces from the **Delivery Protocol Configuration** page:

1. Select the checkbox for the interface you want to remove. Click **Remove Selected**.
2. Click **Save** at the top of the page to make changes persistent across reboots.

HTTP Listen Ports

View or remove selected configured listen ports. See [Figure 84](#).

| HTTP Listen Ports | |
|-------------------|----|
| HTTP Port | 80 |

Remove Selected Port

Figure 87 Delivery Protocol Configuration Page Detail, HTTP Listen Ports

To remove selected HTTP listen ports from the **Delivery Protocol Configuration** page:

1. Select the checkbox for the port you want to remove. Click **Remove Selected Port**.
2. Click **Save** at the top of the page to make changes persistent across reboots.

Configuring Virtual Players (Web Interface)

Create a virtual player and set virtual player options.

- From the left navigation pane in the **Service Config** tab, select **Virtual Player**. The **Virtual Player Configuration** page is displayed.

Add Virtual Player

Add a virtual player and then click **Configure** in the List of Virtual Players to open a new window and make configurations. See [Figure 88](#).

Virtual Player Configuration

Add Virtual Player

Virtual player Name

Virtual Player Type

Add

Figure 88 Virtual Player Page Detail, Add Virtual Player

To configure virtual players from the **Virtual Player Configuration** page, enter this information, and click **Add**:

- **Virtual Player Name**—Name the virtual player; you use this name when adding a virtual player to a namespace.
- **Virtual Player Type**—Choose **Type0** (default) for a generic virtual player; this type has the most configuration options.

Configure, Show, or Remove Virtual Players

Open a configuration window for a new or existing virtual player, or check settings or remove a virtual player. See [Figure 88](#).

| List of Virtual Players Added | | | |
|---------------------------------|---------|---------------------------|----------------------|
| Virtual Player | Type | Configure | Show |
| <input type="checkbox"/> testVP | generic | Configure | Show |

Figure 89 Virtual Player Page Detail, List of Virtual Players Added

To configure, view settings of, or remove selected virtual players from the **Virtual Player Configuration** page:

1. Click the **Configure** link for the virtual player you want to remove. A configuration window opens for that type of virtual player.
2. Click the **Show** link for the virtual player settings you want to view. A n area is displayed with that virtual player's settings.
3. Select the checkbox for the virtual player you want to remove. Click **Remove Selected**.
4. Click **Save** at the top of the page to make changes persistent across reboots.

Virtual Player generic Type Configuration

After adding a **generic** virtual player, click **Configure** in the list of virtual players to open configuration pages (in a new window) for that type of virtual-player. Click **Add/Update** on each configuration page and simply close the window when you are done. Verify your configurations by clicking **Show** in the virtual player list. See [Figure 90](#).

Virtual Player Type generic Configuration Page

Virtual Player Name : testVP

| | |
|---|--|
| Full Download Configuration | Seek Configuration |
| Active <input type="checkbox"/> | Active <input type="checkbox"/> |
| Always <input type="checkbox"/> | Query String Param <input type="text"/> |
| Match String <input type="text"/> | Seek-length Query String Param <input type="text"/> |
| Query String Param <input type="text"/> | Seek FLV Type <input type="text" value="byte-offset"/> |
| Header <input type="text"/> | Seek MP4 Type <input type="text" value="time-msec"/> |
| Assured Flow Configuration | Connection Configuration |
| Active <input type="checkbox"/> | Max Session Rate (kbps) <input type="text" value="0"/> |
| Auto <input type="checkbox"/> | |
| Rate(kbps) <input type="text" value="0"/> | Hash Verify Configuration |
| Query String Param <input type="text"/> | Active <input type="checkbox"/> |
| Fast Start Configuration | Digest <input type="text" value="md-5"/> |
| Active <input type="checkbox"/> | Url Format <input type="text" value="absolute-url"/> |
| Size (KB) <input type="text" value="0"/> | Shared Secret Type <input type="text" value="append"/> |
| Query String Param <input type="text"/> | Shared Secret Value <input type="text"/> |
| | Expiry Time Verify <input type="text"/> |
| | URI Query String Param <input type="text"/> |

Figure 90 Virtual Player Type generic Configuration Page

To configure a generic type virtual player:

1. Enter this information to the text boxes or select the checkbox as described:
 - **Full Download Configuration:** Allow the delivery to download content at the fastest possible speed, limited by the set **connection max-bandwidth** and possibly exceeding the set **assured-flow rate**.
 - **Active**—Select to activate the feature; de-select to de-activate it.
 - **Always**—Download all requests at the maximum possible speed.
 - **Match String**—Download only requests with this value in the URL at the maximum possible speed.
 - **Query String Param**—Enter a query param to signal full download.
 - **Header**—Only download requests with this header at the maximum speed.
 - **Assured Flow Configuration**
 - **Active**—Select to activate the feature; de-select to de-activate it.
 - **Auto**—Not supported in Release 2.0.7.
 - **Rate (kbps)**—Cannot be higher than the configured **connection max-bandwidth**.
 - **Query String Param**—Signals the assured-flow rate.
 - **Fast Start Configuration**
 - **Active**—Select to activate the feature; de-select to de-activate it.
 - **Size (KB)**—Define how many kilobytes should be expedited.
 - **Query String Param**—Signals when to fast start (associated value in kilobytes).
 - **Seek Configuration**
 - **Active**—Select to activate the feature; de-select to de-activate it.
 - **Query String Param**—Tells Media Flow Controller whether or not to implement seek.
 - **Seek-length Query String Param**—Signals the number of bytes of data to send from the **seek** start position.
 - **Seek FLV Type**—Choose one:
 - **byte-offset**—(default) The value of the seek **query-string-param** sent by the client will be in bytes. This option must be set for **seek-length** for FLV.
 - **time-secs**—The value of the seek **query-string-param** sent by the client will be in seconds.
 - **time-msec**—The value of the seek **query-string-param** sent by the client will be in milliseconds.
 - **Seek MP4 Type**—Choose one:
 - **time-secs**—The value of the seek **query-string-param** sent by the client will be in seconds.
 - **time-msec**—(default) The value of the seek **query-string-param** sent by the client will be in milliseconds.
 - **Connection Configuration**
 - **Max Session Rate (kbps)**—Set the maximum bandwidth for a session. The actual session bandwidth is between the AFR (Assured Flow Rate) and this value. Even if there is available bandwidth in the link, Media Flow Controller does not allocate more than this value for a session. When it is a full download, Media Flow Controller tries to allocate the max-bandwidth to the session. Default it is **0** Kbps

(unbounded) with the Media Flow Controller license. You must have the Media Flow Controller license installed to change the default (**200** kbps without the license).

- **Hash Verify Configuration** See [“Using hash-verify” on page 127](#) for details.
 - **Active**—Select to activate the feature; de-select to de-activate it.
 - **Digest**—Only **md-5** is supported in Release 2.0.7.
 - **URL Format**—What part of the request URL to use for the hash calculation:
 - **absolute-uri**—Current and default behavior. The hash calculation should use the entire request URL (including query string up to configured **match query-string-parm** value).
 - **relative-uri**—The hash calculation should use only the URI part of the request URL, excluding the domain, and access method (but including query string up to configured **match query-string-parm** value).
 - **object-name**—The hash calculation should use only the object name part of the request URL (and query string up to configured **match query-string-parm** value).
 - **Shared Secret Type**—**append** or **prefix** to the URL the **shared secret value**.
 - **Shared Secret Value**—A secret key which must match the hash value provided in the URL and indicated by the **URI Query String Parm**.
 - **Expiry Time Verify**—Enter a query-string parameter value to reject incoming requests that have passed the current system time; coupled with hash verification, this helps prevent bandwidth stealing. The value for the query-string parameter must be an expiration time specified as a standard POSIX timestamp (seconds since January 1 1970 00:00:00 UTC). The timestamp value in the URL is generated by the player issuing the request. Media Flow Controller compares this timestamp with the current time to determine if the URL request has expired. An expiry time of **0** (zero) (for example, in cases where the player does not give a timestamp) indicates that this request “does not expire”.
Sample URL request form where “e” indicates the **expiry-time-verify query-string-parm** value:

```
http://www.example.com/media/  
foo.flv?e=3312665958&h=ec41f550878f45d9724776761d6ac416
```

- **URI Query String Param**—This indicates the provided hash value in the URL that, when processed by the hash digest, must match the **Shared Secret Value**.
2. Click **Add/Update**.
 3. Click **Save** at the top of the page to make changes persistent across reboots.

Virtual Player qss-streamlet Type Configuration

After adding a **qss-streamlet** virtual player, click **Configure** in the list of virtual players to open configuration pages (in a new window) for that type of virtual-player. Click **Add/Update** on each configuration page and simply close the window when you are done. Verify your configurations by clicking **Show** in the virtual player list.

Virtual Player Type qss-streamlet Configuration Page

Virtual Player Name : testq

Rate Map Configuration

Active

Match String

Rate

Add/Update

Connection Bandwidth Configuration

Max Session Rate (kbps)

Add/Update

Figure 91 Virtual Player Type qss-streamlet Configuration Page

To configure a **qss-streamlet** type virtual player:

- Enter this information to the text boxes or select the checkbox as described:
 - Rate Map Configuration:** Change the **assured flow** rate associated with each string or specify a parameter to use to find the desired rate in the URL.
 - Active**—Select to activate the feature; de-select to de-activate it.
 - Match String**—Enter a 2-byte string value; for example **01**.
 - Rate**—Enter a value for assured flow for that **Match String**; for example, **300**. Media Flow Controller finds the **Match String** value in the URL (by going to the end of the URL and skipping 12 bytes from the end) and uses its associated **Rate** value for the assured flow rate for each HTTP request.
- Click **Add/Update**.
- Click **Save** at the top of the page to make changes persistent across reboots.

For more information on configuring rate maps, see [“Using virtual-player type qss-streamlet rate-map” on page 128](#).

Virtual Player yahoo Type Configuration

After adding a **yahoo** virtual player, click **Configure** in the list of virtual players to open configuration pages (in a new window) for that type of virtual-player. Click **Add/Update** on each configuration page and simply close the window when you are done. Verify your configurations by clicking **Show** in the virtual player list.

Compute MD-5 hash of query string parameters representing **stream-id**, **auth-id**, a configured **shared-secret**, and **time-interval**; and match the computed value with the specified **match query-string-param <string>**. The HTTP GET proceeds if the computed MD-5 hash matches; if there is no match, the session is rejected. Use **virtual player <name> type 3 no req-auth** to disable.



CAUTION: Virtual-player **yahoo** requires a special license to be configured.

Virtual Player Type yahoo Configuration Page

Virtual Player Name : newY

| | |
|--|---|
| <h4 style="margin: 0;">Req-Auth configuration</h4> <p>Active <input checked="" type="checkbox"/></p> <p>Digest md-5</p> <p>Auth ID URI authid</p> <p>Query String </p> <p>Match URI ticket</p> <p>Query String </p> <p>Shared Secret ysecret</p> <p>String </p> <p>Stream ID Query streamid</p> <p>String Param </p> <p>Time Interval 15</p> | <h4 style="margin: 0;">Seek Configuration</h4> <p>Active <input type="checkbox"/></p> <p>Query String </p> <p>Param </p> <p>Seek-length </p> <p>Query String </p> <p>Param </p> <p>Seek FLV Type byte-offset</p> <p>Seek MP4 Type time-secs</p> |
| <h4 style="margin: 0;">Health-Probe Configuration</h4> <p>Active <input checked="" type="checkbox"/></p> <p>Query String no-cache</p> <p>Param </p> <p>Match string 1</p> | <h4 style="margin: 0;">Assured Flow Configuration</h4> <p>Active <input type="checkbox"/></p> <p>Auto <input type="checkbox"/></p> <p>Rate(kbps) 0</p> <p>Query String </p> <p>Param </p> |

Add/Update

Figure 92 Virtual Player Type yahoo Configuration Page

To configure a **yahoo** type virtual player:

1. Enter this information to the text boxes or select the checkbox as described:
 - **Req-Auth Configuration**
 - **Active**—Select to activate the feature; de-select to de-activate it.
 - **Digest**—Only **md-5** is supported in Release 2.0.7.
 - **Auth ID URI Query String**—String value to be hashed.
 - **Match URI Query String**—String value must match computed hash of **Auth ID URI Query String**, **Shared Secret String**, **Stream ID Query String Param**, and **Time Interval** values.
 - **Shared Secret String**—String value to be hashed.
 - **Stream ID Query String Param**—String value to be hashed.
 - **Time Interval**—Integer value.
 - **Health-Probe Configuration** (See [“Terminology” on page 31](#) for explanation and example of **uol offset** and **uol length**. See [“Using hash-verify” on page 127](#) for explanation).
 - **Active**—Select to activate the feature; de-select to de-activate it.
 - **Query String Param**—This is the value that, when processed by the hash digest, must match the **Match String** value.
 - **Match String**—Set the secret string that the hash attempts to verify.
 - **Seek Configuration**
 - **Active**—Select to activate the feature; de-select to de-activate it.
 - **Query String Param**—Enter a name to tell Media Flow Controller whether or not to implement seek.
 - **Seek-length Query String Param**—Enter a name to signal the number of bytes of data to send from the **seek** start position; referenced value must be in bytes. For FLV support of this option, **seek-flv-type** must be set to **byte-offset**.
 - **Seek FLV Type**—Choose one:
 - **byte-offset**—The value of the seek **query-string-param** sent by the client will be in bytes. This option must be set for **seek-length** for FLV.
 - **time-secs**—The value of the seek **query-string-param** sent by the client will be in seconds.
 - **time-msec**—The value of the seek **query-string-param** sent by the client will be in milliseconds.
 - **Seek MP4 Type**—Choose one:
 - **time-secs**—The value of the seek **query-string-param** sent by the client will be in seconds.
 - **time-msec**—The value of the seek **query-string-param** sent by the client will be in milliseconds.

- Assured Flow Configuration
 - **Active**—Select to activate the feature; de-select to de-activate it.
 - **Auto**—Not supported in Release 2.0.7.
 - **Rate (kbps)**—Set a rate no higher than the configured **connection max-bandwidth**.
 - **Query String Param**—Signals the desired assured-flow rate.
 - **Connection Bandwidth Configuration**
 - **Max Session Rate (kbps)**—Set the maximum bandwidth for a session. The actual session bandwidth is between the AFR (Assured Flow Rate) and this value. Even if there is available bandwidth in the link, Media Flow Controller does not allocate more than this value for a session. When it is a full download, Media Flow Controller tries to allocate the max-bandwidth to the session. Default it is **0** Kbps (unbounded) with the Media Flow Controller license. You must have the Media Flow Controller license installed to change the default (**200** kbps without the license).
2. Click **Add/Update**.
 3. Click **Save** at the top of the page to make changes persistent across reboots.

Virtual Player youtube Type Configuration

After adding a **youtube** virtual player, click **Configure** in the list of virtual players to open configuration pages (in a new window) for that type of virtual-player. Click **Add/Update** on each configuration page and simply close the window when you are done. Verify your configurations by clicking **Show** in the virtual player list. See [“Configuring YouTube Video Caching \(CLI\)” on page 131](#) for implementation details.

Virtual Player Type youtube Configuration Page

Virtual Player Name : newYT

Seek Query Configuration

Active

Seek query string param

Seek-length query string param

Seek FLV Type

Seek MP4 Type

Fast Start Configuration

Active

Size (KB)

Query String Param

Connection Bandwidth Configuration

Max Session Rate (kbps)

Cache Name Configuration

Active

Video-id(Query string)

Format Tag

Assured Flow Configuration

Active

Auto

Rate(kbps)

Query String Param

Figure 93 Virtual Player Type youtube Configuration Page

To configure a **youtube** type virtual player:

1. Enter this information to the text boxes or select the checkbox as described:
 - **Seek Configuration**
 - **Active**—Select to activate the feature; de-select to de-activate it.
 - **Seek query string param**—Enter a name to tell Media Flow Controller whether or not to implement seek.
 - **Seek-length query string param**—Enter a name to signal the number of bytes of data to send from the **seek** start position; referenced value must be in bytes.
 - **Seek FLV Type**—Choose one:
 - **byte-offset**—The value of the seek **query-string-param** sent by the client will be in bytes. This option must be set for **seek-length** for FLV.
 - **time-secs**—The value of the seek **query-string-param** sent by the client will be in seconds.

- **time-msec**—The value of the seek **query-string-param** sent by the client will be in milliseconds.
 - **Seek MP4 Type**—Choose one:
 - **time-secs**—The value of the seek **query-string-param** sent by the client will be in seconds.
 - **time-msec**—The value of the seek **query-string-param** sent by the client will be in milliseconds.
 - **Connection Bandwidth Configuration**
 - **Max Session Rate (kbps)**—Set the maximum bandwidth for a session. The actual session bandwidth is between the AFR (Assured Flow Rate) and this value. Even if there is available bandwidth in the link, Media Flow Controller does not allocate more than this value for a session. When it is a full download, Media Flow Controller tries to allocate the max-bandwidth to the session. Default it is **0** Kbps (unbounded) with the Media Flow Controller license. You must have the Media Flow Controller license installed to change the default (**200** kbps without the license).
 - **Assured Flow Configuration**
 - **Active**—Select to activate the feature; de-select to de-activate it.
 - **Auto**—Not supported in Release 2.0.7.
 - **Rate (kbps)**—Set a rate no higher than the configured **connection max-bandwidth**.
 - **Query String Param**—Enter a query param to signal the desired assured-flow rate.
 - **Fast Start Configuration**
 - **Active**—Select to activate the feature; de-select to de-activate it.
 - **Size (KB)**—Define how many kilobytes should be expedited.
 - **Query String Param**—Specify a query param (associated value must be in kilobytes).
 - **Cache Name Configuration**
 - **Active**—Select to activate the feature; de-select to de-activate it.
 - **Video-id (Query string)**—Specify a query param string whose value provides the requested video ID (for example, **id**). YouTube video URI requests do not specifically associate a name to a video asset in the URI, instead a unique query param is used.
 - **Format Tag**—Media Flow Controller detects the bit-rate of the content being served and sets AFR to that. Specify a query param string whose value provides the requested format (for example, **fmt** or **itag**). YouTube video URI requests do not specifically associate a format to a video asset in the URI, instead a unique query param is used. Acceptable format values are shown in [“Using Virtual Player Type YouTube” on page 131](#).
2. Click **Add/Update**.
 3. Click **Save** at the top of the page to make changes persistent across reboots.

Virtual Player smoothstream-pub Type Configuration

After adding a **smoothstream-pub** virtual player, click **Configure** in the list of virtual players to open configuration pages (in a new window) for that type of virtual-player. Click **Add/Update** on each configuration page and simply close the window when you are done. Verify your configurations by clicking **Show** in the virtual player list. See [“Configuring SmoothStream Video Caching \(CLI\)” on page 134](#) for implementation details.

Virtual Player Type smoothstream-pub Configuration Page

Virtual Player Name : newSSP

Configuration

Fragment tag

Quality tag

Figure 94 Virtual Player Type smoothstream-pub Configuration Page

To configure a **smoothstream-pub** type virtual player:

1. Enter this information to the text boxes or select the checkbox as described:
 - **Configuration**
 - **Fragment tag**—Set an identifier whose value describes the timestamp of the requested media segment. Default is **Fragments** (case sensitive).
 - **Quality Tag**—Set an identifier to describes the bit rate of the requested media segment. Default is **QualityLevels** (case sensitive).
2. Click **Add/Update**.
3. Click **Save** at the top of the page to make changes persistent across reboots.

Virtual Player flashstream-pub Type Configuration

After adding a **flashstream-pub** virtual player, click **Configure** in the list of virtual players to open configuration pages (in a new window) for that type of virtual-player. Click **Add/Update** on each configuration page and simply close the window when you are done. Verify your configurations by clicking **Show** in the virtual player list. See [“Configuring FlashStream Video Caching \(CLI\)” on page 135](#) for implementation details.

Virtual Player Type flashstream-pub Configuration Page

Virtual Player Name : newFSP

Configuration

| | |
|----------------------------|-----------------------------------|
| Fragment tag | <input type="text" value="Frag"/> |
| Segment tag | <input type="text" value="Seg"/> |
| Segment fragment delimiter | <input type="text"/> |

Figure 95 Virtual Player Type flashstream-pub Configuration Page

To configure a **flashstream-pub** type virtual player:

- Enter this information to the text boxes or select the checkbox as described:
 - Configuration**
 - Fragment tag**—Enter a string whose value describes to Media Flow Controller the string to search while parsing a Zeri HDS request. The number immediately following this tag in the request denotes the fragment number. Default is **Frag** (case sensitive).
 - Segment Tag**—Enter a string to describe to Media Flow Controller the string to search while parsing a Zeri HDS request. The number immediately following this tag in the request denotes the segment number. Default is **Seg** (case sensitive).
 - Segment fragment delimiter**—Set an identifier to describe to Media Flow Controller the separator to search while parsing a Zeri HDS request which acts as a delimiter to the segment and fragment tags. Default is - (dash).
- Click **Add/Update**.
- Click **Save** at the top of the page to make changes persistent across reboots.

Configuring NameSpaces (Web Interface)

Create a namespace and set namespace options; these are set in the CLI with **namespace**.

A namespace is a named collection of parameters that set delivery policies in a granular manner; you can configure up to 256 namespaces. To set global delivery policies, see **delivery** for CLI details. See [Chapter 6, “Configuring Namespaces \(CLI\)”](#) for task details and implementation particulars. See **namespace** for CLI details.

Tip! At a minimum, a **namespace** configuration requires a **name**, **domain**, **origin-server**, **match** setting, and **status** activation. Other settings can be left at defaults.

Add Namespace

Enter a name for the new namespace and click **Add Namespace**. See [Figure 96](#).

The screenshot shows a form titled "Add Namespace". It contains a text input field with the label "Namespace" and a button labeled "Add" positioned below the input field.

Figure 96 Service Config > Namespace Page Detail, Add Namespace

Configuration List

Select a namespace and click **Configure** to open a new window and make configurations; click **Show** to see current settings; click **Activate** / **Deactivate** to take those actions. See [Figure 97](#).

The screenshot shows a table titled "Configuration List" with the following data:

| Namespace | Active | Configure | Show |
|------------------------------------|--------|---------------------------|----------------------|
| <input type="checkbox"/> austin | no | Configure | Show |
| <input type="checkbox"/> baltimore | no | Configure | Show |

Below the table, there are two buttons: "Activate" and "Deactivate".

Figure 97 Service Config > Namespace Page Detail, Configuration List

Namespace Configuration

Use this window to configure a new namespace or make changes to an existing one. After each configuration area, click **Apply**. When done click **Done**, at the page bottom. Namespace configuration is broken into six areas:

- [“Configuring Namespace Origin Server \(Web Interface\)” on page 303](#)
- [“Configuring Namespace HTTP Match Details \(Web Interface\)” on page 304](#)
- [“Configuring Namespace Parameters \(Web Interface\)” on page 305](#)
- [“Configuring Namespace Pre-Stage User \(Web Interface\)” on page 306](#)
- [“Configuring Namespace HTTP Origin Fetch \(Web Interface\)” on page 307](#)
- [“Configuring Namespace RTSP Origin Fetch \(Web Interface\)” on page 308](#)

Configuring Namespace Origin Server (Web Interface)

Configure fetching content upon a cache-miss. In Release 2.0.7 only one (1) origin server is supported (either HTTP or NFS); multiple origin servers can be configured using a **server-map**; see [namespace](#) for CLI details. Click **Apply** when done. See [Figure 98](#).

Namespace : new

Origin Server Configuration

HTTP Origin Hostname Port
 Server-map
 Absolute-URL
 Follow-Header

NFS Origin Hostname Port
 Server-map

Domain Host
 Regex

Figure 98 Service Config > Namespace **Configure** Page Detail, Origin Server Configuration

To configure a **namespace origin server**:

1. Enter this information to the text boxes or select the radio button as described:
 - **HTTP Origin**—Use HTTP for origin-server delivery.
 - **Hostname** and **Port**—Use HTTP for origin-server delivery; multiple origin servers (up to four) can be specified. Specify either a **hostname** as an FQDN (Fully Qualified Domain Name) or **IP address**. Specify a **port number** (default is **80**).
 - **Server-map**—Enter the name of a **server-map** defined on the system. See [Chapter 8, “Configuring Media Flow Controller Server Maps”](#) for task details.
 - **Absolute-URL**—Select to derive the origin server from the absolute URL set against the HTTP access method (GET, HEAD, and so forth.). Use **absolute-url** to configure Media Flow Controller as a mid-tier, proxy (if set, **show namespace** output has Proxy Mode: forward). Media Flow Controller uses the absolute URL to

contact the origin-server. If **absolute-url** is configured and the incoming request (REQ header) does *not* have the absolute URL, then the request is rejected with the appropriate error code.

- **Follow-Header**—Enter a header name, used in the incoming request, to be the origin server. For example, you could configure **follow header HOST**, and the value of the HOST header in the incoming request is used as the origin server. If the configured header does not exist, then the request is rejected. The **<header name>** can be any of the well-defined headers OR a custom header. If set, **show namespace** output has Proxy Mode: virtual. In Release 2.0.7, only **host** is allowed.
 - **NFS Origin**—Use NFS for origin-server delivery.
 - **Hostname** and **Port**—Specify a **port number** (default is **2049**) for that origin server.
 - **Server-map**—Enter the name of a **server-map** defined on the system. See [server-map](#) for CLI details; [Chapter 8, “Configuring Media Flow Controller Server Maps”](#) for task details.
 - **Domain**—Enter a FQDN (fully qualified domain name) or REGEX (regular expression) that is matched with the incoming HOST header. If there is a match, the request is refined with the **match** value. See [“Using namespace domain <FQDN:Port>” on page 144](#) for implementation details.
 - **Host**—Domain name.
 - **Regex**—A regex to indicate the domain name.
2. Click **Apply**.
 3. Click **Done** at the bottom of the page if you are finished.

Configuring Namespace HTTP Match Details (Web Interface)

Refine the path of incoming requests (enclose all **regex** entries in double quotes). All **match** options may utilize the optional **precedence** argument to break ties when namespaces are defined with the same **match** criteria. Click **Apply** when done. See [Figure 99](#).

HTTP Match Details

Precedence ▼

URI uri-name uri-regex

Header header-name header-value header-regex

Query-string query-name query-value query-regex

Virtual-host virtual-IP virtual-port

Figure 99 Service Config > Namespace **Configure** Page Detail, Match Details

To configure **namespace HTTP Match Details**:

- Select the radio button as described and enter this information to the text boxes:
 - Precedence**—Map incoming requests to a namespace, the lower the **precedence** number, the higher the preference for that namespace; **0** (zero) is default and highest; see [“Using namespace match <criteria> precedence” on page 145](#), for details.
 - Uri**—Specify a uri-prefix **match** criteria. See [“uri-prefix” on page 34](#) for usage details.
 - uri-name**—Enter a uri-prefix, use / (slash) for “any.” See [“Terminology” on page 31](#) for definition and example of uri-prefix.
 - uri-regex**—Enter a regex to indicate the uri-prefix.
 - Header**—A header name and value; can also be a **regex**. Optionally, set a **precedence** (defined above).
 - header-name** and **header-value**
 - header-regex**
 - Query-string**—A name and value; can also be a **regex**. Optionally, set a **precedence** (defined above).
 - query-name** and **query-value**
 - query-regex**
 - Virtual-host**—Enter the address and port (optional) of a virtual host.
 - virtual-IP** and **virtual-port**—The IP address must be a /32 address; it can be **0.0.0.0**, which means any IP address. Port number specification is optional. To map requests by TCP port number only, set the IP address to 0.0.0.0 and configure the port number. If you set the domain to **any**, configure **virtual-host IP** to **0.0.0.0**, then requests can be assigned to a namespace based solely on the port number on which the request comes in to Media Flow Controller. Optionally, set a **precedence** (defined above).
- Click **Apply**.
- Click **Done** at the bottom of the page if you are finished.

Configuring Namespace Parameters (Web Interface)

Set cache inherit, status, virtual player, and delivery protocol options. See [Figure 100](#).

The screenshot shows a configuration panel titled "Parameters" with a light blue background. It contains the following fields and controls:

- Cache Inherit**: A text input field.
- Status Active**: A checkbox that is currently unchecked.
- Virtual Player**: A text input field.
- Delivery Protocol**: Two radio buttons, "http" (checked) and "rtsp" (unchecked).
- Apply**: A button located at the bottom left of the panel.

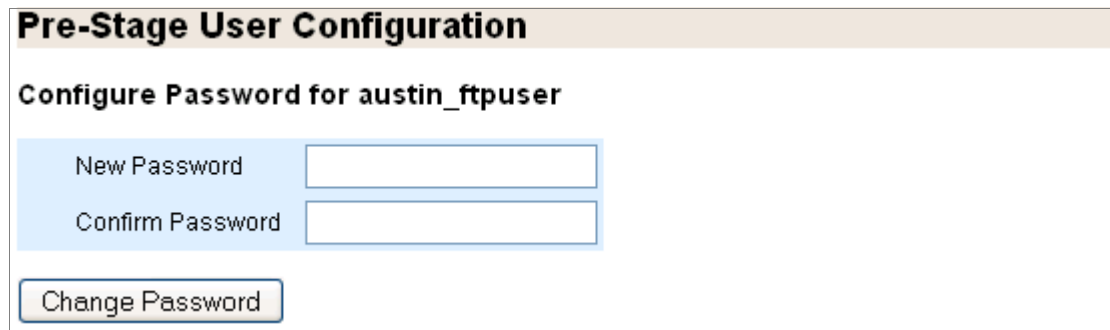
Figure 100 Service Config > Namespace **Configure** Page Detail, Parameters

To configure **namespace Parameters**:

1. Enter this information to the text boxes or select the checkbox as described:
 - **Cache Inherit**—Add the specified namespace's cache to this one; see [“Using namespace cache-inherit” on page 142](#), for details.
 - **Status Active**—Activate the namespace; de-select to de-activate but not delete.
 - **Virtual Player**—Assign an existing virtual player to this namespace; the virtual player settings override the global **network connection** settings (for this namespace).
 - **Delivery Protocol**—Set a protocol for responses from this namespace.
2. Click **Apply**.
3. Click **Done** at the bottom of the page if you are finished.

Configuring Namespace Pre-Stage User (Web Interface)

Set a password for the auto-created FTP user for this namespace. See [Figure 101](#).



Pre-Stage User Configuration

Configure Password for austin_ftpuser

New Password

Confirm Password

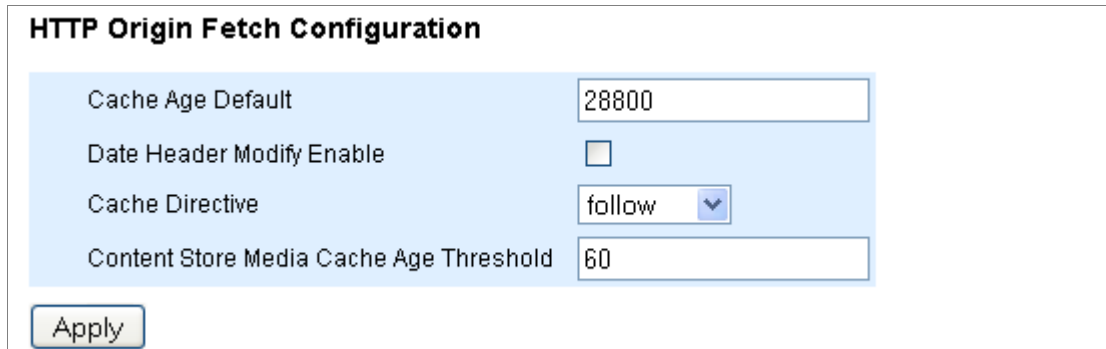
Figure 101 Service Config > Namespace Configure Page, Pre-Stage User Configuration

To configure **namespace Pre-Stage User**:

1. Enter this information to the text boxes or select the checkbox as described:
 - **New Password** and **Confirm Password**—For each configured namespace FTP user (created automatically), set or reset a password.
2. Click **Change Password**.
3. Click **Done** at the bottom of the page if you are finished.

Configuring Namespace HTTP Origin Fetch (Web Interface)

Configure options for HTTP fetching content from origin upon a cache miss. See [Figure 102](#).



| HTTP Origin Fetch Configuration | |
|---|-------------------------------------|
| Cache Age Default | <input type="text" value="28800"/> |
| Date Header Modify Enable | <input type="checkbox"/> |
| Cache Directive | <input type="text" value="follow"/> |
| Content Store Media Cache Age Threshold | <input type="text" value="60"/> |

Figure 102 Service Config > Namespace **Configure** Page, HTTP Origin Fetch Configuration

To configure **namespace HTTP Origin Fetch**:

1. Enter this information to the text boxes or select the checkbox as described:
 - **Cache Age Default**—Specify a cache age value in case it is not specified in the data fetched from the origin server. Default is **28800** seconds (8 hours).
 - **Date Header Modify Enable**—Select to enable Media Flow Controller to set the **Date** header to the current time when the content is served to the client (no adjustments are made to the **Cache-Control: max-age = <seconds>** header, and the **Age** header sent from origin is maintained). De-select to disable (default); Media Flow Controller does not reset the **Date** header.
 - **Cache Directive**—Choose **follow** (default) to tell Media Flow Controller to obey the cache-directive (**Cache-control : no-cache** or **Pragma: no-cache**) in the HTTP header when data is fetched from the origin. Choose **override** to tell Media Flow Controller to always cache.
 - **Content Store Media Cache Age Threshold**—Set a time threshold for newly-fetched content stored in **media** cache (non-volatile) instead of RAM. By default, new content with a cache age under **60** seconds is stored only in RAM (the expectation is short cache-age implies that the content will not be served for too long and is not worth storing in media cache). To have new content always stored in media cache, set this value to **0** (zero).
2. Click **Apply**.
3. Click **Done** at the bottom of the page if you are finished.

Configuring Namespace RTSP Origin Fetch (Web Interface)

Configure options for RTSP fetching content from origin upon a cache miss. See [Figure 103](#).

| RTSP Origin Fetch Configuration | |
|---|-------------------------------------|
| Cache Age Default | <input type="text" value="28800"/> |
| Cache Directive | <input type="text" value="follow"/> |
| Content Store Media Cache Age Threshold | <input type="text" value="60"/> |
| <input type="button" value="Apply"/> | |
| <input type="button" value="Done"/> | |

Figure 103 Service Config > Namespace **Configure** Page, RTSP Origin Fetch Configuration

To configure **namespace HTTP Origin Fetch**:

- Enter this information to the text boxes or select the checkbox as described:
 - Cache Age Default**—Specify a cache age value in case it is not specified in the data fetched from the origin server. Default is **28800** seconds (8 hours).
 - Cache Directive**—Choose **follow** (default) to tell Media Flow Controller to obey the cache-directive (**Cache-control : no-cache** or **Pragma: no-cache**) in the HTTP header when data is fetched from the origin. Choose **override** to tell Media Flow Controller to always cache.
 - Content Store Media Cache Age Threshold**—Set a time threshold for newly-fetched content stored in **media** cache (non-volatile) instead of RAM. By default, new content with a cache age under **60** seconds is stored only in RAM (the expectation is short cache-age implies that the content will not be served for too long and is not worth storing in media cache). To have new content always stored in media cache, set this value to **0** (zero).
- Click **Apply**.
- Click **Done** at the bottom of the page if you are finished.

Managing the Media-Cache (Web Interface)

Manage media cache disks. The media caches/disks are active and enabled by default and typically require no configuring. However, you must deactivate and disable disks and caching to change disks. See [media-cache](#) for CLI details.

Disk Name

Select the disk you want to configure and Activate, Deactivate, Cache Enable, Cache Disable, or Format it (Repair not supported in Release 2.0.7). See [Figure 104](#).

Click **Deactivate** if you need to pull the disk for any maintenance purposes; for example, to upgrade to a higher capacity disk, or replace a failed disk.



CAUTION: Media Flow Controller allows OIR (On-line Insertion and Removal) of HDD (Hard Disk Drives). However, **the HDD MUST be made inactive to be removed**. When a new HDD is in the disk, it must be made active and (if so decided) enabled for caching.

Disk Cache

| | |
|---------------|----------------------------------|
| Disk Name | dc_1 ▾ |
| Activate | <input checked="" type="radio"/> |
| Deactivate | <input type="radio"/> |
| Cache Enable | <input type="radio"/> |
| Cache Disable | <input type="radio"/> |
| Format | <input type="radio"/> |
| Repair | <input type="radio"/> |

Figure 104 Service Config > Media-Cache Page Detail

Configuring Service Logging (Web Interface)

Set Media Flow Controller **Access Log** and **Stream Log** options, including automatic upload.

Access Log Configuration

The Accesslog records HTTP transactions. See [Figure 105](#).

Access Log Configuration

| | |
|-------------------------|---|
| Syslog Replicate Enable | <input type="checkbox"/> |
| Log Format | <pre style="font-family: monospace; font-size: 0.9em;">%c %h %V %u %t "%r" %s %b %N "%{Cache-Control}i" "%{Pragma}i" "%{Cache-Control}o" "%{Pragma}o" "%{Vary}o" %y</pre> |

Figure 105 Log Configuration Page Detail, Access Log Configuration

To configure the accesslog:

1. Enter this information to the text boxes or select the checkbox as described:
 - **Enable**—Enable access logging; this records all user activity on the system. The no variant disables. Default is enabled.
 - **Syslog Replicate Enable**—Specify whether or not the access log messages are seen as part of SYSLOG also. By default the option is **disabled** (access log is not seen as part of SYSLOG).
 - **Log Format**—Specify the format in which access log is to be obtained. See [accesslog](#) for CLI details on log formatting.
2. Click **Apply**.
3. Click **Save** at the top of the page to make changes persistent across reboots.

Access Log Copy/Auto Download Configuration

Set automatic uploading of the accesslog. See [Figure 106](#).

Access Log Copy/Auto Upload Configuration

Upload via scp (pseudo-URL format: scp://username@hostname/path/filename):

URL:

Password:

Figure 106 Log Configuration Page Detail, Access Log Copy/Auto Upload Configuration

To download the accesslog:

1. Enter this information to the text boxes:
 - **URL**—The URL to which the accesslog should be downloaded when it reaches the specified download trigger size, by default when filesize reaches 100MB; this can be changed using the CLI (**rotate filesize**), see [accesslog](#) for details.
 - **Password**—The password needed for the download.
2. Click **Apply**.
3. Click **Save** at the top of the page to make changes persistent across reboots.

Tip! If this Media Flow Controller is going to be managed by a CMC server, set the auto-upload URL to the address of the CMC server and the filepath to **/log**.

Stream Log Configuration

This log records RTSP streaming transactions. See [Figure 105](#).

Stream Log Configuration

Enable

Syslog Replicate Enable

Log Format `%h %c %t %x \"%r\" %s %I %O`

Apply

Figure 107 Log Configuration Page Detail, Stream Log Configuration

To configure the streamlog:

1. Enter this information to the text boxes or select the checkbox as described:
 - **Enable**—Enable access logging; this records all user activity on the system. The no variant disables. Default is enabled.
 - **Syslog Replicate Enable**—Specify whether or not the stream log messages are seen as part of SYSLOG also. By default the option is **disabled** (access log is not seen as part of SYSLOG).
 - **Log Format**—Specify the format in which stream log is to be obtained. See [streamlog](#) for CLI details on log formatting.
2. Click **Apply**.
3. Click **Save** at the top of the page to make changes persistent across reboots.

Stream Log Copy/Auto Download Configuration

Set automatic downloading of the streamlog. See [Figure 108](#).

Stream Log Copy/Auto Upload Configuration

Upload via scp (pseudo-URL format: scp://username@hostname/path/filename):

URL:

Password:

Apply

Figure 108 Log Configuration Page Detail, Stream Log Copy/Auto Upload Configuration

To download the streamlog:

1. Enter this information to the text boxes:
 - **URL**—The URL to which the streamlog should be downloaded when it reaches the specified download trigger size, by default when filesize reaches 100MB; this can be changed using the CLI (**streamlog rotate filesize**), see [streamlog](#) for details.
 - **Password**—The password needed for the download.
2. Click **Apply**.
3. Click **Save** at the top of the page to make changes persistent across reboots.

Tip! If this Media Flow Controller is going to be managed by a CMC server, set the auto-upload URL to the address of the CMC server and the filepath to **/log**.

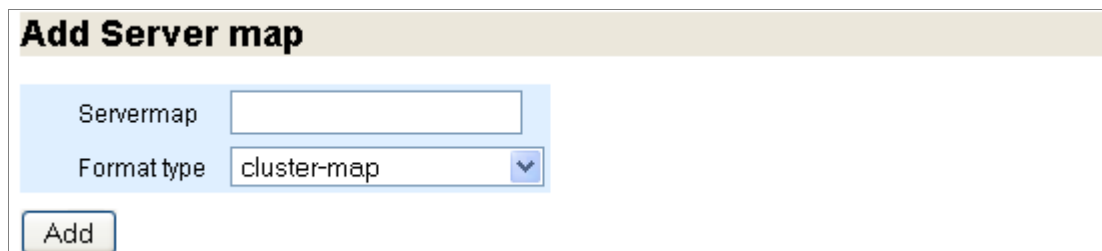
Configuring Server Maps (Web Interface)

Media Flow Controller can use an XML file to resolve incoming client requests to the right origin server when Media Flow Controller encounters a cache-miss. Create the XML file following the conventions outlined in [Chapter 8, “Configuring Media Flow Controller Server Maps”](#) and save it; then use these pages to name the server map, and choose a format-type, reference it to Media Flow Controller with a URL, and set other parameters. After it is configured, assign the server-map to a namespace origin-server. See [server-map](#) and [namespace](#) for CLI details.

See [Chapter 8, “Configuring Media Flow Controller Server Maps.”](#) for information on creating **server-map** XML files.

Add Server Map

Enter a name for the new server and click **Add**. See [Figure 96](#).



The screenshot shows a web interface for adding a server map. The title is "Add Server map". There are two input fields: "Servermap" which is currently empty, and "Format type" which is set to "cluster-map" with a dropdown arrow. Below these fields is an "Add" button.

Figure 109 Server map Configuration Page Detail, Add Servermap

Configuration List

Select a server map and click **Configure** to open a new window and make configurations; click **Show** to see current settings; click **Remove** to delete the server map. See [Figure 110](#).

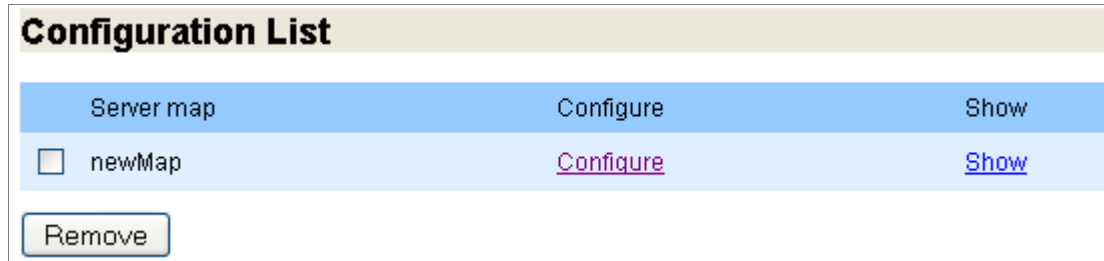


Figure 110 Server map Configuration Page Detail, Configuration List

Refresh Force

Select a server and click **Refresh-force** to cause Media Flow Controller to immediately check the map's XML file for changes. See [Figure 111](#).

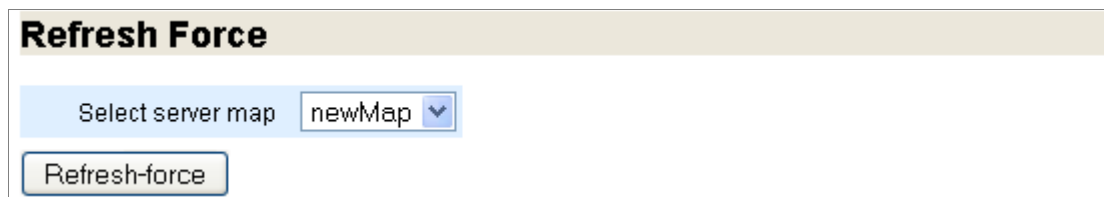


Figure 111 Server map Configuration Page Detail, Refresh Force

Server Map Configuration

Use this window to configure a new server map or make changes to an existing one. See [Figure 98](#).

The screenshot displays the 'Server map Configuration' window. It is divided into two main sections: 'Server map : newMap' and 'Monitoring Parameters'. The 'Server map : newMap' section contains two input fields: 'URL' with the value 'http://example.com/vod,' and 'Refresh interval' with the value '9000'. Below these fields is an 'Add' button. The 'Monitoring Parameters' section contains four input fields: 'Allowed failures' with the value '3', 'Connection timeout' with the value '100', 'Interval' with the value '100', and 'Read time-out' with the value '100'. Below these fields are two buttons: 'Add' and 'Done'.

Figure 112 **Server map Configuration** Page Detail, Server map Configuration

To configure a **server map**:

1. Add the new server map to Media Flow Controller; enter this information to the text boxes:
 - **URL**—The location of the server map XML file.
 - **Refresh interval**—How often Media Flow Controller should check the server map XML file for changes. Permitted values for time are **0** (no refresh) or **300** seconds (minimum) to **86400** seconds (maximum). Default is **0**.
2. Click **Add**.
3. Add monitoring parameters; enter this information to the text boxes:

- **Allowed failures**—How many request failures are allowed before the node being requested is declared down. Default is **3**, minimum allowed value is **0** (zero); maximum value is **32**.
 - **Connection timeout**—The allowable time in milliseconds for the socket connect to complete.
 - **Interval**—The time in milliseconds for nodes to wait before sending a “heartbeat” signal to the other nodes indicating availability status.
 - **Read time-out**—The allowable time in milliseconds for the socket read to complete after the connection is established.
4. Click **Add**.
 5. Click **Done** at the bottom of the page if you are finished.

Viewing Logs Overview

Media Flow Controller supplies several logs. [Figure 113](#) shows the **Logs** menu.

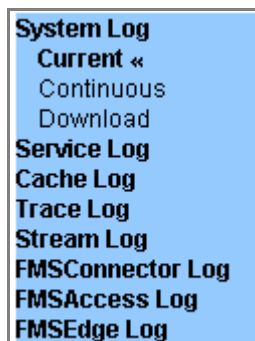


Figure 113 Logs tab left navigation menu

To view system log information see:

- [“Viewing the System Log \(Web Interface\)” on page 315](#)
- [“Viewing the Service Log \(Web Interface\)” on page 316](#)
- [“Viewing the Cache Log \(Web Interface\)” on page 316](#)
- [“Viewing the Trace Log \(Web Interface\)” on page 317](#)
- [“Viewing the Stream Log \(Web Interface\)” on page 317](#)
- [“Viewing the FMSCconnector Log \(Web Interface\)” on page 317](#)
- [“Viewing the FMSSAccess Log \(Web Interface\)” on page 317](#)
- [“Viewing the FMSEdge Log \(Web Interface\)” on page 318](#)

Viewing the System Log (Web Interface)

Use this tab to access system logs; this log (syslog) records system activity. See [Figure 114](#) for graphic. See [“Reading the Media Flow Controller System Log” on page 201](#) for additional details.

Continuous log, updated every 10 seconds.

Current log, that day's activity.

Download log—Opens a **File Open** dialog so you can save the trace log locally.

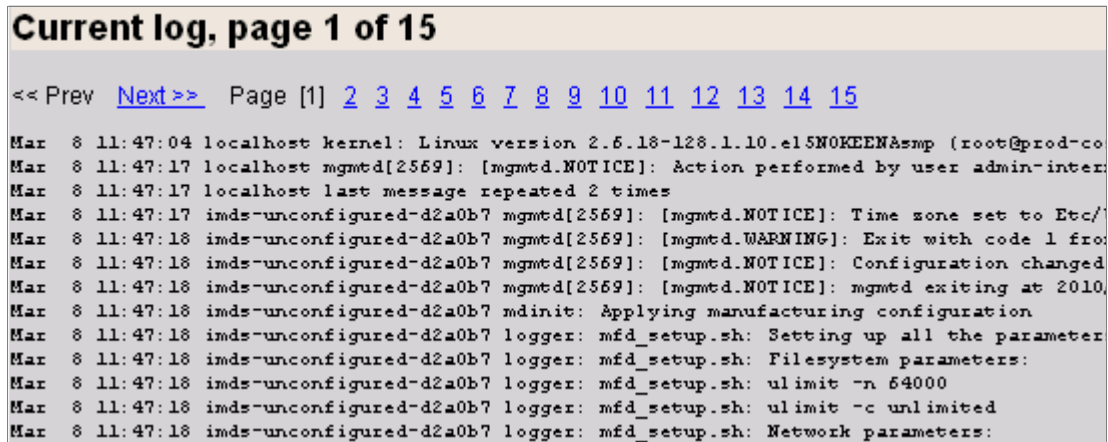
Archived (does not display if not applicable) log (1 - n), past logs.

Example:

```
Jan 6 00:10:00 MFC httpd: [Wed Jan 06 00:10:00 2010] [notice] Apache configured -- resuming normal
operations
```

Fields (some may not display):

Date_and_Time System_Hostname_and_Service_Name: [Process_Date_and_Time]
[Severity_Level] Event



```

Current log, page 1 of 15
<< Prev Next>> Page [1] 2 3 4 5 6 7 8 9 10 11 12 13 14 15
Mar 8 11:47:04 localhost kernel: Linux version 2.6.18-128.1.10.el5NOKEEMAsmp (root@prod-co
Mar 8 11:47:17 localhost mgmtd[2569]: [mgmtd.NOTICE]: Action performed by user admin-inter
Mar 8 11:47:17 localhost last message repeated 2 times
Mar 8 11:47:17 imds-unconfigured-d2a0b7 mgmtd[2569]: [mgmtd.NOTICE]: Time zone set to Etc/
Mar 8 11:47:18 imds-unconfigured-d2a0b7 mgmtd[2569]: [mgmtd.WARNING]: Exit with code 1 fro
Mar 8 11:47:18 imds-unconfigured-d2a0b7 mgmtd[2569]: [mgmtd.NOTICE]: Configuration changed
Mar 8 11:47:18 imds-unconfigured-d2a0b7 mgmtd[2569]: [mgmtd.NOTICE]: mgmtd exiting at 2010
Mar 8 11:47:18 imds-unconfigured-d2a0b7 mdinit: Applying manufacturing configuration
Mar 8 11:47:18 imds-unconfigured-d2a0b7 logger: mfd_setup.sh: Setting up all the parameter
Mar 8 11:47:18 imds-unconfigured-d2a0b7 logger: mfd_setup.sh: Filesystem parameters:
Mar 8 11:47:18 imds-unconfigured-d2a0b7 logger: mfd_setup.sh: ulimit -n 64000
Mar 8 11:47:18 imds-unconfigured-d2a0b7 logger: mfd_setup.sh: ulimit -c unlimited
Mar 8 11:47:18 imds-unconfigured-d2a0b7 logger: mfd_setup.sh: Network parameters:

```

Figure 114 Media Flow Controller Example **Current** Log

Viewing the Service Log (Web Interface)

The Media Flow Controller service (access) log records each HTTP transaction going through Media Flow Controller; in the CLI this is the [accesslog](#). See [“Reading the Service Log \(accesslog\)” on page 190](#) for details including Status Codes and Status Sub Codes.

Continuous log, updated every 10 seconds.

Current log, that day's activity.

Download log—Opens a **File Open** dialog so you can save the trace log locally.

Archived (does not display if not applicable) log (1 - n), past logs.

Example:

```
172.19.172.192 172.19.172.130 - [04/Jan/2010:20:33:40 +0000] "GET /HNAP1/ HTTP/1.1" 404 0 "-"
"Mozilla/4.0 (compatible; MSIE 5.5; Win32)" 52004
```

Fields:

Client IP Media Flow Controller IP - [Date_and_Time] "Method_and_Path_of_File"
Status_Code Bytes_Returned "Referrer" "Agent" Status_Sub_Code

Viewing the Cache Log (Web Interface)

The Media Flow Controller Cache log records all cache related activity; in the CLI this is the [cachelog](#). Use this tab to view Media Flow Controller caching events. See [“Reading the Cache Log \(cachelog\)” on page 190](#) for details including event types.

Continuous log, updated every 10 seconds.

Current log, that day's activity.

Download log—Opens a **File Open** dialog so you can save the trace log locally.

Archived (does not display if not applicable) log (1 - n), past logs.

Example:

```
[Fri Jun 26 19:37:09.754 2009] ADD "/http-cl18:ed239a85/100k-files/117/29" SAS dc_3 32768 [Fri Jun 26 19:38:14 2009]
```

Fields (display according to event type):

| [Date] | Event_Type | "<URI_name>" | Cache_Tier_Name | Cache_Name |
|-------------------------|----------------------------|--------------|-----------------|------------|
| Content_Length_In_Bytes | [Expiry_time_for_this_URI] | | | |

Viewing the Trace Log (Web Interface)

Media Flow Controller includes a delivery trace facility to help diagnose the handling of a particular HTTP request; trace results are written to the Trace Log; in the CLI this is the **tracelog**. See [“Reading the Trace Log \(tracelog\)” on page 195](#) for details including trace points.

Continuous log, updated every 10 seconds.

Current log, that day's activity.

Download log—Opens a **File Open** dialog so you can save the trace log locally.

Archived (does not display if not applicable) log (1 - n), past logs.

Viewing the Stream Log (Web Interface)

This Media Flow Controller service log records RTSP streaming transactions; in the CLI this is the **streamlog**. Use this tab to access the Media Flow Controller Service log. See [“Reading the Stream Log \(streamlog\)” on page 194](#) for details.

Continuous log, updated every 10 seconds.

Current log, that day's activity.

Download log—Opens a **File Open** dialog so you can save the trace log locally.

Archived (does not display if not applicable) log (1 - n), past logs.

Viewing the FMSCollector Log (Web Interface)

This Media Flow Controller service log records RTSP transaction details including what URIs are accessed and how many bytes are returned by the FUSE module. In the CLI this is the **fuselogs**. Use this tab to access the Media Flow Controller Service log. See [“Reading the FMSCollector Log / fuselogs” on page 194](#) for details.

Continuous log, updated every 10 seconds.

Current log, that day's activity.

Download log—Opens a **File Open** dialog so you can save the trace log locally.

Archived (does not display if not applicable) log (1 - n), past logs.

Viewing the FMSAccess Log (Web Interface)

This Media Flow Controller service log lists all FMS server command executions. Streaming fmsaccesslog events include play, pause, seek, and stop events; session fmsaccesslog

events include connect, disconnect, and connect-pending events, by default. This log is generated by the FMS server (must be installed, see [“Installing and Using FMS in Media Flow Controller \(CLI\)” on page 110](#)). In the CLI this is the **fmsaccesslog**. Use this tab to access the Media Flow Controller Service log. See [“Reading the FMSAccess Log \(fmsaccesslog\)” on page 192](#) for details.

Continuous log, updated every 10 seconds.

Current log, that day's activity.

Download log—Opens a **File Open** dialog so you can save the trace log locally.

Archived (does not display if not applicable) log (1 - n), past logs.

Viewing the FMSEdge Log (Web Interface)

This Media Flow Controller service log lists transactions to the FMS edge server; for example, “Connection rejected by server,” “Edge disconnected from core,” “Listener started for clients.” This log is generated by the FMS server (must be installed, see [“Installing and Using FMS in Media Flow Controller \(CLI\)” on page 110](#)). In the CLI this is the **fmsedgelog**. Use this tab to access the Media Flow Controller Service log. See [“Reading the FMSEdge Log \(fmsedgelog\)” on page 193](#) for details.

Continuous log, updated every 10 seconds.

Current log, that day's activity.

Download log—Opens a **File Open** dialog so you can save the trace log locally.

Archived (does not display if not applicable) log (1 - n), past logs.

Viewing the Dashboard Overview

Media Flow Controller supplies several dashboards. [Figure 115](#) shows the **Dashboard** menu.

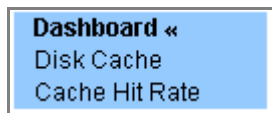


Figure 115 Dashboard Tab Left Navigation Menu

To view the dashboards see:

- [“Dashboard” on page 318](#)
- [“Dashboard: Disk Cache” on page 321](#)
- [“Dashboard: Cache Hit Rate” on page 322](#)

Dashboard

This is the opening dashboard with system usage information.

Statistics

- **Cumulative since**—The time since this Media Flow Controller has been running without reboot or shutdown.
- **GB delivered**—The total byte count of all objects Media Flow Controller has delivered since running.

- **Byte hit ratio**—The percentage of cumulative data (bandwidth) served from RAM/Disk/TFM compared to Total Data Served since the start of Media Flow Controller.
- **Cache hit ratio**—The number of objects Media Flow Controller served from RAM or Disk divided by the total number of objects served.
 - **Bandwidth**—Total number of bytes delivered from RAM or Disk / Total number of bytes delivered.
 - **Number of Requests**—Total number of objects delivered from RAM or Disk / Total number of objects delivered (irrespective of size).
- **Objects Delivered**—The total number of objects served by this Media Flow Controller since running.

Graphs

See [Figure 116](#).

- **Active Sessions**— Media Flow Controller connections to the client, both HTTP and RTSP; and origin manager connections (**om-session**).
- **Weekly Bandwidth Savings**—Saved bandwidth is bandwidth used by traffic that did not come from origin.
- **Cache Throughput**—Bandwidth and place from which data was served.
- **Cache Tier Throughput**—Green is served from cache, Yellow is indicates cache promotion, Red is evicted from cache.

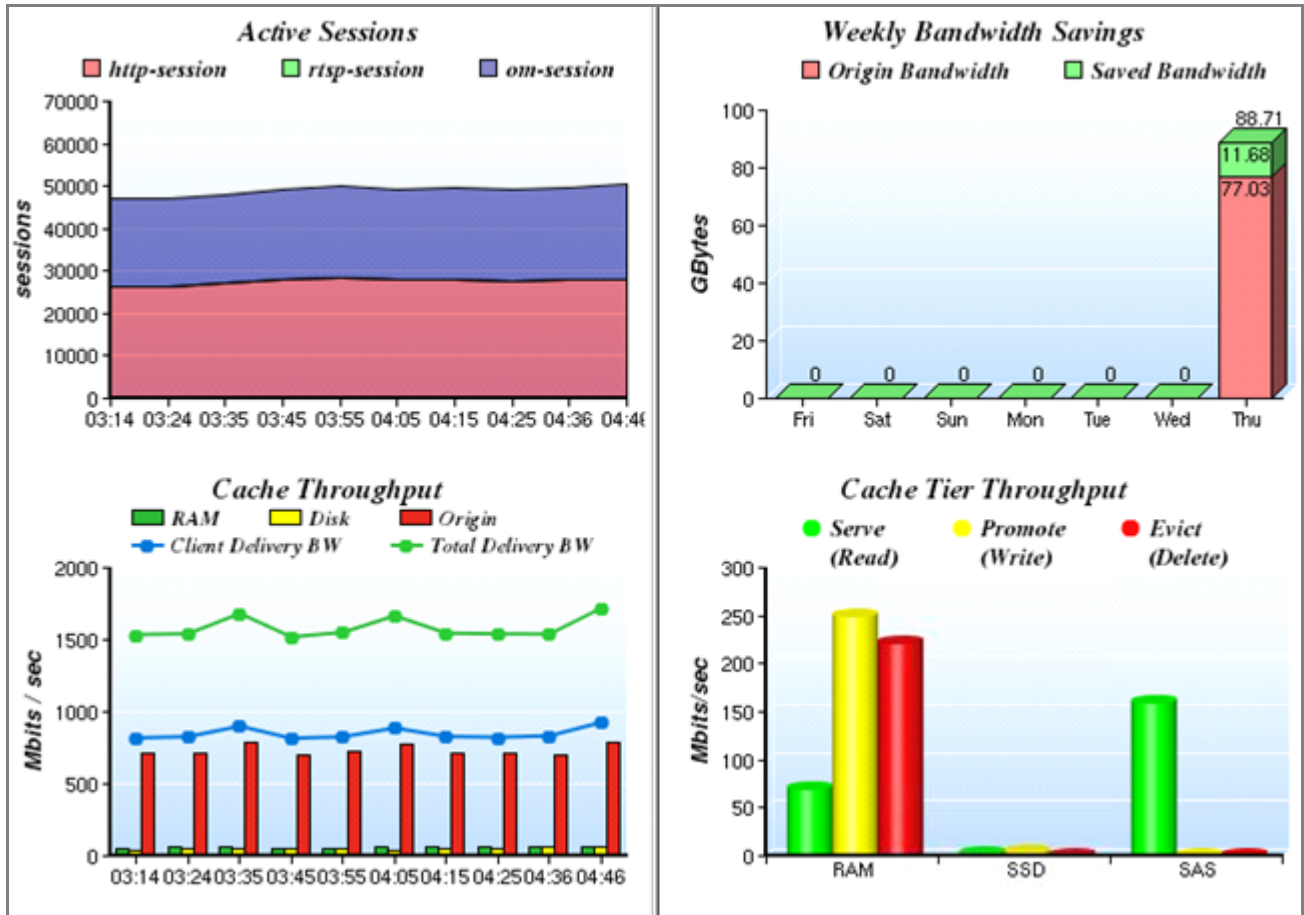


Figure 116 Media Flow Controller Dashboard Example

Dashboard: Disk Cache

See [Figure 117](#). **Disk Cache** page example; each disk is shown.

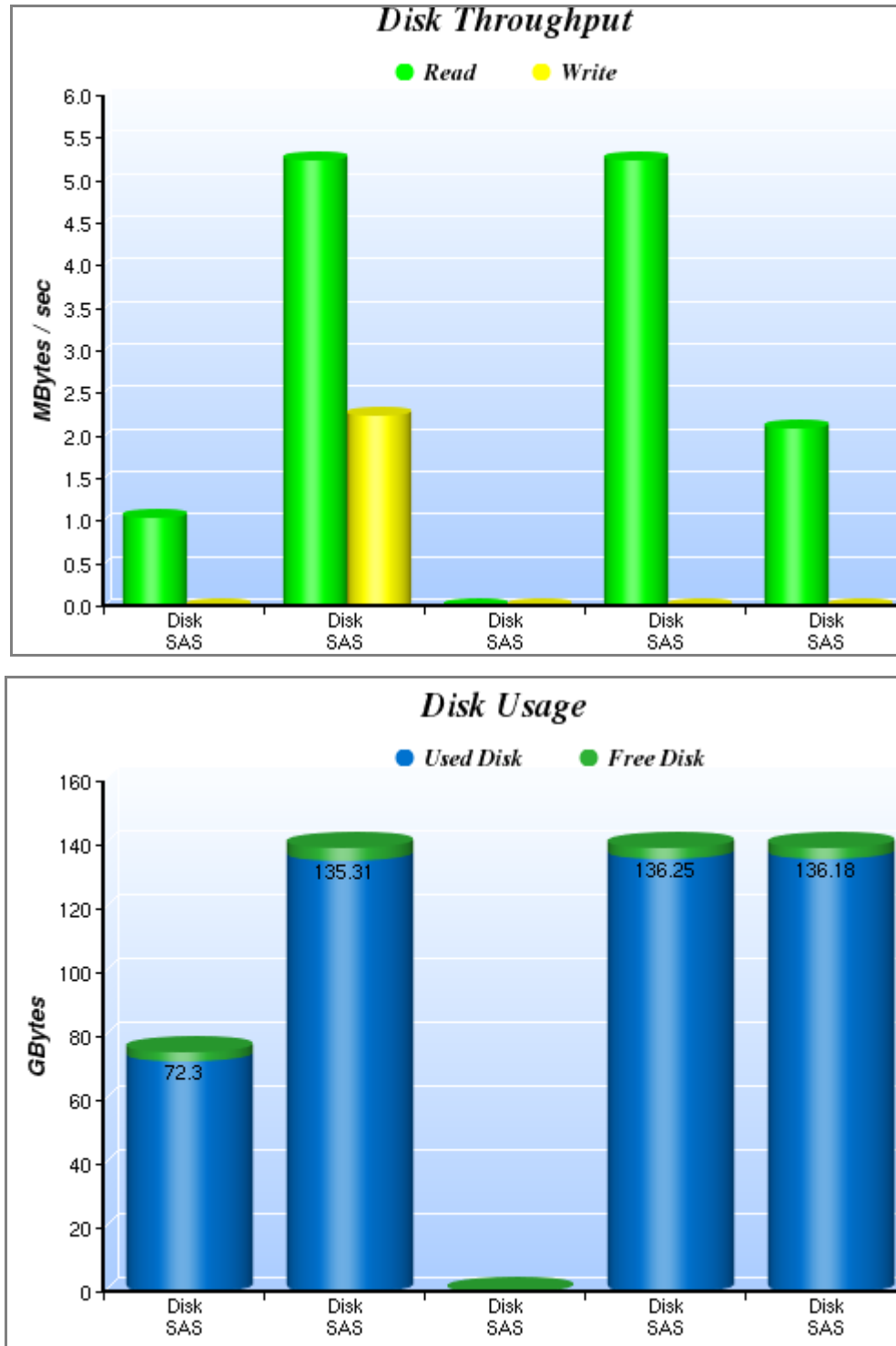


Figure 117 Media Flow Controller Disk Cache Graph Example

Dashboard: Cache Hit Rate

See [Figure 118](#). The **Cache Hit Rate** graph displays an hourly cache-hit ratio for the last 24 hour.

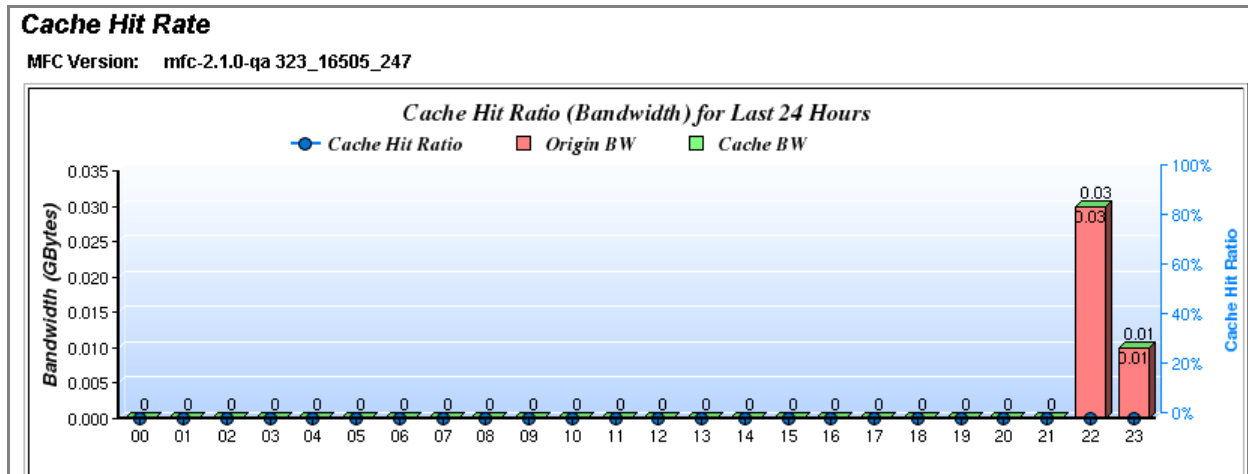


Figure 118 Media Flow Controller Log Analysis Page Example, Domain Hotness Analysis

Viewing Reports (Interface Statistics)

The **Reports** tab provides **Interface** and **Bandwidth** usage statistics. [Figure 119](#) shows the **Reports** menu.

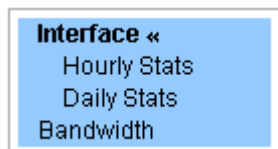


Figure 119 Reports tab left navigation menu

To view the reports see:

- [“Network Usage Last 24 hours” on page 322](#)
- [“Network Usage Hourly Stats” on page 324](#)
- [“Bandwidth Usage Last 24 hours” on page 325](#)

Network Usage Last 24 hours

See [Figure 120](#). Go to **Reports** to view network usage reports for the Last 24 hours and Last 7 days (not shown).

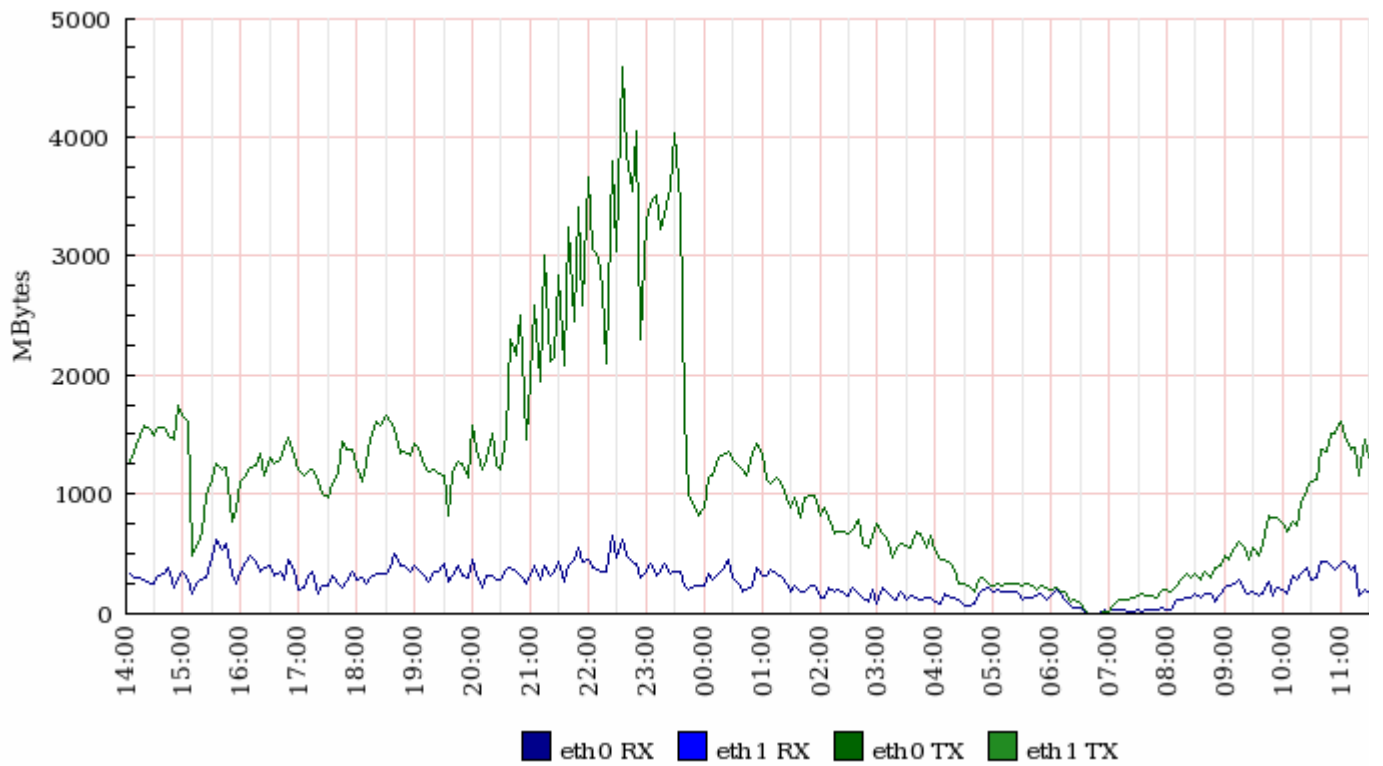


Figure 120 Network Usage Report

Network Usage Hourly Stats

See [Figure 121](#). Go to **Reports > Hourly Stats** to view network usage hourly stats reports and network usage daily stats reports (not shown).

Last 24 Hours of Activity

Interface Name: ▼

| Date and Time | Received (MBytes) | Transmitted (MBytes) |
|----------------------|-------------------|----------------------|
| 23-Mar-2011 01:30:02 | 2.36 MB | 0.07 MB |
| 23-Mar-2011 02:30:02 | 4.95 MB | 0.00 MB |
| 23-Mar-2011 03:30:02 | 4.96 MB | 0.00 MB |
| 23-Mar-2011 04:30:01 | 4.99 MB | 0.00 MB |
| 23-Mar-2011 05:30:03 | 5.09 MB | 0.07 MB |
| 23-Mar-2011 06:30:02 | 4.99 MB | 0.00 MB |
| 23-Mar-2011 07:30:02 | 5.03 MB | 0.00 MB |
| 23-Mar-2011 08:30:02 | 4.96 MB | 0.08 MB |
| 23-Mar-2011 09:30:02 | 4.94 MB | 0.00 MB |
| 23-Mar-2011 10:30:02 | 4.91 MB | 0.00 MB |
| 23-Mar-2011 11:30:02 | 4.93 MB | 0.00 MB |
| 23-Mar-2011 12:30:02 | 4.98 MB | 0.00 MB |
| 23-Mar-2011 13:30:02 | 4.97 MB | 0.00 MB |
| 23-Mar-2011 14:30:02 | 4.92 MB | 0.00 MB |
| 23-Mar-2011 15:30:02 | 4.91 MB | 0.00 MB |
| 23-Mar-2011 16:30:03 | 4.87 MB | 0.08 MB |
| 23-Mar-2011 17:30:02 | 4.99 MB | 0.00 MB |
| 23-Mar-2011 18:30:02 | 4.98 MB | 0.08 MB |
| 23-Mar-2011 19:30:02 | 4.97 MB | 0.00 MB |
| 23-Mar-2011 20:30:01 | 4.92 MB | 0.01 MB |
| 23-Mar-2011 21:30:02 | 5.04 MB | 0.05 MB |
| 23-Mar-2011 22:30:00 | 5.11 MB | 0.96 MB |
| 23-Mar-2011 23:30:00 | 5.09 MB | 1.45 MB |

Figure 121 Network Usage Daily Stats Last 24 Hours of Activity

Bandwidth Usage Last 24 hours

See [Figure 122](#). You can view bandwidth usage reports for the Last 24 hours and Last 7 days (not shown).

Go to **Reports > Bandwidth** to see bandwidth usage towards the client and from the origin

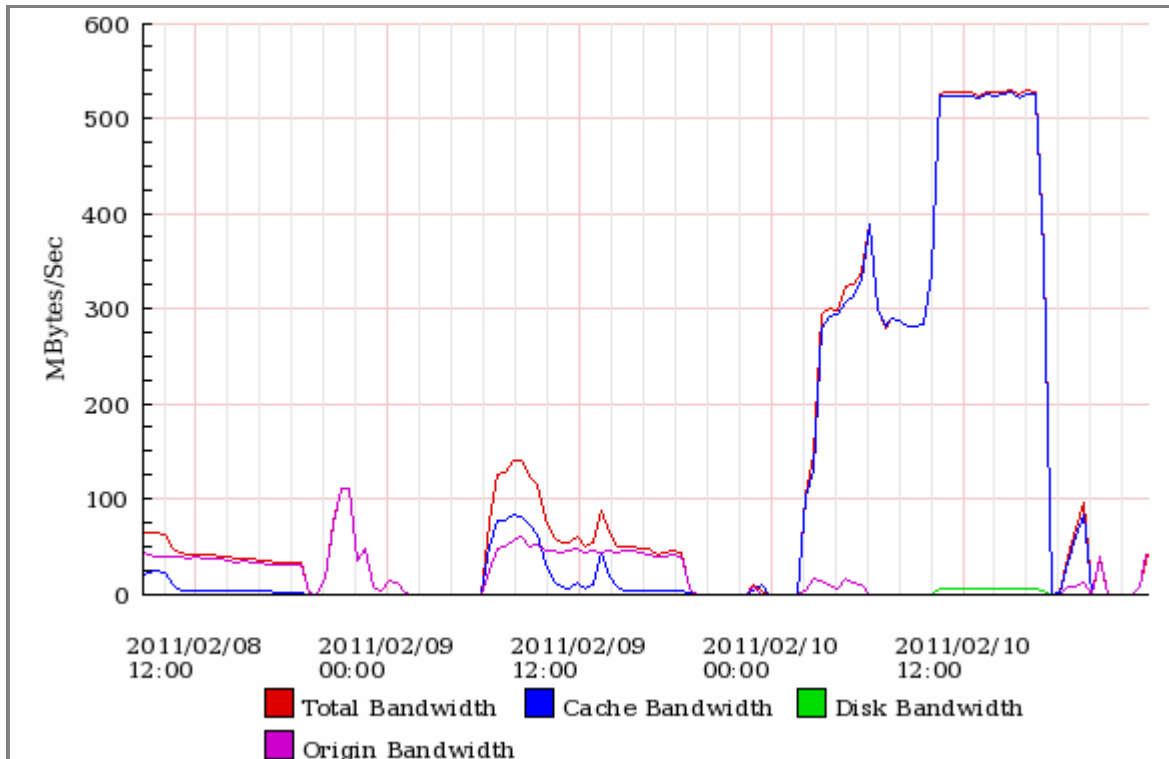


Figure 122 Bandwidth Usage Report

CHAPTER 12

SNMP Support

- [“About SNMP and Media Flow Controller” on page 327](#)
- [“SNMP Protocol Support” on page 327](#)
- [“Configuring the SNMP Agent \(Web Interface\)” on page 328](#)
- [“Configuring the SNMP Agent \(CLI\)” on page 331](#)
- [“Configuring Media Flow Controller SNMP and SNMP Alarms” on page 331](#)



NOTE: See [Media Flow Technical Documentation Enterprise MIBs](#) for Media Flow Controller MIB files.

About SNMP and Media Flow Controller

The Simple Network Management Protocol (SNMP) is a network management protocol used to monitor network-attached devices for fault conditions (problems). An SNMP-managed network consists of three key components: the managed device or devices, the “agent” (configured software on managed devices), and the network management system (NMS) software. Media Flow Controller provides an SNMP agent that can be configured to monitor various system statistics and parameters.

Media Flow Controller’s implementation of SNMP is limited to monitoring. No SNMP configuration features are supported by Media Flow Controller. Table 26 describes the SNMP protocol features supported by Media Flow Controller.

For a broader introduction to SNMP, see a technical reference like “Essential SNMP” (O’Reilly 2005).

SNMP Protocol Support

[Table 26](#) describes the level of supported for different SNMP protocol features provided by the SNMP agent in Media Flow Controller.

Table 26 SNMP Protocol Support

| Feature | Support | Description |
|-------------------|-------------|--|
| Protocol Versions | v1, v2c, v3 | Media Flow Controller supports all three protocol versions, though it only exposes a subset of the full functionality of v3. |

Table 26 SNMP Protocol Support

| Feature | Support | Description |
|---------------------------|---------|---|
| Get GetNext GetBulk | yes | The SNMP interface in Media Flow Controller supports network monitoring of system information. These requests provide the standard mechanism for obtaining the values of SNMP variables and counters. |
| Set | no | The Media Flow Controller MIB and SNMP agent do not allow clients to set any variables or traps. The purpose of the SNMP interface in Media Flow Controller is limited to system monitoring. |

Media Flow Controller MIB Versions

Media Flow Controller 2.1 includes two similar MIBs. These are described in [Table 27](#).

Table 27 Media Flow Controller MIB Versions

| MIB | Object Identifier (OID) | Description |
|-----------------|-------------------------|---|
| TM-MIB | 35000 | Fully supported, but will not be expanded to support any new variables or counters. |
| JUNIPER-MFC-MIB | 2636 | Partially supported. This new MIB was designed to restructure and expand the information that Media Flow Controller provides. It follows the Juniper Networks enterprise OID scheme. Some counters and variables cannot be queried with this MIB. |

Configuring the SNMP Agent (Web Interface)

The procedures in the following sections demonstrate how to configure the SNMP agent.

- [Basic SNMP Agent Configuration](#)
- [Configuring Trap Sinks](#)
- [Adding a New Trap Sink](#)
- [Configuring SNMP](#)
- [snmp traps events](#)
- [SNMP Alarms](#)

Basic SNMP Agent Configuration

1. From the left navigation pane in the **System Config** tab, select **SNMP**. The **SNMP** page is displayed. See [Figure 123](#).

SNMP

SNMP Configuration

| | |
|-------------------------|-------------------------------------|
| Enable SNMP | <input checked="" type="checkbox"/> |
| Enable SNMP communities | <input checked="" type="checkbox"/> |
| Enable SNMP traps | <input checked="" type="checkbox"/> |
| Sys Contact | <input type="text"/> |
| Sys Location | <input type="text"/> |
| Read-Only Community | <input type="text" value="public"/> |
| Default Trap Community | <input type="text" value="public"/> |

Figure 123 System Config > SNMP Page

2. Enable the following:
 - **Enable SNMP**—Enable the SNMP server. Un-check to disable; this stops serving SNMP variables and the sending of SNMP traps.
 - **Enable Communities**—Enable or disable (by un-checking) community-based authentication on this system; the SNMP "public" community is enabled by default. If disabled, the community configured is ignored.
 - **Enable Traps**—Enable or disable (by un-checking) sending SNMP traps from this system. The SNMP server must be enabled first. See [snmp traps](#) for details.
3. Enter this information to the text boxes:
 - **Sys Contact**—Set the **syscontact** variable served from the System MIB in MIB-II.
 - **Sys Location**—Set the **syslocation** variable served from the System MIB in MIB-II.
 - **Read-Only Community**—Set a name for read-only (**ro**) SNMP requests; this means only queries are performed. In Release 2.0.7, only SNMP **ro** is supported.
 - **Default Trap Community**—The string used if no string has been set for the trap.
4. Click **Apply** to complete SNMP configuration, **Cancel** to revert to existing configuration.
5. Click **Save** at the top of the page to make changes persistent across reboots.

Configuring Trap Sinks

An SNMP “trap” is an unsolicited message issued by a managed device to the management station; a “trap sink” is where the trap is sent. See [Figure 124](#). See also [snmp traps events](#).

Figure 124 System Config > SNMP Page

To view, **Remove**, **Enable**, or **Disable** trap sinks; trap sinks are created enabled.

- View this information on configured Trap Sinks:
 - Host**—The SNMP server host for the Trap Sink.
 - Community**—The SNMP community of the Trap Sink.
 - Version**—The SNMP version of the Trap Sink.
 - Enabled**—Whether or not this Trap Sink is enabled. Disabling a Trap Sink makes it inactive but does not delete it from the system.
- Select a Trap Sink and **Remove Trap Sink**, **Enable Trap Sink**, or **Disable Trap Sink**.
- Click **Save** at the top of the page to make changes persistent across reboots.

Adding a New Trap Sink

Add hosts to receive traps. See [Figure 125](#).

Figure 125 System Config > SNMP Page

To add a new Trap Sink:

- Enter this information to the text boxes:
 - Trap Sink IP**—Add hosts to receive SNMP traps.
 - Community**—Which community of traps to send to this host; default is "public."
 - Trap Type**—The SNMP version of traps to send to this host; choices are **v1** or **v2c**.
 - Click **Add New Trap Sink** to complete adding the new trap sink.
- Click **Save** at the top of the page to make changes persistent across reboots.

Configuring the SNMP Agent (CLI)

You can configure the SNMP agent running in the Media Flow Controller to integrate with third-party Network Management Systems (NMS). The following are the key Media Flow Controller configuration items:

- Configure the NMS IP address:
`snmp host <NMS_IP_address>`
- Configure the community string, a shared secret between the NMS and Media Flow Controller:
`snmp host <NMS_IP_address> traps <community_string>`
- Enable generation of traps in the Media Flow Controller:
`snmp enable traps`
- Configure the events for which traps have to be generated and sent to the NMS:
`snmp traps event <trap_event>`

See also [snmp traps events](#).

Configuring Media Flow Controller SNMP and SNMP Alarms

Media Flow Controller has SNMP agents that can respond to SNMP operations. An SNMP manager can ask Media Flow Controller for information and generate notifications of SNMP events to configured notify recipients.

Configuring SNMP

Configuring SNMP on Media Flow Controller consists of enabling the SNMP server, configuring community strings, traps, traps events, and hosts (also referred to as trap sinks) to receive traps, as well as authentication parameters. For more information on SNMPv2, see [RFC 2578](#).

To configure SNMP:

1. Configure SNMP hosts, or trap sinks, to receive SNMP query responses, or traps, sent from Media Flow Controller; after configured, the host is enabled, you can disable it with **snmp-server host <IP_address> disable** to temporarily stop Media Flow Controller sending it traps. You need to know the trap **version** and **community** for the traps to be sent to that host; the **community** identifies who can ask whom what.

```
snmp-server host <IP_address>  
snmp-server host traps version <version> <community_string>
```

The traps associated with that version and community are sent to that host.

2. Enable community-based authentication if you intend to use communities other than the default **public** community, and the sending of traps (traps are not sent until a host to receive them is configured). The Media Flow Controller SNMP server function is enabled by default; you can disable it with **no snmp-server enable**, which stops the serving of SNMP variables and sending of traps.

```
snmp-server enable communities  
snmp-server enable traps
```

The default community, **public**, is enabled and additional communities may be configured. These traps are sent to the enabled host configured for those traps and communities: **cold boot** (may include SNMP configuration changes), **link up or down**, **CPU load too high**, **paging activity too high**, **a process crash**, **a process unexpected exited**.

- SNMP communities determine which agents can make queries to which SNMP servers. There are some pre-defined communities, like the default **public** community, or you can create community names yourself; if the name contains spaces, enclose it in quotes. Media Flow Controller only allows read-only, **ro**, communities that permit only SNMP queries, or GETs. SNMP SETs for configuration changes are not allowed. Only the agents querying from a community configured in the Media Flow Controller SNMP server community list can make queries to that Media Flow Controller SNMP server. To enable read-only, **ro**, communities:

```
snmp-server community <community_string> ro
```

The traps associated with that community are sent to the **host**, or trap sink, enabled and configured with that community and for those traps.

- Configure SNMP listen interfaces to restrict access to a configured list of interfaces. The list is enabled by default; after you add an interface, SNMP queries are restricted to the interface or interfaces in the listen interfaces list. These interfaces should be statically configured with DHCP and zeroconf disabled. You can disable the list with **no snmp-server listen enable**. To add listen interfaces:

```
snmp-server listen <interface_ID>
```

Only the configured interfaces listen for SNMP queries. If the interface is also running as a DHCP client, it is as if the interface was not added. If DHCP is later turned off on this interface, it is as if the interface was then added to the listen list.

- Set the SNMP server contact and location; these are the syscontact and syslocation variables served from the system MIB in MIB-II.

```
snmp-server contact <contact_name>
snmp-server location <location_of_system>
```

The manager of this SNMP server is identified. The SNMP server location is identified.

- Specify which types of events to send as SNMP traps. By default, the entire list of notify-able events are sent as SNMP traps to any configured trap sinks. See [“SNMP Traps Notify-able Events” on page 333](#) for event names.

```
snmp-server traps event <event_name>
```

Only the configured traps events are sent.

- Specify who should receive traps. See [“Configuring Media Flow Controller Fault Notifications \(CLI\)” on page 208](#) for details on additional options.

```
email notify recipient <email_address>
```

Traps events are sent to the configured e-mail address.

snmp traps events

Media Flow Controller generates a number of traps to notify you about critical system events. You can configure Media Flow Controller to send SNMP traps/alarms to a 3rd party network management system. SNMP traps notify-able events are described in [Table 28](#).

Table 28 SNMP Traps Notify-able Events

| Trap | Description |
|----------------------------------|---|
| <code>cpu-util-high</code> | CPU utilization has risen too high. |
| <code>cpu-util-ave-ok</code> | CPU utilization has fallen back to acceptable levels. |
| <code>disk-io-high</code> | Disk I/O per second has risen too high. |
| <code>disk-space-low</code> | Filesystem free space has fallen too low. |
| <code>interface-down</code> | An interface's link state has changed to down. |
| <code>interface-up</code> | An interface's link state has changed to up. |
| <code>liveness-failure</code> | A process in the system was detected as hung. |
| <code>memusage-high</code> | Memory usage has risen too high. |
| <code>netusage-high</code> | Network utilization has risen too high. |
| <code>paging-high</code> | Paging activity has risen too high. |
| <code>process-crash</code> | A process in the system has crashed. |
| <code>process-exit</code> | A process in the system unexpectedly exited. |
| <code>smart-warning</code> | Smartd warnings. |
| <code>unexpected-shutdown</code> | Unexpected shutdown. |

SNMP Alarms

This section describes the SNMP alarms that may reach you through an [email event name Options](#) (configurable).

For Release 2.0.7, Media Flow Controller supports the **Entity** and **Asset** SNMP MIBs for discovery, asset management, alarms, and traps; that is HOST-REOURCES, HOST-RESOURCES-TYPES, IF, IP, IDP, and TCP MIBs. To view these MIBs, go to the Customer Support Website.

Typically, SNMP uses UDP (User Datagram Protocol) ports 161 for the agent and 162 for the manager. The Manager may send requests from any available port (source port) to port 161 in the agent (destination port). The agent response is given back to the source port. The Manager receives traps on port 162. The agent may generate traps from any available port. See [Table 29](#), for information on SNMP traps, possible causes, and recommended actions.



NOTE: At this time, Media Flow Controller does not support provisioning over SNMP V3.

Table 29 SNMP Alarms, Possible Causes, and Recommended Actions

| Event Name (MIB/ stat Name) | Cause | Action |
|---|--|---|
| System Traps | | |
| Interface UP/Down (cpu) | The network interface on Media Flow Controller went operationally UP or DOWN. This event could happen because a wire was unplugged, faulty cable, loose cabling, or network configuration issue. This event causes high-availability to kick in and requests may be forwarded to another Media Flow Controller or POP. | Check the network cables that go into the NIC card and make sure that all cabling is OK. Check the network configuration between the switch, or router, and server. Make sure that the interface speeds match on the switch and router because this mismatch could cause speed auto-negotiation problems. |
| Process Crashed (procmgr) | The Media Flow Controller process can restart if there is a software problem. A core file is generated. This event causes high-availability to kick in and requests may be forwarded to another Media Flow Controller or POP. The crashed Media Flow Controller process should restart very quickly after which new connections/ requests are served. If the core Media Flow Controller server process does not crash, requests are served in a normal manner. For example, if the management process restarts, requests are not affected. | Escalate to Juniper Networks Support to look at the core file. However, request processing may proceed automatically because the Media Flow Controller process restarts on its own. |
| Process Got Stuck (proclivenessFailure) | The Media Flow Controller is not serving out any more requests. This event causes high-availability to kick in and requests are forwarded to another Media Flow Controller or POP. This Media Flow Controller may not be able to recover. | Processing may proceed automatically because the Media Flow Controller process restarts on its own; if it does not, escalate to Juniper Networks Support to diagnose the issue. |
| Process Not Running (procExit) | The Media Flow Controller process is not serving out any more requests. This event causes high-availability to kick in and requests are forwarded to another Media Flow Controller or POP. This Media Flow Controller may not be able to recover. | Processing may proceed automatically because the Media Flow Controller process restarts on its own; if it does not, escalate to Juniper Networks Support to diagnose the issue. |
| Unexpected Shutdown (unexpectedShutdown) | The server may unexpectedly shut down due to power reasons. This is a highly unlikely event. This event causes high-availability to kick in and requests are forwarded to another Media Flow Controller or POP. This Media Flow Controller may not be able to recover. | Check all power cables. Restart Media Flow Controller and all process and configurations restart normal operation. |
| Hardware Resources Traps | | |
| Disk Failure Warning (smartError) | One or more disks are issuing warnings that a disk sector was corrupted, un-writeable, or unreadable. Media Flow Controller still serves out requests from other disks. If the ROOT disk is corrupted, the system may not be functional and needs to be looked at. | Escalate to Juniper Networks Support to diagnose the issue. |

Table 29 SNMP Alarms, Possible Causes, and Recommended Actions (Continued)

| Event Name (MIB/ stat Name) | Cause | Action |
|---------------------------------------|--|--|
| CPU Utilization High (cpuUtilHigh) | The CPU utilization may be high due to increased processing of requests, disk I/O, or some error. When CPU utilization is high, Media Flow Controller performance may be compromised. This event may be transitory or permanent. | If high CPU utilization is permanent, escalate to Juniper Networks Support to look at the problem. |
| Disk Space Low (diskSpaceLow) | Disk space low may be reported on the ROOT disk because logs may be taking too much of disk space. This may cause the ROOT disk to un-writeable. | Upload access log, cache log, error logs to a networked server and delete unneeded logs from the Juniper Networks server. |
| Paging High (pagingActivityHigh) | Memory utilization may be high on the Media Flow Controller server. When this happens, Media Flow Controller performance may be compromised. | Escalate to Juniper Networks Support to look at the problem. |
| Disk I/O High (disk-io-high) | Disk I/O per second has risen above 5 MB. Indicates that more content is served from disks. | Check if the amount of hot (popular) content is more than the available RAM size. If the RAM sizing is inadequate, consider increase the RAM in the system. |
| Memory Usage High (memusage-high) | Memory usage has risen above 90% | Check if a single process is hogging the system memory. If so, restart the process. |
| Network Usage High (netusage-high) | Network utilization has risen too high | Check if Media Flow Controller is constantly fetching content from origin server. If so, verify the cache-ability parameters on the origin server (such as Cache-Control, Expiry, Last Modified Date). Check if Media Flow Controller is trying to upload access logs to an unavailable server. |
| Cache Resources Traps | | |
| Bandwidth Limit (total_byte_rate) | Configured bandwidth limit has crossed on this server | Configure traffic redirection thresholds in the external router or load balancer to reduce the number of requests redirected to the Media Flow Controller |
| Cache Bandwidth (cache_byte_rate) | Current Cache bandwidth usage is too high | Configure traffic redirection thresholds in the external router or load balancer to reduce the number of requests redirected to the Media Flow Controller |

Table 29 SNMP Alarms, Possible Causes, and Recommended Actions (Continued)

| Event Name (MIB/ stat Name) | Cause | Action |
|---|--|---|
| Origin Bandwidth (origin_byte_rate) | Current Origin bandwidth usage is too high | Check if disk cache and RAM cache are properly utilized by monitoring the dashboard. Media Flow Controller may be serving lots of long-tail content or the content cannot be accommodated in the RAM disk cache. Check if the available disk RAM cache sizing is adequate for the traffic load. |
| Disk Bandwidth (disk_byte_rate) | Current Disk BW usage is too high | Check if the amount of hot (popular) content is more than the available RAM size. If the RAM sizing is inadequate, consider increase the RAM in the system. |
| Averaged Cache Bandwidth (avg_cache_byte_rate) | Cache Average bandwidth usage, since the system uptime, is too high | Configure traffic redirection thresholds in the external router or load balancer to reduce the number of requests redirected to the Media Flow Controller. |
| Averaged Origin Bandwidth (avg_origin_byte_rate) | Origin Average bandwidth usage, since the system uptime, is too high | Check if disk cache and RAM cache are properly utilized by monitoring the dashboard. Media Flow Controller may be serving lots of long-tail content or the content cannot be accommodated in the RAM disk cache. Check if the available disk RAM cache sizing is adequate for the traffic load. |
| Connection Rate (connection_rate) | Incoming connection rate is too high | Configure traffic redirection thresholds in the external router or load balancer to reduce the number of requests redirected to the Media Flow Controller. |
| Transaction Rate (http_transaction_rate) | HTTP transaction rate is too high | Configure traffic redirection thresholds in the external router or load balancer to reduce the number of requests redirected to the Media Flow Controller. |
| Port Bandwidth (perportbyte_rate) | Network port bandwidth usage is high | Configure traffic redirection thresholds in the external router or load balancer to reduce the number of requests redirected to the Media Flow Controller. Configure additional network ports in Media Flow Controller to redistribute the requests evenly across the available ports. |

CHAPTER 13

Deploying SmoothFlow for Media Flow Controller

- [“SmoothFlow Deployment Overview” on page 337](#)
- [“Evaluating Your Needs” on page 338](#)
- [“Creating SmoothFlow Media Assets Overview” on page 343](#)
- [“Creating Assets Using an SaaS” on page 343](#)
- [“Creating On-Demand Assets” on page 352](#)
- [“Deploying the SmoothFlow Reference Client Player” on page 356](#)
- [“Deployment Checklist” on page 358](#)

SmoothFlow Deployment Overview

Deploying SmoothFlow (SF) involves several tasks at various points, some on your Media Flow Controller and some on a different system, these tasks are described below. See [“Media Flow Controller SmoothFlow” on page 49](#) for an overview of the function.



NOTE: This task list assumes that all Media Flow Controller system configurations have been done.

The SmoothFlow Implementation tasks are:

1. Evaluating your needs.
See [Evaluating Your Needs](#).
2. Configure Media Flow Controller for SmoothFlow.
See [Configuring Media Flow Controller for SmoothFlow \(CLI\)](#).
3. Create SmoothFlow Media Assets; you have two options:
 - [Creating Assets Using an SaaS](#)
 - [Creating On-Demand Assets](#)
4. (Optional) Deploy the SmoothFlow Reference Player
See [Deploying the SmoothFlow Reference Client Player](#).

Evaluating Your Needs

Know your business model; you use this to determine how many different bit-rate profiles you need per video. This largely depends on the quality of video delivery you want to provide. For example, if you are a Video On Demand (VOD) provider, and your customers are subscription-based, delivery quality is important and you will want to encode 5 or 7 different bit-rate profiles for each video (some providers encode 13 different bit-rate profiles). If you are a User Generated Content (UGC) provider, your revenue is derived from advertising (not subscriptions), and you must be careful with your storage and delivery costs, you may only want to encode three different bit-rate profiles for each video.

After you know how many, and at what rates, bit-rate profiles you want to create, you are ready to begin. An example of different bit-rate profile encoding schemes:

- A premium content owner for Standard Definition video might use the following (300, 500, 700, 900, 1100, 1400, 1700, and 2000) kbps encoding scheme.
- A premium content owner for High Definition video might use the following (500, 800, 1100, 1500, 1800, 2100, 2600, and 3000) kbps encoding scheme.
- A user generated content (UGC) content owner might use (300, 500, and 750) encoding scheme; usually UGC customers stay below 1000 kbps for any video.

Encoding Requirements

To enable SmoothFlow delivery, there are a set of media codec/container requirements and encoding guidelines you must observe when creating the different bit-rate profiles. The codec and containers supported for SmoothFlow delivery are given in [Table 30](#).

Table 30 Media Flow Controller Acceptable Containers and Codecs

| Container | Video Codec | Audio Codec |
|-----------|-------------|-------------|
| FLV | H.264 | MP3/AAC |
| MP4 | H.264 | AAC |



NOTE: Juniper Networks SmoothFlow supports legacy containers and codecs like H.263, VP6 for video, and MP3 for audio in FLV formats, but we recommend using the more optimal H.264/AAC combination instead.

General encoding requirements:

- The same audio codec and video codec should be used across the various bit-rate profiles.
- Maintain key frames (*IDR frames for H.264*) at the same points across the bit-rate profiles to allow sync for switching (*fixed GOP intervals*).
- Specific to H.264: The same encoding profile (encoding tools) tools should be used across the bit-rate profiles.
- Keep the audio bit-rate constant across all the bit-rate profiles.
- Maintain the same video resolution across all the bit-rate profiles. Flash run-time has a known issue that causes jerkiness when switching between different video resolutions.

Tip! We recommend having at least one key frame every 2 seconds as profiles can only be switched at key frames (the closer the key frame, the quicker SmoothFlow can adapt). A 2-second interval is not required, but a standard interval across all the bit-rate profiles is required. Key frames at intervals smaller than 2 seconds may impact the encoding performance and quality. Key frames at intervals larger than 10 seconds may severely impact adaptability due to network fluctuations.

Configuring Media Flow Controller for SmoothFlow (CLI)

The following Media Flow Controller configurations are required for SmoothFlow functioning. They must be done by the Media Flow Controller administrator and are included here as an example. See [virtual-player type smoothflow](#) for CLI details.

- [“Configuring SmoothFlow Virtual Player \(CLI\)” on page 339](#)
- [“Configuring SmoothFlow Namespaces \(CLI\)” on page 341](#)

Configuring SmoothFlow Virtual Player (CLI)

Configure **smoothflow** type virtual players if you want to use the SmoothFlow feature.



NOTE: All virtual-player parameters are disabled until a value is set.

To configure the **type smoothflow** (formerly Type 4) virtual player:

1. Configure a virtual player with a **name** and **type smoothflow** (enters you to virtual-player configuration mode).

```
virtual-player <name> type smoothflow
```
2. Configure the **control point** for signalling bandwidth changes at the client side; either **server** or **player** for SmoothFlow signaling. If **server**, Media Flow Controller detects the bandwidth variations at the client side and adjusts the bit-rate of the video accordingly. If **player**, the player at the client side explicitly signals the bandwidth changes and Media Flow Controller adjusts the bit-rate of the video accordingly.

```
control-point {player | server}
```
3. Set signal names for **session-id**, **state**, and **profile** delivery functions. Your client player must understand the query params you use; for example **sid** (for session-id), **sf** (for SF state), **pf** (for profile). See [“virtual-player type smoothflow” on page 378](#) for further information on SmoothFlow states.

```
signals session-id query-string-parm <string> state query-string-parm  
    <string> profile query-string-parm <string>
```
4. Configure hash verification options. **Note!** In Release 2.0.7, only **md-5** digest is supported. Set a **shared secret** value to be appended or prefixed to the URL as specified, for matching against the hash value provided in the URL and indicated by the **match query-string-parm** you configure.

```
hash-verify digest md-5 shared-secret <string> {append | prefix} match  
    query-string-parm <string>
```
5. Optionally, configure **connection max-bandwidth** delivery optimization. Default is **0** (unbounded) with the Media Flow Controller license, **200** kbps without it; you must have the license to change the unlicensed default. Use **no connection** to reset default.

```
connection max-bandwidth {0 | <kbps>}
```

6. Optionally, configure **seek** delivery optimization specifying a query param for when seek should start and how long it should last. Query params must be used.

```
seek query-string-param <string> [seek-length query-string-param <string>]
```

7. Verify configurations with **show virtual-player <name>**. Type **exit** to leave virtual-player configuration mode.

Example:

```
test-vos (config) # virtual-player testSF type smoothflow
test-vos (config virtual-player testSF) # control-point player
test-vos (config virtual-player testSF) # signals session-id query-string-
    parm sid state query-string-parm sf profile query-string-parm pf
test-vos (config virtual-player testSF) # hash-verify digest md-5 match
    query-string-parm h shared-secret zpzp prefix
test-vos (config virtual-player testSF) # connection max-bandwidth 0
test-vos (config virtual-player testSF) # seek query-string-parm toff
test-vos (config virtual-player testSF) # exit
test-vos (config) # show virtual-player testSF
```

```
Virtual Player : testSF
```

```
  Type : smoothflow
```

```
Control Point : player
```

```
Signals Configuration:
```

```
  Enabled: yes
```

```
  Session-Id string: sid
```

```
  State string: sf
```

```
  Profile string: pf
```

```
Hash Verify Configuration
```

```
  Enabled: yes
```

```
  Digest: md-5
```

```
  Data String:
```

```
  Data UOL Offset: 0
```

```
  Data UOL Length: 0
```

```
  Match query string: h
```

```
  Shared secret: ****
```

```
Connection Configuration
```

```
  Max Bandwidth: 0 kbps
```

```
Seek Configuration
```

```
  Enabled: yes
```

```
  URI Query: toff
```

```
test-vos (config) #
```

Configuring SmoothFlow Namespaces (CLI)

Another required SmoothFlow configuration is a **namespace** to which you add a configured SmoothFlow virtual player.



NOTE: SmoothFlow functioning requires at least two namespaces, one (or more) for SmoothFlow and one for the crossdomain policy with a lower (or lowest) **precedence**; see [Adobe Cross-domain Policy File Specification](#) for details.

1. Configure a namespace with a **name** (enters you to namespace configuration mode).
`namespace <name>`
2. Configure **delivery protocol** (HTTP and RTSP may be set) options (enters you to namespace delivery protocol mode, use **exit** to leave), and **domain** settings if needed (default is **any**) for the namespace. Only one **domain** per namespace is allowed; if multiple **domains** require namespaces, create additional namespaces with the same domain and set **match <criteria> precedence** appropriately (the lower the value, the higher the precedence).
`delivery protocol {http | rtsp}`
`domain <FQDN> [precedence <number>]`
3. While still in **delivery protocol http** mode (for both **delivery protocol http** and **delivery protocol rtsp**) configure **match** criteria for the namespace. See "[uri-prefix](#)" on page 34 for definition and example. **Note!** only **uri** and **virtual-host** are **delivery protocol rtsp match** options. You can optionally set a **precedence** for any **match** option.
`match {header | query-string | uri | virtual-host}`
4. Optionally, while still in **delivery protocol http** mode (for **delivery protocol http** only), configure **client-response** and **client-request** parameters.
`client-request [cookie | query-string] action {cache [exclude-query-string] | no-cache}`
`client-response [header <name> [<value>] action {add | delete}]`
5. Optionally, while still in **delivery protocol http** mode (for both **delivery protocol http** and **delivery protocol rtsp**) configure **origin-fetch** parameters for managing the cache, modifying the **Date** header, and fetching content in an offline manner; not needed for **crossdomain** namespace. Defaults are shown as first option.
`origin-fetch`
`cache-age {content-type<string><seconds> | content-type-any<seconds>}`
`cache-age-default {28800 | <seconds>}`
`cache-directive no-cache {follow | override}`
`cache-fill {aggressive | client-driven}`
`content-store media [cache-age-threshold<seconds>] [object-size<bytes>]`
`date-header modify {permit | deny}`
 If pre-staging to an edge, increase the **cache-age-default** to a higher value.
6. Exit **namespace delivery protocol** mode and return to **namespace** mode.
`exit`
7. Configure origin-server settings; for Release 2.0.7, multiple HTTP or NFS origin servers can be configured with the **server-map** option.
`origin-server http {absolute-url | follow {dest-ip | header <header> | server-map <name> | <FQDN/path [<port#>]}}`
8. Configure parameters for pre-staging content from origin; not needed for **crossdomain** namespace. Authentication schemes must be pre-configured to be used. The **ftp user** is

auto-generated as `<namespace>_ftpuser`, *without a password*. Set the password here; this entry overrides a `user <namespace>_ftpuser` password setting.

```
pre-stage ftp user <name> password {RADIUS | TACACS | <password>
    [encrypt]}
```

9. Add an existing **smoothflow virtual-player** to the new namespace; not needed for **crossdomain** namespace.


```
virtual-player <smoothflow_virtual-player_name>
```
10. Activate the namespace.


```
status active
```
11. Type **exit** to leave namespace configuration mode. Type **configuration write** to save the settings. Type **show namespace <name>** to see the configuration.

Example:

```
test-vos (config) # namespace test
test-vos (config namespace test) # match uri / precedence 3
test-vos (config namespace test) # domain any
test-vos (config namespace test) # origin-server http example.com 80
test-vos (config namespace test) # pre-stage ftp user test_ftpuser
    password 678
test-vos (config namespace test) # delivery protocol http origin-fetch
test-vos (config namespace test delivery protocol http origin-fetch) #
test-vos (config namespace test delivery protocol http origin-fetch) #
    cache-age-default 2880000
test-vos (config namespace test delivery protocol http origin-fetch) #
    cache-directive no-cache follow
test-vos (config namespace test delivery protocol http origin-fetch) #
    content-store media cache-age-threshold 60
test-vos (config namespace test delivery protocol http origin-fetch) #
    exit
test-vos (config namespace test) # virtual-player sfplayer
test-vos (config namespace test) # status active
test-vos (config namespace test) # exit
test-vos (config) # configuration write
test-vos (config) # show namespace test
```

```
Namespace: test
  Active: yes
  Precedence: 3
  Proxy Mode: reverse
  URI Prefix - /
  Origin-Server: http://example.com:80
    KeepAlive: no
    Weight: 50
  Virtual Player: sfplayer
  Origin Fetch Configuration:
    Cache-Age: 2880000 (seconds)
    Cache-Directive: follow
    Cache-Revalidate: yes
    Cache Age Threshold: 60 (seconds)
    Convert HEAD to GET: yes
    Object Size Threshold: NONE (Always Cached)
    Host Header Inherit: deny
```

```
Modify Date Header: yes
Offline File Fetch Size: 10000 (kbytes)
Pre-stage FTP Configuration:
User: test_ftpuser
```

Creating SmoothFlow Media Assets Overview

Media assets must be created and pre-staged, to origin. You have two options:

- [Creating Assets Using an SaaS](#)—Juniper Networks provides python publishing scripts that work with **encoding.com** Software as a Service (SaaS) to encode, process, and pre-stage data to a reliable origin server for delivery via Media Flow Controller. Additional encoding services will be supported in the future.
- [Creating On-Demand Assets](#)—You encode the media into multi-bit-rate profile files using the method of your choice, create the Asset Description file, pre-stage the assets to an origin server, and initiate SmoothFlow processing from Media Flow Controller so the asset is ready for SmoothFlow delivery when a request arrives (on-demand).

Creating Assets Using an SaaS

You begin by creating an account with encoding.com and posting a single video file to your Linux server. See [Figure 126](#) for an illustration.

Step 1 – [Initiating Encoding Using an SaaS](#) using the **SFAssetGenerator.py** script and XML files you create to encode the multi-bit-rate (MBR) profiles.

Step 2 – [Verifying that Encoding has Completed](#).

Step 3 – [Preparing Media Flow Controller for Assets Created Using an SaaS](#) using the **SFSegment.py** script and XML files you create to publish the assets to Media Flow Controller, and render them ready for delivery.

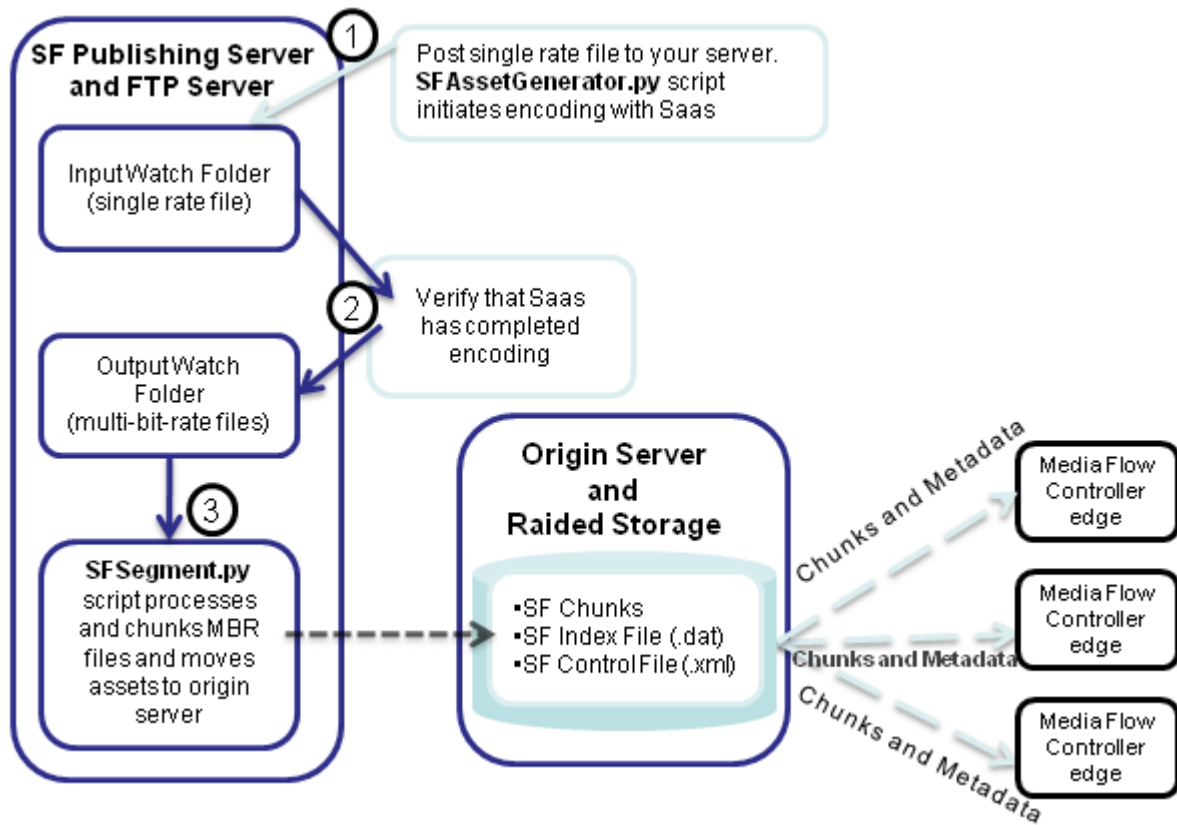


Figure 126 Publishing Workflow, Typical Steps

Before You Begin Creating Assets Using an SaaS

- You must create an account with encoding.com; you provide that login/ID to the SFAssetGenerator script.
- The scripts require Python to be installed in the machine that will run the scripts.
- The scripts currently run only on Linux-based systems.
- An FTP service on the system must be enabled and FTP credentials for the Output folder used must be available to the scripts.
- You must have proximity to an origin server system; the same system can be used as an origin server or the origin server can be NFS mounted to this system.

i **NOTE:** A typical configuration would have the FTP server, scripts, and origin server running on the same system or the origin storage can be NFS mounted on the publishing system.

NOTE: The entire multi-bit-rate publishing workflow is not fully automated and requires content owners to take two steps of manual intervention to trigger the scripts and configure the encoding parameters.

Steps for Creating Assets Using an SaaS

Steps 3 through 9 are described in this section.

1. Create an account with [encoding.com](#).
2. Post one single-bit-rate video file to your Linux server.
3. [Initiating Encoding Using an SaaS](#) involves creating a **setup.xml** file describing the Website of your chosen encoding service and your login credentials for the service, where you want the multi-bit-rate profiles output to, and credentials for that (for example, FTP site and credentials), and how often you want status polling done.
4. For each asset, create an **asset.xml** file describing the single-bit-rate video file, how many multi-bit-rate profiles you want created for it and at what rates, including frames-per-second, container format, and keyframe interval.
5. Run the **SFAssetGenerator.py** python script referencing the **setup.xml** and **asset.xml** files you created. This script outputs the **jobid.xml** file with descriptions of the multi-bit-rate profiles created by the SaaS.
6. [Verifying that Encoding has Completed](#) and the assets are transferred to your FTP output account given in your **setup.xml** file.
7. [Preparing Media Flow Controller for Assets Created Using an SaaS](#) involves creating the **segment_config.xml** file describing the multi-bit-rate assets for Media Flow Controller including the IP address of the Media Flow Controller, where the assets are located, access credentials for obtaining the assets, and a temporary directory in the system for SmoothFlow processing.
8. Run the **SFSegment.py** python script referencing the **segment_config.xml** file you created to initiate SmoothFlow processing and publishing to Media Flow Controller, and render the assets ready for delivery.
9. Check the [Logs for Assets Created Using an SaaS](#).

Using Scripts to Create Assets Using an SaaS

Juniper Networks has two Python scripts, SFAssetGenerator and SFSegmenter, that can be used to create media assets using [encoding.com](#) SaaS. Obtain these scripts through Juniper Networks Customer Support (see ["Requesting Technical Support" on page 35](#)).

- **SFAssetGenerator script** interfaces with [encoding.com](#) service and is responsible for the creation of the multiple bit-rate profiles in accordance with Media Flow Controller encoding guidelines. This script works with two XML files, **setup** and **asset**, and outputs one, **jobid**:
 - **setup.xml**—Specifies the global setup configuration parameters.
 - **asset.xml**—Specifies the encoding configuration parameters for each asset.
 - **jobid.xml**—Contains a list of IDs provided by [encoding.com](#) that are used as input to the SFSegmenter.py script.
- **SFSegmenter script** takes the prepared media assets, and SmoothFlow processes them, and then pre-stages them to a specified origin server, preparing the asset for delivery.

Initiating Encoding Using an SaaS

The SFAssetGenerator script works with two XML files, **setup.xml** and **asset.xml**, and outputs one file, **jobid.xml**. You use this script to provide the chosen encoding service with

information for the encoding to generate the multi-bit-rate assets. This section describes the three XML files and provides usage options. Initiating the encoding requires these steps:

1. Create a **setup.xml** file describing the Website of the encoding service and your login credentials plus where you want the multi-bit-rate profiles output to and credentials for that (for example, FTP site and credentials), and how often you want status polling done.
2. Create the **asset.xml** file describing each single-bit-rate video and how many multi-bit-rate profiles you want created and at what rates; including frames-per-second, container format, and keyframe interval.
3. Run the **SFAssetGenerator.py** python script to initiate the encoding of the referenced single-bit-rate videos into multi-bit-rate (MBR) profiles. This script outputs the **jobid.xml** file with descriptions of the multi-bit-rate profiles created by the SaaS. Use the **SFAssetGenerator.py** python script by running the **python** command at the command line and using the following tags. Example follows.

- **-s**—setup.xml
- **-a**—asset.xml
- **-o**—jobid.xml

Example:

```
python SFAssetGenerator.py -s <setup_xml_file> -a <asset_xml_file> -o
<jobid_xml_file>
```

setup.xml

This file specifies the global setup configuration parameters; variables are shown in curly brackets ({ }). Can be called **setup.xml** and its location specified via the **-s** command line parameter. If it is not specified, the script looks in the current folder for a **setup.xml** file. If the file is absent, the script throws an error and stops.

The configurable parameters for **setup.xml** are described in [Table 31](#), an example file follows.

Table 31 **setup.xml** File Parameters

| Parameters | Tags and Descriptions |
|--|--|
| <service_provider> | <name> – name of encoding service; for example, www.encoding.com . The "pre_process=1" corresponds to the encoding service; currently only encoding.com is supported. <login> – your encoding.com credentials |
| <ftp_source_location> <ftp_dest_location> | Information regarding the output profiles with following attributes <username> – the authorized FTP user <password> – the authorized FTP user password <server_ip> – FTP server IP address, source and destination, respectively, (if different) <path> – path in or out, respectively |
| <status_polling_duration> | How often you want progress status messages to display at the command line. |

```
< ?xml version="1.0"?>
<!--Global Setup Configuration Parameters-->
<setup_info>
```

```
<!--Provide URI and login credentials to access encoding service provider-->
<service_provider>
<name pre_process="1"> {name_of_encoding_service} </name>
<login userid={value} userkey={value}> </login>
</service_provider>

<!--Provide the source and destination locations for the FTP folders where the
files will be picked up, encoded, and the results transferred to-->

<ftp_source_location>
<username> {username} </username>
<password> {password} </password>
<server_ip> {ftp_source_server_IP} </server_ip>
<path> {/in} </path>
</ftp_source_location>

<ftp_dest_location>
<username> {username} </username>
<password> {password} </password>
<server_ip> {ftp_dest_server_IP} </server_ip>
<path> {/out} </path>
</ftp_dest_location>

<!-- polling interval for status messages at the command line, in seconds-->
<status_polling_duration> {9000} </status_polling_duration>
</setup_info>
```

asset.xml

This file specifies the encoding configuration parameters for each asset. The name of this XML file is specified via the `-a` command line parameter. The configurable parameters for `asset.xml` are described in [Table 32](#), followed by an example file.

Table 32 `asset.xml` File Parameters

| Parameters | Tags and Descriptions |
|--------------------|--|
| <asset_parameters> | <p><videoname> – Video name with full path, if not in the current directory of the script</p> <p><profile_cnt> – Number of multi bit-rate files that need to be generated (default is 3)</p> <p><profile_info> — Information regarding the output profiles with following attributes. The three default profiles that will be generated will be at a video bit-rate of 250 kbps, 500 kbps, 800 kbps and audio bit-rate 64 kbps (constant for all profiles)</p> <ul style="list-style-type: none"> • p – profile number • ab – audio bit-rate for profile • vb – video bit-rate for profile • width – pixel width for profile • height – pixel height for profile • deinterlace – option to convert interlaced video, like common analog television signals, into a non-interlaced form (for example, field to frame conversion) <p><fps> – Frames per second of the video</p> <p><format> – Container format for the encoded media (either flv, mp4)</p> <p><keyframe_interval> – Key frame positions</p> <p>The recommended keyframe interval is two seconds; however, this file specifies the keyframe interval in numbers of frames, not seconds. To have a two-second frame interval, make this value double the value entered for <code><fps></code> (frames per second).</p> |

```

< ?xml version="1.0"?>
<!--Asset Level Encoding Configuration Parameters-->
<asset_parameters>
<videoname> {video_file_name} </videoname>
<profile_cnt> {number_of_profiles} </profile_cnt>
<profile_info> p= {index} 1 ab= {audio-bitrate in kbps} vb= {video-bitrate in
kbps} width={width in pixels} height={height in pixels} deinterlace={true/
false , default = true} </profile_info>
<profile_info> p= {index} 1 ab= {audio-bitrate in kbps} vb= {video-bitrate in
kbps} width={width in pixels} height={height in pixels} deinterlace={true/
false, default = true} </profile_info>
<fps> 25</fps>
<format>flv</format>
<keyframe_interval>50</keyframe_interval>
</asset_parameters>
    
```

jobid.xml

The SFAssetGenerator.py script run with the **setup.xml** and **asset.xml** files, outputs this file, **jobid.xml**, listing IDs provided by encoding.com and used as input to **SFSegment.py**. It also provides the number of profiles (equals number of job IDs) and the bit-rates of each profile. The parameters for **jobid.xml** are described in [Table 33](#), followed by an example file.

Table 33 **jobid.xml** File Parameters

| Parameters | Tags and Descriptions |
|--------------|---|
| <video_list> | <p><video_details – Video name.</p> <p><profile_count> – Number of multi-bit-rate files that were generated.</p> <p><profile_list> – Information regarding the output profiles.</p> <p><profile_info> – Information regarding the output profiles with following attributes.</p> <ul style="list-style-type: none"> • p – profile number • audiobit – audio bit-rate for profile • videobit – video bit-rate for profile |

```
<?xml version="1.0"?>
<!-- video list -->
<video_list>
<!-- video name -->
<video_details video_name="sf-enc.mp4">
<profile_count>2
</profile_count>
<!-- profile info -->
<profile_list>
<profile_info p="1" audiobit="64" videobit="128"></profile_info>
<profile_info p="2" audiobit="64" videobit="256"></profile_info>
</profile_list>
</video_details>
</video_list>
```

Verifying that Encoding has Completed

In order to verify that the requested encoding has completed, you can look at your account with [encoding.com](#); use the Job ID given in the output of the python command to verify a particular set of encoding requests.

Or, by setting the **<status_polling_duration>** element in the **setup.xml** file, the SFAssetGenerator script can provide real-time status updates about the encoding progress. You will see status messages being displayed:

```
profile id: 1 status: downloading
profile id: 1 status: waiting for encoder
profile id: 1 status: processing
profile id: 1 status: saving
```

The status messages signify the following:

- **downloading**—Transferring the original source files into encoding.com local storage.

- **waiting for an encoder**—Waiting for encoding.com to assign an encoder from the cloud compute.
- **encoding in progress**—Encoding is under-way.
- **saving**—Transferring the encoded files back into user's FTP account.

After the requested bit-rate profiles and metadata files appear in the output folder, you are ready to request SmoothFlow publishing of the encoded assets.

Preparing Media Flow Controller for Assets Created Using an SaaS

The **SFSegment.py** script works with one XML file, **segment_config.xml**. You use this script to tell Media Flow Controller where the encoded multi-bit-rate file chunks are located. Preparing Media Flow Controller requires these steps:

1. Create the **segment_config.xml** file describing the multi-bit-rate assets for Media Flow Controller including the IP address of the Media Flow Controller, where the assets are located, access credentials for obtaining the assets, and a temporary directory in the system for SmoothFlow processing.
2. Run the **SFSegment.py** script to initiate SmoothFlow processing and publishing to Media Flow Controller, and render the assets ready for delivery. Use the **SFSegment.py** python script by running the **python** command at the command line and using the following tags.
 - **-s**—segment_config.xml
 - **-i**—jobid.xml
 - **-of**—output video format
 - **-af**—Assured Flow Rate (AFR) threshold (should be set at a value greater than one half the bit-rate of the highest bit-rate profile in the asset)

Example:

```
python SFSegment.py -s segment_config.xml
-i <jobid_xml_file_generated_by_SFAssetGen>
[-of <output_video_format>] [-af <AFR_threshold>]
```

segment_config.xml file

The configurable parameters for **segment_config.xml** are described in [Table 34](#), followed by an example file. The **<mfd-ip>** tag = Media Flow Controller IP address.

Table 34 **segment_config.xml** Parameters

| Parameters | Tags and Descriptions |
|---------------|---|
| <mfd-ip> | IP address of the Media Flow Controller which hosts the video content. This is needed to publish the HTML files containing the SmoothFlow player |
| <origin-path> | Path to which the segmented data needs to be copied. Needs to be the origin server's doc root (www/var/html) directory. This script can be run either on the origin server, or on another system where the doc root of the origin server is mounted as an NFS mount |

Table 34 **segment_config.xml** Parameters (Continued)

| Parameters | Tags and Descriptions |
|----------------|--|
| <source-type > | <p><ftp-location> – The location of the multi-bit-rate files, if the location of these files is different from the system where it is being run. This node requires the following additional sub-nodes:</p> <ul style="list-style-type: none"> <username> – the authorized FTP user <password> – the authorized FTP user password <server_ip> – FTP server IP address, source and destination, respectively, (if different) <path> – path in or out, respectively <p><local-path> – The location of the multi-bit-rate files if they are co-located in the same system where this script is run (default is current directory where script is run)</p> |
| <scratch_path> | A temporary directory on the system where the processing can take place; is automatically removed when the processing finishes |

```

< ?xml version="1.0"?>
<segment_config_info>

<!--IP address of the Media Flow Controller; required for generating the HTML
content-->
<mfd-ip> {IP_address_of_Media Flow Controller} </mfd-ip>

<!--Where the final chunks reside, can be local or nfs-->
<origin-path> {/var/www/html/smoothflow} </origin-path>

<!--Where the multi-bitrate files are present; can be local or FTP. If FTP,
then the script downloads the files to the given scratch-path. If local, then
the script copies the multi-bitrate files to the given scratch path-->
<source-type loc= ftp | local> <!--Choose either ftp or local-->

<ftp_location>
<username> {username} </username>
<password> {password} </password>
<server_ip> {ftp_server_IP} </server_ip>
<path> {/test_out} </path>
</ftp_location>

<!--If ftp service is running on the same system-- >
<local-path> {/root/test_out} </local-path>

</source-type>

<scratch_path> {temp_dir_removed_by_script_when_done} </scratch_path>
</segment_config_info>

```

Logs for Assets Created Using an SaaS

The system generates three log files when you use the encoding using an SaaS procedure:

- **pub-asset-encoder.log**—Captures the completion status of SFAssetGenerator.py.
- **pub-asset-polling.log**—Captures the multiple transaction IDs provided by encoding.com, and associates them with unique keys generated by SFAssetGenerator.py.
- **pub-asset-segmenter.log**—Captures the completion status of the SFSegment.py script.

Table 35

```
pub-asset-encoder.log

Fields:

Unique-Key, Date, Time, Publishing-Type, Service Provider, User-ID, FTP-Source,
FTP-Dest, Video-FileName, Num-Profiles, Bit-Rates (kbps), Time-To-Upload, Time-To-
Encode, Completion-Time, Success-Status

AX-001, MM/DD/YYYY, HH:MM:SS, Smoothflow/iPhone, encoding.com, xyga, username@ip:/
in-path, username@ip:/out-path, foo.flv, 2, 500; 1000, HH:MM:SS, HH:MM:SS,
HH:MM:SS, PASS
```

Table 36

```
pub-asset-polling.log

Unique-Key, Date, Time, Num-TxID's, Tx-ID's (semi-colon delimited string)

AX-001, MM/DD/YYYY, HH:MM:SS, 2, 123432;21300
```

Table 37

```
pub-asset-segmenter.log

Fields:

Unique-Key, Date, Time, Time-To-Segment, Num-Objects/Chunks, Origin-Path, Success-
Status

AX-001, MM/DD/YYYY, HH:MM:SS, 00:02:40, 2039, ip:/path, Error (Code:093)
```

Creating On-Demand Assets

You may use another encoding source (rather than encoding.com), or do the encoding yourself. In this scenario, after creating the multi-bit-rate profiles of a particular video, and an Asset Description file, you pre-stage the encoded assets to Media Flow Controller. The set of multi-bit-rate video profiles *along with* the Asset Description file, are, together, "the asset."



CAUTION: You encode the multi-bit-rate video profiles yourself if using this procedure.

These assets are called "On-Demand" because the multi-bit-rate assets (already encoded) and the Asset Description file are pre-staged to Media Flow Controller and the appropriate profiles are served as client requests arrive (on-demand). When bandwidth at a client rises, a higher bit-rate profile chunk is served; when bandwidth falls, a lower bit-rate profile chunk is served.

Tip! Set the **cache-age** to a high value for the SmoothFlow namespace so that content does not get expired.

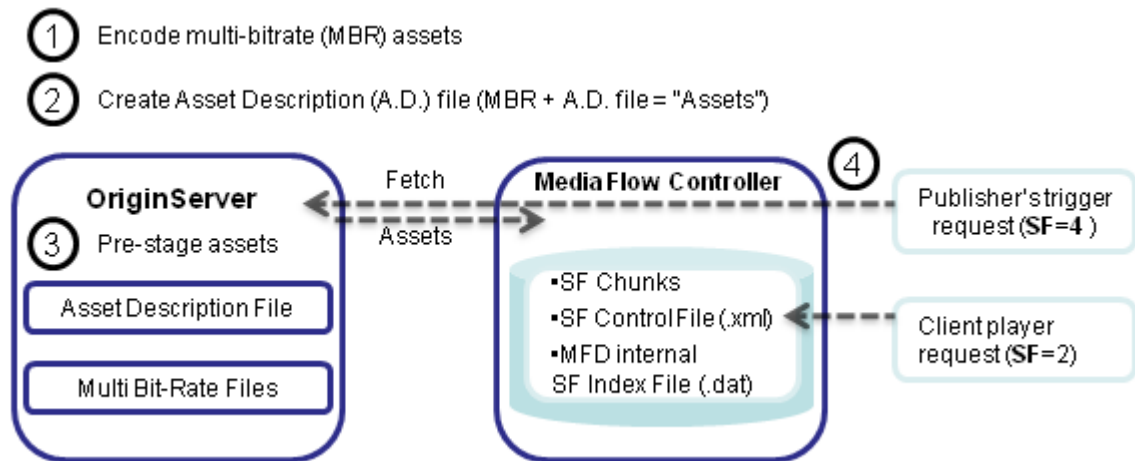


Figure 127 On-Demand Workflow, Typical Steps

Before You Begin Creating On-Demand Assets

You begin by determining how many bit-rate profiles for a video you want to encode; see [“Encoding Requirements” on page 338](#) for more information.

Juniper Networks uses the multi-bit-rate profiles, a metadata file that you create (the **AssetDescription.dat** text file), and commands that you issue at the command line, to create SmoothFlow assets on-demand.

Steps for Creating On-Demand Assets

1. Encode multi-bit-rate profiles for each video using the method of your choice and the naming guidelines given in [Bit-Rate Profiles Naming Conventions for On-Demand Assets](#).
2. For each set of multi-bit-rate profiles, create an asset description file describing the multi-bit-rate profiles including frame rate, keyframe interval, and assured flow rate and threshold. See [Creating the AssetDescription.dat File for On-Demand Assets](#).
3. Pre-stage the assets (the set of multi-bit-rate profiles and AssetDescription.dat file), placing assets for each video on the Media Flow Controller or your origin-server. See [Pre-Staging On-Demand Assets](#).
4. Prepare the assets for on-demand SmoothFlow delivery by [Initiating SmoothFlow Processing for On-Demand Assets](#).

Bit-Rate Profiles Naming Conventions for On-Demand Assets

When naming the bit-rate profiles after encoding, Media Flow Controller SmoothFlow requires this convention:

```
<name>_p<NN>.flv
```

Where **name** is the name of the video, followed by an underscore, the lower-case letter “p” and two digits, **NN**, representing the bit-rate profile number (**01** - **99** with **01** being the lowest bit-rate profile and increasing numbers denoting higher encoding bit-rates). For example, **foo_p01.flv** is presumed to be a lower bit-rate encoding than **foo_p05.flv**. The maximum number of profiles that are supported by the naming convention is 99.



NOTE: SmoothFlow only supports .flv files, but re-containerizes MP4 files to .flv as needed.

Please see [“Encoding Requirements” on page 338](#) for important information.

Creating the AssetDescription.dat File for On-Demand Assets

Along with the bit-rate profiles, you must pre-stage to Media Flow Controller an Asset Description file, as a simple text file (ASCII, with the "dat" extension), in the format described. The Asset Description file does not need to be located with the bit-rate profiles (it tells Media Flow Controller where they are) but must be located in the origin server configured in the namespace you intend to use.

Asset Description file format example, variables are underlined.

```
Version:1.0.4
Profiles:4
Frame Rate:24
KeyFrameInterval:2
Sequence duration:664
Profile_1:250
Profile_2:500
Profile_3:750
Profile_4:1000
URI: http://media.example.com/foo_p01.flv
URI: http://media.example.com/foo_p02.flv
URI: http://media.example.com/foo_p03.flv
URI: http://media.example.com/foo_p04.flv
Assured Flow rate:1
AFR Threshold:600
```

Asset Description file required information:

- Number of profiles (streams) in the asset
- Key Frame Interval in seconds
- Total bit-rate of each profile in the form **Profile_<number> <kbps>**
- URI for each profile
- Assured Flow rate; hardcoded at 1
- AFR Threshold (should be set at a value greater than 1/2 the bit-rate of the highest bit-rate profile in the asset; see [About AFR Threshold](#))

These fields are optional:

- Frame Rate in fps (frames per second)
- Sequence duration in seconds

The **Version** tag is automatically set to the appropriate Release number.

About AFR Threshold

AFR (Assured Flow Rate) is a feature in Media Flow Controller that guarantees that Media Flow Controller delivers at the specified bit-rate and never goes below (or too much above) that rate. See [“Media Flow Controller AssuredFlow” on page 56](#) for overview.

The AFR threshold value can be specified in the Asset Description File (.dat), if using the On-Demand Publishing scheme (or using a Python script if using **encoding.com**). Its unit is kbps. Do not specify this value in the Smoothflow Control File; it automatically picks up the value from the Asset Description File (.dat) when SmoothFlow processing is initiated.

The rationale behind providing the AFR value is empirical. For example, if there are five profiles each at (200, 400, 600, 800 and 1000 kbps). We recommend you set the value for AFR threshold to be greater than half the bit rate of the highest profile. In this case you can set it to any value greater than 500 kbps. This accelerates the switching speed from lower to higher profiles. In this example, the player will switch from 200 to 400 to 600 very quickly and will then slow down in terms of switching to 800 and 1000 kbps. Typically, content providers do not like players to very quickly switch to the highest bit-rate because this increases their bandwidth costs. If you want switching to the highest bit-rate to happen faster, then set the AFR threshold value to be equal to the highest available rate (for example, 1000 kbps).

Pre-Staging On-Demand Assets

The Asset Description file provides Media Flow Controller with the information it needs to SmoothFlow process and serve the encoded videos. Typically, it is placed on your Web server with the media assets; however it can be kept on any defined origin server in the namespace you use. A few options to stage the assets are as follows:

- Place it on your Web server, define that Web server as an origin server in the SmoothFlow namespace, use that namespace's defined uri-prefix in your SmoothFlow processing request.
- FTP it to your Media Flow Controller origin server; if your SmoothFlow processing request is to that Media Flow Controller, it looks for the file locally.

Tip! Typically, assets are pre-staged to an origin server with non-volatile storage (NFS, RAID, or NAS); however, you may pre-stage assets to an edge cache. In that case, be sure to also pre-stage them to the origin server as well, since edge caches are typically configured to delete files after a short time (cache expiry).

Initiating SmoothFlow Processing for On-Demand Assets

After the SmoothFlow assets and the Asset Description file are created and pre-staged to origin, and Media Flow Controller is configured, you send a request to Media Flow Controller to initiate SmoothFlow processing. After the files are fetched, SmoothFlow builds the SmoothFlow Control file (see [About SmoothFlow Control File](#)) that is sent to the client player with asset information.

A processing request with SmoothFlow state **sf=4**, of the form shown, must be sent to initiate SmoothFlow processing. You can do this through a browser, or use wget or curl.

```
http://<name_of_mfc>/sf/<name_of_MediaAssetDescription_file>.dat?sf=4
```

Media Flow Controller internally decomposes this request into the relevant configured namespace and, if the virtual player type assigned to that namespace matches the SmoothFlow player type (Type 4), then, by recognizing the state of **sf = 4**, Media Flow Controller initiates SmoothFlow processing for the asset referenced in the retrieved Asset

Description file. SmoothFlow processing is only done when a SmoothFlow process request from the publisher, in the form given above, is received, or when there is a cache miss in real time. If the pre-staging is to an attached NFS library, then the publisher can see the listing of the files published using their FTP client. If the pre-staging is to Media Flow Controller directly, then they can use the **namespace object list all** command to see the listing.

About SmoothFlow Control File

This XML file, called the SmoothFlow Control file, is created by the Media Flow Controller SmoothFlow processor when SmoothFlow processing is successfully initiated; it provides the information the player needs to request different bit-rates in response to changes in bandwidth. Your client player must know how to read it. You can imbue your client player with this intelligence through the Juniper Networks SmoothFlow SDK or API, available through Juniper Networks Customer Support (not necessary when using the Juniper Networks SmoothFlow Reference player). This file is requested with SmoothFlow state 2 (**sf=2**). The Smoothflow Control File should not be copied from other assets—each one is unique and generated by the publishing process for that particular asset. The schema for this file:

```
<xml>
<sf_version> X.X </sf_version>
<n_profiles> N </n_profiles>
<profile_map>
<profile_name> p1 </profile_name>
<br> B </br> ! <kbps>
</profile_map>
<profile_map>
<profile_name> p2 </profile_name>
<br> B </br> ! <kbps>
</profile_map>
<profile_map>
<profile_name> p3 </profile_name>
<br> B </br> ! <kbps>
</profile_map>
.
.
.
</xml>
```

Field names:

- **sf_version** (optional)—Version of Smoothflow being used in Media Flow Controller
- **n_profiles**—Number of available profiles for a given media asset
- **profile_name**—Identifier string for a profile; this needs to be sent if the client wants to request a specific profile
- **br**—Bit-rate of the profile

Deploying the SmoothFlow Reference Client Player

A binary SmoothFlow client player is provided as a reference and for testing or evaluation purposes. This section details how to install the Juniper Networks Smoothflow Player for testing purposes onto your Web page.



NOTE: Adobe Flash Player versions 9 and above are supported.

The **SmoothFlow_ReferencePlayer.zip** file contains everything you need to get started. Contact Juniper Networks Customer Support (see [“Requesting Technical Support” on page 35](#)) to obtain the zip, and unpack it to your Web server.

You work with the following files retrieved from the zip:

- **fPlayerIn.swf**—This is the Adobe SWF (shockwave flash) file with SmoothFlow playback functionality that must be embedded in your browser page.
- **AC_RunActiveContent.js**—This file is used to dynamically generate JavaScript-based active content embedding tags for the browser to display your Flash movie. This file contains functions that embed your active content based on the parameters it receives from the main.html page.
- **fPlayerInternal.html**—This is the file that you request via your portal server to view the sample SmoothFlow content. Be sure to set **base_uri** and **video_name** tags correctly.
- **crossdomain.xml**—Flash uses this file to control the cross-domain resource-access policy, for both HTTP and socket connections. Use it to control which resources a Flash application can access, when that application did not originate in the domain of the site. The crossdomain.xml file must be hosted and delivered via Media Flow Controller.

Follow these steps to deploy the SmoothFlow Client player:

1. Copy **fPlayerIn.swf**, **AC_RunActiveContent.js**, **fPlayerInternal.html**, and **crossdomain.xml** from the zip to the same directory in the doc root of your Web server.
2. Modify the HTML below (source for **fPlayerInternal.html**). Reset the underlined references as needed. [Table 38](#) gives a brief explanation for the configurable parameters and is followed by an example file ([Example fPlayerInternal.html](#)).
3. Use the URI for the player as follows. For example, if the HTML page in which you have embedded the player is **fPlayerInternal.html**, then any anchor links to that page should be of the form:

```
www.somedomain.com/fPlayerInternal.html
```

Table 38 Required Configurable Nodes and Parameters

| Parameter | Description |
|--------------|---|
| 'base_uri' | The domain hostname/IP address where the video sequences reside (must be configured). |
| 'video_name' | The rest of the URI, which gives the logical path and video name, without any extensions. For example, if the video's full URI is <code>http://www.somedomain.com/sf-test/foo.flv</code> the video_name will be /sf-test/foo The player automatically builds the rest of the URI which will conform to the SmoothFlow Query Parameter API specifications. |
| 'ext' | The file's extension; usually .flv . |

Example fPlayerInternal.html

```

<script language="javascript">
if (AC_FL_RunContent == 0) {
alert("This page requires AC_RunActiveContent.js.");
} else {
AC_FL_RunContent(
'codebase', 'http://download.macromedia.com/pub/shockwave/cabs/flash/
swflash.cab#version=9,0,0,0',
'width', '100%',
'height', '100%',
'src', 'fPlayerIn',
'quality', 'high',
'pluginspage', 'http://www.macromedia.com/go/getflashplayer',
'align', 'middle',
'play', 'true',
'loop', 'true',
'scale', 'default',
'wmode', 'window',
'devicefont', 'false',
'id', 'fPlayerIn',
'xml', 'smoothflow.xml',
'FlashVars', 'base_uri=162.19.162.39&video_name=sf-test/sf-comp-04&ext=flv',
'bgcolor', '#000000',
'name', 'fPlayerIn',
'menu', 'true',
'allowFullScreen', 'true',
'allowScriptAccess','sameDomain',
'movie', 'fPlayerIn',
'salign', 'TC'
); //end AC code
}
</script>
<noscript>

```

Media Flow Controller URI-Prefix Video Name

Deployment Checklist

First, you must determine how many bit-rate profiles you want for each video. See [“Evaluating Your Needs” on page 338](#) and [“Encoding Requirements” on page 338](#).

For [“Creating Assets Using an SaaS” on page 343](#):

1. Create an account with [encoding.com](#)
2. Post one single-bit-rate video file to your Web server.
3. Create the **setup.xml** file describing the Website of your chosen encoding service and your login credentials for the service, where you want the multi-bit-rate profiles output to, and credentials for that (for example, FTP site and credentials), and how often you want status polling done. See [“Using Scripts to Create Assets Using an SaaS” on page 345](#).
4. For each asset, create the **asset.xml** file describing the single-bit-rate video file, how many multi-bit-rate profiles you want created for it and at what rates, including frames-per-second, container format, and keyframe interval. See [“Using Scripts to Create Assets Using an SaaS” on page 345](#).

5. Use the **SFAssetGenerator.py** script to [“Initiating Encoding Using an SaaS” on page 345](#) to create multi-bit-rate (MBR) profiles for each single-bit-rate video file.
6. [“Verifying that Encoding has Completed” on page 349](#) and the assets are transferred to your FTP output account given in the **setup.xml** file.
7. [“Preparing Media Flow Controller for Assets Created Using an SaaS” on page 350](#) by creating the **segment_config.xml** file describing the multi-bit-rate assets for Media Flow Controller including the IP address of the Media Flow Controller, where the assets are located, access credentials for obtaining the assets, and a temporary directory in the system for SmoothFlow processing.
8. Use the **SFSegment.py** script referencing the **segment_config.xml** file to initiate SmoothFlow processing and publishing to Media Flow Controller, and render the assets ready for delivery.
9. Check [“Logs for Assets Created Using an SaaS” on page 351](#).

For [“Creating On-Demand Assets” on page 352](#):

1. Encode multi-bit-rate profiles for each video using the method of your choice; follow [“Bit-Rate Profiles Naming Conventions for On-Demand Assets” on page 353](#).
2. For each asset, [“Creating the AssetDescription.dat File for On-Demand Assets” on page 354](#) by creating an **AssetDescription.dat** (text) file describing the multi-bit-rate profiles including frame rate, keyframe interval, and assured flow rate and threshold.
3. [“Pre-Staging On-Demand Assets” on page 355](#) by placing the assets on the Media Flow Controller or your origin-server.
4. [“Initiating SmoothFlow Processing for On-Demand Assets” on page 355](#) to prepare the assets for on-demand SmoothFlow delivery.

Finally, see [“Deploying the SmoothFlow Reference Client Player” on page 356](#) to complete SmoothFlow deployment.

PART 2

Media Flow Controller Command Reference

CHAPTER 14

Media Flow Controller CLI Command Reference

Certain commands only appear if you are in one of the three command modes: **Standard** (no configuration permissions), **Enable** (few configuration permissions), and **Configure** (full configuration permissions); see [“CLI Options” on page 84](#) for details. Commands that do not require being in **Configure** mode are **EXEC** commands.

aaa (authentication, authorization, accounting) Set authentication and authorization.

accesslog Configure access log.

analytics Configure cache analytics options.

application Configure Flash Media Server integration.

arp (Address Resolution Protocol). Set ARP servers.

banner Manage Web banners.

bond Configure bonded interfaces.

boot Configure system booting.

bridge Configure bridge groups for Spanning Tree Protocol (STP).

cachelog Configure cache log.

clear EXEC Clear the arp cache.

cli CLI shell options.

clock Set the system date and time.

cmc Not Supported.

collect counters Special debugging tool for use with Customer Support.

configuration Manipulate configuration files.

configure Go to **Configuration** mode for additional commands; disallowed for **unpriv** users.

debug Generate a "dump" of the system debugging utility.

delivery Set delivery protocol options, including enabling delivery trace.

email Configure e-mail and event notification via e-mail.

enable Go to **Enable** mode for additional commands; disallowed for **unpriv** users.

errorlog Configure error log.

exit Leave **Configure** mode, or close the CLI window if in **Standard** mode.

file Manipulate stats and tcpdump reports.

fmsaccesslog Lists all FMS server command executions.

fmsedgelog Lists transactions to the FMS edge server.

fuselog Records RTMP transaction details.

help View the interactive help system.

hostname Set the system hostname.

image Manage software images.

interface Configure network interfaces.

ip Configure IP addresses.

ldap Set up Light-weight Directory Access Protocol (LDAP).

license Activate features using license keys.

logging Configure event logging.

management Configure management interface (eth0).

media-cache EXEC Configure disk-cache and file-cache settings.

mfdlog Configure accesslog and errorlog ports and interfaces.

namespace Configure namespaces. Includes **EXEC** command **namespace object list**.

network Make network layer configurations.

no Negate or clear certain configuration options.

ntp and **ntpdate** Configure Network Time Protocol (NTP) servers and system clock.

ping EXEC Send ICMP echo requests to a specified host.

radius-server Configure RADIUS server settings.

ram-cache RAM cache options.

reload Reboot or shut down the system.

reset Reset the system to its factory state.

scheduler Configure the number of real-time scheduler threads.

server-map Specify a file and protocol for resolving incoming cache-miss requests.

service Restart certain services after an IP address or delivery protocol port change.

show List system configuration or statistics; applies to most commands; for example, **show files** lists available files or lists their content, if the file is specified. Includes special subcommands. Many are **EXEC** commands.

slogin EXEC Log in to another system securely using SSH.

snmp-server Configure SNMP server options.

ssh Configure secure shell (SSH) settings.

stats Configure statistics and alarms.

streamlog Configure streaming log.

tacacs-server Configure TACACS+ server settings.

tcpdump List packets on a network.

tech-support EXEC Collect system information.

telnet EXEC Log into another system using telnet.

telnet-server Enable/disable the TELNET server.

terminal EXEC Set terminal options. See [“CLI Options” on page 84](#).

tracelog Configure trace log.

traceroute EXEC Trace the route packets take to a destination.

upload Upload the accesslog, errorlog, or a **namespace** object.

username Configure user accounts and set capabilities.

virtual-player Configure Media Flow Controller player management functions.

web Configure the Web interface, also known as the Management Console.

write Save the running configuration to persistent storage.

aaa

Configure authentication, authorization and accounting (AAA) settings; AAA accounting options are not supported at this time. RADIUS or TACACS+ authentication must be configured before these options can be specified with this command.

aaa (authentication)

Configure authentication settings.

```
aaa authentication login default <method> [<method>] [<method>] [<method>]
```

Notes:

- **authentication login default <method>**—Set the list of acceptable authentication methods for system logins. Choose from **ldap**, **local**, **radius**, and **tacacs+**. The order in which the methods are specified is the order in which they are attempted. Default is **local**. Use **no aaa authentication login** to reset default.

aaa (authorization)

Configure authorization settings.

```
aaa authorization map
  default-user <user>
  order {remote-only | remote-first | local-only}
```

Notes:

- **default-user <username>**—Specify what local account a non-local user authenticated via RADIUS or TACACS+ is logged on as; you must enter a **username** that exists locally and is enabled. This mapping is used depending on the setting of **authorization map order**. Use **no** to reset default (**admin**).
- **order**— Determine how the remote user mapping behaves when authenticating users via RADIUS or TACACS+. Again, if the authenticated user name is valid locally, no mapping is performed. Use **no aaa authorization map order** to reset default (**remote-first**).

Arguments:

- **remote-only** — Only try to map a remote authenticated user if the authentication server sends a local-user mapping attribute; otherwise, no further mapping is tried.
- **remote-first** (default) — If a local-user mapping attribute is returned and is a valid local user name, map the authenticated user to the local user specified in the attribute.

Otherwise, if the attribute is not present or not valid locally, use the user specified by the **default-user** command.

- **local-only** — All remote users are mapped to the user specified by the **aaa authorization map default-user <user name>** command. Any vendor attributes received by an authentication server are ignored.

```
show aaa
```

List current authentication and authorization settings.

accesslog

Use these commands to set access log options. The accesslog records all command executions. See [“Configuring Media Flow Controller Service Logs \(CLI\)” on page 188](#) for task details including information on log rotation. See [“Reading the Service Log \(accesslog\)” on page 190](#) for usage information, status codes, and sub-codes.

```
accesslog
  copy <SCP>
  filename {access.log | <filename>}
  format {<field1 field2 ...> | clf | display | ncsa | ncsa-ext}
  max-fileid <integer>
  on-the-hour {disable | enable}
  rotate {filesize <integer> | time-interval <hours>}
  syslog replicate {enable | disable}
```

Type **accesslog** to enter accesslog configuration mode; only **accesslog** commands are available. Type **exit** to leave accesslog configuration mode.

Notes:

- **copy**—Set auto-upload (when the set **rotate filesize** criteria is reached) for access log using SCP (secure channel protocol), to the server specified using **hostname**. If **username** and **password** are provided, Media Flow Controller uses that for authentication of the SCP session. Use **no accesslog copy** to disallow auto-upload. See [“Terminology” on page 31](#) for the **scp** URL format and requirements.
- **filename**—Configure the name of the file where the access log is stored. Default is **accesslog.<num>.yyyymmdd_hour:min:sec** (numbered sequentially).
- **format**—Specify a format for the access log; see [Table 18, “Accesslog Format Options,” on page 184](#), for **format field** options. Default is **ncsa-ext**.
 - **<field1 field2 ...>**—Choose available field options, described in [Table 18, “Accesslog Format Options,” on page 184](#).
 - **clf**—Common Log Format, by default: **%h %V %u %t %r %s %b**.
 - **display**—Either **enable** (default) or **disable** the display of the format in the log.
 - **ncsa**—National Center for Supercomputing Applications, default **ncsa-ext** format without the “**{User-Agent}**”%y field.
 - **ncsa-ext**—Default; **%c %h %V %u %t “%r” %s %b %N “%{Cache-Control}i” “%{Pragma}i” “%{Cache-Control}o” “%{Pragma}o” “%{Vary}o” %y**.
- **max-fileid**—Set the number of log files to retain. Default is **10**.

- **on-the-hour**—Set hourly log rotation with **enable**. Default is **no** (disabled). This setting takes precedence over a **rotate time-interval <minutes>** setting.
- **rotate**—Media Flow Controller allows access log rotation based on file size or time. If **copy** auto-upload has been configured, the log is uploaded to the specified **copy** URL; if **copy** auto-upload has not been configured, the log is replaced with a new log.
 - **filesize**—Default. Set rotation based on file size. Media Flow Controller creates "access.log.1," "access.log.2," and so on all the way to "access.log.10," after which it wraps around. By default, **rotate filesize** is **100**. We highly recommend not increasing the size; huge file transfers take a lot of time, and if there is a system reset, large volumes of data are at risk.
 - **time-interval**—Set rotation based on time. Specify a time in minutes after which the access log is rotated. If you set **rotate time-interval 5** and also **on-the-hour enable**, the **on-the-hour** setting takes precedence and accesslog rotates on the hour. Default is **0** (zero), which means no rotation based on time.
- **syslog replicate**—Specify whether or not the access log messages are seen as part of syslog also. Default is **no** (disabled), access log is not seen as part of syslog.

```
show accesslog [continuous | last]
```

List access log settings; or use **last** to see the last few lines of the log and **continuous** to view the accesslog as it is written.



NOTE: Media Flow Controller accesslog is enabled by default and cannot be disabled.

NOTE: View the Media Flow Controller accesslog through the Web interface, **Logs > Service Log** page.

analytics

Configure cache analytics options. See [“Caching and Origin Clustering” on page 45](#) for details.

```
analytics
```

```
  cache-ingest size-threshold <bytes>
  cache-promotion [disable | enable | hotness-threshold <number>]
  packing-policy hybrid
```

Notes:

- **cache-ingest size-threshold <bytes>**—Set the maximum size of an object that can be optionally ingested into the fastest cache tier in the disk cache. Objects smaller than, or equal to, the configured size are automatically written to the fastest cache tier. Default is **0** (zero), no objects are directly promoted to the fastest tier. Maximum allowed value is **4294967295** (4GB).
- **cache-promotion**—This function moves “hot” objects (most requested) to a higher cache tier and “cold” objects (least requested) to a lower cache tier. Options:
 - **enable** and **disable**—Either **enable** or **disable** cache promotion analytics; default is enabled; if **disable** is used, no cache promotion occurs.

- **hotness-threshold**—Set a threshold for "hotness" value after which an object is candidate for promotion to a higher tier in disk cache. Default is **3**, an object requested three times becomes a candidate for cache promotion.
- **packing-policy hybrid**—Validate packing or performance-related issues. Default is **hybrid**.

```
show analytics
```

List current analytics settings.

application

Flash Media Server (FMS) configuration on Media Flow Controller. See [“Installing and Using FMS in Media Flow Controller \(CLI\)” on page 110](#) for task details.

```
application fms
  download <URL>
  install <filename>
  shell
```

Notes:

- **download <URL>**—Download the FMS from the specified URL. Available download options are SCP, HTTP, and FTP.
- **install <filename>**—Installs the downloaded FMS.
- **shell**—Open a shell window and configure FMS for Media Flow Controller; see [“Installing and Using FMS in Media Flow Controller \(CLI\)” on page 110](#).

arp

Manage the Address Resolution Protocol (ARP) cache.

```
arp <IP_address> <MAC_address>
```

Add or delete (with **no**) static entries to the ARP cache.

```
show arp [static]
```

List contents of ARP cache. This should contain all of the statically-configured ARP entries, as well as any that the system has picked up dynamically. Use the subcommand **static** to list only statically-configured ARP entries.

banner

At various login points, some legal and welcome text can be displayed. See [“Configuring Media Flow Controller Clock and Banners \(CLI\)” on page 92](#) for task details.

```
banner
  login <message_string>
  motd <message_string>
```

Notes:

- **login**—Set system Login Banner. Use **no banner login** to delete the message.
- **motd**—Set system Message of the Day banner. Use **no banner motd** to delete the message.

show banner

List contents of currently configured banners.

bond

Configure bonding interfaces to create a port-channel, or aggregated link, for load distribution across links and for increased link availability. Use the **interface <interface_name> bond** command to add or delete (with **no**) interfaces from the bonding interfaces; see [interface](#). See [“Creating and Configuring Link Bonding and Static Routes \(CLI\)” on page 93](#) and [“Configuring Interfaces, Hostname, Domain List, DNS, and Default Gateway \(CLI\)” on page 88](#) for task details.



NOTE: Individual links under a bond of 1-Gigabyte interfaces cannot be configured for different speeds (10/100/1000). Only one bonded interface is allowed; if a second bonded interface is created it will not work well. Up to four interfaces can be bonded. Mixed mode bonding (1-Gigabyte and 10-Gigabyte interfaces bonded in a single bond) is not supported.

```
bond <bonding_interface>
  down-delay-time <milliseconds>
  link-mon-time <milliseconds>
  mode <mode_name>
  up-delay-time <milliseconds>
```

Create the named bonding interface. Use **no bond <bonding_interface>** to delete.

Notes:

- **down-delay-time**—Wait this long before disabling a slave after a link failure is detected.
- **link-mon-time**—Monitor links with this frequency.
- **mode**—Set the bonding policy:
 - **balance-rr**—“Round-robin” mode. Sends TCP/IP packets belonging to the same session across multiple links. Out-of-order TCP packets coming through different links are retransmitted; supports load balancing and failover.
 - **backup**— Not supported in Release 2.0.7.
 - **balance-xor**— Not supported in Release 2.0.7.
 - **balance-xor-layer3+4**— Traffic to a particular network peer goes across multiple links, although packets belonging to a single connection or session do not span multiple links; supports load balancing and failover. Link selection based on TCP port and IP address.
 - **broadcast**— Not supported in Release 2.0.7.
 - **link-agg**— Not supported in Release 2.0.7.
 - **link-agg-layer3+4**—Link Aggregation Control Protocol (LACP). Allows the automatic negotiation of port bundling to form a single logical channel between LACP-enabled links; supports load balancing and failover.

- `balance-tlb`— Not supported in Release 2.0.7.
- `balance-alb`— Not supported in Release 2.0.7.
- `up-delay-time`—Wait this long before enabling a slave after detecting a link recovery.

`show bonds [<bonding_interface>]`

List configuration information about all or the specified bonding interface.



NOTE: When any change is made to the Media Flow Controller delivery mechanism, including configuring bonded interfaces, you must restart the delivery service (**service restart mod-delivery**).

boot

Configure system booting parameters.

`boot`

```
bootmgr password [0 <cleartext_password> | 7 <encrypted_password> |
  <cleartext_password>]
next fallback-reboot enable
system {location <location_ID> | next}
```

Notes:

- `bootmgr password`—Set the system boot manager password.
 - `0 <cleartext_password>`—Allows the password to be specified in cleartext.
 - `7 <encrypted_password>`—Allows the password to be provided in the same encrypted form in which it would be stored in the system password file. Useful for **show configuration**, since the cleartext password cannot be recovered after it is set.
 - `<cleartext_password>`—Enter a cleartext password; if none is specified, the user is prompted for the password, with entries obscured, requiring the same string to be entered twice for confirmation.
- `next fallback-reboot enable`—Allow or disallow (with **no**) enabling fallback reboot if the configuration file cannot be applied after an upgrade or downgrade is attempted.
- `system`—Specify which **location** the system should boot from by default; use **1** or **2** for location ID. Use **next** to set the boot location to be the next one after the one currently booted from. This does not mean the next one after the one Media Flow Controller is currently set to boot from; thus the command is idempotent, and does not cycle through all of the available locations.

`show bootvar`

Similar to **show images** in that it lists what images are on the two locations, and which are the active and default location; but not all of the **show images** data is displayed.

bridge

Create and configure bridge groups of interfaces. See [interface](#) to add or remove interfaces from a configured bridge group. See also [Spanning Tree Protocol on Wikipedia](#) for details.

```
bridge <bridge_interface>
  ageing-time
  enable
  forward-time
  hello-time
  max-age
  priority
  spanning-tree
```

Notes:

- **ageing-time <seconds>**—Set the aging time of a dynamic (learned) entry in a bridge group's forwarding table. This is the length of time, in seconds, that an entry can remain in the forwarding table. When an entry reaches its aging time, it "ages out" of the forwarding table. The **no** version restores the default value, **300** seconds.
- **enable**—Enable this bridge interface; use **no** to disable.
- **forward-time <seconds>**—Set the forward delay interval for the specified bridge. Default is **30** seconds.
- **hello-time**—Set the interval between "hello" bridge protocol data units (BPDUs). Default is **2** seconds.
- **max-age**—Set the interval that the bridge waits to hear BPDUs from the spanning tree root. If the bridge does not hear BPDUs from the spanning tree root within this specified interval, it assumes that the network has changed and recomputes the spanning-tree topology. Default is **15** seconds.
- **priority**—Set the spanning tree priority for the bridge. The spanning tree protocol (STP) uses the configured bridge **priority** to select the spanning tree root. The lower the priority, the more likely it is that the bridge will become the spanning tree root. The radio and Ethernet interfaces and the native VLAN on the bridge are assigned to bridge group 1 by default. When you enable STP and assign a priority on bridge group 1, STP is enabled on the radio and Ethernet interfaces and on the primary VLAN, and those interfaces adopt the priority assigned to bridge group 1. You can create bridge groups for sub-interfaces and assign different STP settings to those bridge groups.
- **spanning-tree enable**—Enable or disable SPT on this bridge; STP is enabled for all interfaces assigned to the specified bridge group.

```
show bridges [<bridge_interface>]
```

List configuration information about all or the specified bridging interface.

cachelog

Configure cache log options. See "[Reading the Cache Log \(cachelog\)](#)" on page 190 for usage information. See "[Configuring Media Flow Controller Service Logs \(CLI\)](#)" on page 188 for task details, including information on log rotation.

```

cachelog
  copy <SCP>
  filename <name>
  on-the-hour {disable | enable}
  rotate {filesize <integer> | time-interval <hours>}
  syslog replicate {disable | enable}

```

Notes:

- **copy**—Auto-upload (when the set **rotate** criteria is reached) the cachelog using the (secure channel protocol (SCP), to the server specified using **hostname**. If **username** and **password** are provided, Media Flow Controller uses that for authentication of the SCP session. The **no** variant disallows auto-upload. See [“Terminology” on page 31](#) for the **scp** URL format); you must have an SCP server installed in order to send files to your machine.
- **filename**—Configure the name of the file where the cache log is stored. Default is **cachelog.<num>.yyyymmdd_hour:min:sec** (numbered sequentially).
- **on-the-hour**—Set hourly log rotation. Default is **no** (disabled).
- **rotate**—Media Flow Controller allows cache log rotation based on file size or time.
 - **filesize**—Set rotation based on file size. Media Flow Controller creates "cache.log.1," "cache.log.2," and so on up to "cache.log.10," after which it wraps around. By default, **rotate filesize** is **100** MB. We highly recommend not increasing the size; huge file transfers take a lot of time, and if there is a system reset, large volumes of data are at risk.
 - **time-interval**—Set rotation based on time. Specify a time in hours after which the cache log is rotated.
- **syslog replicate**—Specify whether (**enable**) or not cache log messages are seen as part of syslog; default is **no** (disabled), cache log is not seen as part of syslog.

```
show cachelog [continuous | last]
```

List log settings; or use **last** to see the last few lines of the log and **continuous** to view the log as it is written.

clear

```
clear arp-cache
```

EXEC command. Clear dynamic entries from the arp-cache.

cli

Configure CLI shell options.

```

cli
  clear-history
  default
    auto-logout <length_in_minutes>
    paging enable
    prefix-modes enable

```

```

progress
prompt
    confirm-reload
    confirm-reset
    confirm-unsaved
    empty-password
show
session
    auto-logout <length_in_minutes>
    paging enable
    prefix-modes enable
    terminal
        length
        resize
        type
        width
    x-display full

```

Notes:

- `clear-history`—**EXEC** command. Clears the command history of the current user.
- `default`—Configure default CLI options for all future sessions.
 - `auto-logout`—Control the length of user inactivity (in minutes) required before the CLI logs a user out. The **no** variant disables the automatic logout feature.
 - `paging enable`—Enable or disable (with **no**) paging of CLI output. If paging is enabled, all command output, as well as all help text printed when the question mark (?) key is pressed, is displayed one screen at a time, using the same pager as the **show log** command. If the text to be displayed fits on a single screen, it is displayed normally and the pager is not used. The abbreviated list of commands is displayed when **<tab>** is hit twice is not paged, even in the unlikely event that it does not fit on the screen. Additionally, if the CLI does not have a terminal (for example, it is being driven by a script), paging is disabled automatically regardless of the default setting, and cannot be re-enabled for this session. However, even in this case, the default setting can still be changed.
 - `prefix-modes enable`—Enable/disable the use of prefix modes in the CLI. If prefix modes are disabled, the commands that were used to enter prefix modes may or may not remain valid standalone commands, depending on the command. Changing this option's default affects this session as well as all future ones, but does not affect other sessions already in progress.
 - `progress enable`—Enable/disable progress updates for long operations.
 - `prompt`—Configure when the CLI should prompt you for input.
 - `confirm-reload`—Enable or disable (with **no**) confirmations of rebooting or halting the system using the **reload** command. This confirmation is in addition to any separate confirmations that may be displayed for unsaved changes.
 - `confirm-reset`—Enable or disable (with **no**) confirmations of resetting the entire system to its factory default state using the **reset factory** command.
 - `confirm-unsaved`—Enable or disable (with **no**) confirmations of cases where you might accidentally lose unsaved changes. Currently, this is just for the **reload [halt]** command; other cases where you might lose configuration are some of the

configuration commands, which have no confirmations since they are explicitly for configuration.

- **empty-password**—Enable or disable (with **no**) prompting for a password in certain cases where a password was permitted but the user did not specify one. Mainly, this applies to pseudo-URLs of the form **scp://username:password@hostname/path/filename** where the **:password** part was omitted. If the prompt is enabled, the CLI asks for a password to be entered. If the prompt is disabled, the CLI assumes there is no password. If you only eliminate the password itself but leave the colon (:), this is treated as an explicit declaration that there is no password, and there is no prompt regardless of this setting.
- **show config-hidden enable**—Enable or disable (with **no**) viewing hidden commands with **show config** commands.
- **session**—**EXEC** commands. Configure CLI options for this session only.
 - **auto-logout**—Control the length of user inactivity (in minutes) required before the CLI automatically logs a user out. The **no** variants of this command disable the automatic logout feature.
 - **paging enable**—Enable or disable (with **no**) paging of CLI output. See **default paging enable** command description for details.
 - **prefix-modes enable**—Enable or disable (with **no**) the use of prefix modes in the CLI. If prefix modes are disabled, the commands that were used to enter prefix modes may or may not remain valid standalone commands, depending on the command. Changing this option's default affects this session as well as all future ones, but does not affect other sessions already in progress.
 - **progress enable**—Enable/disable progress updates for long operations.
 - **x-display full <display>**—Set the display to use for X Windows applications.
 - **terminal**—Set terminal parameters.
 - **length**—Override the auto-detected size of the terminal. This is useful mostly when the size could not be auto-detected and the CLI is using the default **80x24**. These settings are persistent only for the current CLI session. They are also lost if the terminal is resized and the CLI is able to auto-detect its new size.
 - **width**—Override the auto-detected size of the terminal. This is useful mostly when the size could not be auto-detected and the CLI is using the default **80x24**. These settings are persistent only for the current CLI session. They are also lost if the terminal is resized and the CLI is able to auto-detect its new size.
 - **resize**—Resize the CLI terminal settings to match with your real terminal.
 - **type <type>**—Set the type of the terminal. The **no** variants clear the terminal setting, which causes the session to be treated as a 'dumb' terminal.

show cli

List CLI settings: the inactivity timeout, whether or not paging is enabled, the terminal size and type. For settings which have configured defaults, both those and the current session settings are displayed.

clock

Set the system clock and time zone.

```
clock
  set <hh>:<mm>:<ss> [<yyyy>/<mm>/<dd>]
  timezone <zone> [<zone_word>] [<zone_word>] ...
```

Notes:

- **set**—Set the system clock. The time must be specified. The date is optional; if not specified, the date is left the same.
- **timezone**—Set the system time zone. Default is **UTC**. The **no** variant resets to default. The **timezone** may be specified in one of three ways:
 - A nearby city whose time zone rules follow. The system has a large list of cities that can be displayed by the **help** and completion system. They are organized hierarchically because there are too many of them to display in a flat list. A given city may be required to be specified in two, three, or four words, depending on the city. The possible forms this could take include:


```
<continent> <city>
<continent> <country> <city>
<continent> <region> <country> <city>
<ocean> <island>
```
 - An offset from GMT. This is in the form:


```
GMT-offset GMT (default)
GMT-offset GMT+<1-12>
GMT-offset GMT-<1-14>
```
 - UTC. This is almost identical to GMT.

```
show clock
```

Current system time, date, and time zone.

cmc

Not Supported.

```
cmc
  auth {ssh | ssh-dsa2 | ssh-rsa2}
  client {bw-limit | confirm-config | connection | enable | server}
  rendezvous [client] [service-name]
```

collect counters

This command is for debugging use and requires the assistance of Customer Support.

```
collect counters detail <name> frequency <seconds> duration <seconds>
```

The **collect counters detail <name>** value will be supplied by Customer Support to help pinpoint an issue; see [“Requesting Technical Support” on page 29](#). The allowable names are available only through Customer Support except the **all** counter detail name, which is - (dash).

The **frequency** value determines how often the collection is run, and the **duration** value determines how long the collection is run.

The **collect counter** command creates a file, **counters-<mfc-name>-<date-time>.txt**, in the directory **/var/opt/tms/sysdumps**.

Example:

```
collect counters detail - frequency 2 duration 2
```

This command will capture all counter values twice every second for a total duration of 2 seconds. You will have four sets of counter data in the output file.

```
collect counters detail http frequency 2 duration 2
```

This command will capture all counter values whose name pattern matches with the keyword **http**, twice every second for a total duration of 2 seconds. You will have four sets of counter data in the output file.

Use **reset counters** to reset the counter values for the counter fields displayed with **show counters**.

```
show counters {http | rtsp}
```

Notes:

- **show counters**—List the **stats counter** details. The **http** option lists HTTP-related counters alone; the **rtsp** option lists RTSP-related counters. Without any option given, lists the common counters data, HTTP-related and RTSP-related data.

configuration

The system can store one or more configuration files on persistent storage with one of the files is designated as **active**: the file that configuration is loaded from on boot, and to which configuration is saved upon a save request. Configuration changes are immediately applied to the running configuration, but are not made persistent until they are explicitly saved using **configuration write**. See [“Saving and Applying Configurations, Resetting Factory Defaults \(CLI\)” on page 118](#) for task details.

configuration

```
copy {initial.bak | initial | <source_filename>} <dest_filename>
delete {<filename> | initial.bak | initial}
fetch {<URL> | <SCP>}
merge {<filename> | initial.bak | initial}
move {<source_filename> | initial.bak | initial} <dest_filename>
new <filename> [factory [keep-basic] [keep-connect]]
revert {factory [keep-basic] [keep-connect] }
switch-to {<filename> | initial.bak | initial}
text
  fetch <URL_or_SCP>
    apply [discard][fail-continue][verbose]
    filename <filename> [apply [fail-continue][verbose]]
  file <filename>
    apply [fail-continue][verbose]
    delete
    rename <new_filename>
```



```

    upload <URL or SCP>
generate
    active {running | saved} [save <filename>] [upload <URL_or_SCP>]
    file {<filename> | initial | initial.bak} [save <filename>]
        [upload <URL_or_SCP>]
upload {active <URL> | initial.bak | initial}
write [to <filename>] [no switch]

```

Notes:

- **copy**—Copy a configuration file. This does not affect the current running configuration. The active configuration file may not be deleted or renamed, nor may it be the target of a move or copy. It may be the source of a copy, in which case the original remains active.
- **delete**—Delete a configuration file. This does not affect the current running configuration. The active configuration file may not be deleted.
- **fetch**—Download (fetch) a configuration file. A file may not be downloaded over the active configuration file. If no **filename** is specified for a **configuration fetch**, it is given the same name as it had on the server. See [“Terminology” on page 31](#) for the **scp** URL format and requirements.
- **merge**—Merge the shared configuration from one (non-active) configuration file into the running configuration. No configuration files are modified during this process.
- **move**—Move a configuration file. This do not affect the current running configuration. The active configuration file may not be the target of a move or copy.
- **new**—Create a new configuration file under the specified **filename**. The arguments specify what configuration, if any, to carry forward from the current running configuration. The **factory** argument creates the new file with only factory defaults. The optional **keep** arguments preserve portions of the running configuration:
 - **keep-basic**—Preserves licenses, SSH host keys, and CMC rendezvous settings.
 - **keep-connect**—Preserves anything necessary to maintain network connectivity to the system: interfaces, routes, and ARP (does not preserve hostname).

Either, both, or neither may be selected after **factory**; if neither are specified, the default is **keep-basic**. Thus **configuration new <filename>** is the same as **configuration new <filename> factory keep-basic**.
- **revert factory**—Revert both running and saved configurations to factory defaults. Reverting to the factory defaults wipes the IP address of your management interface; you must use the serial console to re-configure or switch to a saved configuration.
 - **keep-basic**—Preserves licenses, and SSH host keys.
 - **keep-connect**—Preserves anything necessary to maintain network connectivity to the system: interfaces, routes, and ARP.
- **text**—Manage text configuration files; see [configuration text](#) for details.
- **switch-to**—Load configuration from the specified file and change that to be the active configuration file. The current running configuration is lost, and not automatically saved to the previous active configuration file.
- **upload**—Upload a configuration file. If **active** is specified for a configuration upload, the currently-active configuration file is uploaded. No configuration file may be named **active**.

- **write**—Write the running configuration to persistent storage. If the **to** keyword is not used, write to the currently active file. If **to** is used, write to persistent storage to the specified file, and change the active file to that one. If **no-switch** is specified after **to**, the active configuration file is not changed to the named file after the save.

If downloading configuration files from another system running the management system, they can be found in the **/config/db** directory. So an example command line to fetch the **initial** configuration database would be:

```
configuration fetch scp://admin:password@hostname/config/db/initial
write [memory] [terminal]
```

These commands perform the same functions as the as **configuration write** commands; included for ease-of-use. Notes:

- **write memory**—Same as **configuration write**.
- **write terminal**—Same as **show running-config**.

```
show configuration
  files [<filename>]
  full
  running [full]
  text files
```

List the CLI commands needed to bring the state of a fresh system up to match the current persistent (saved) state of this system. A short header is included, containing the name and version number of the configuration, in a comment. Arguments:

- **files**—If no **filename** is specified, list configuration files in persistent storage. If **filename** is specified, list the commands to recreate the configuration in that file; only non-default commands are shown.
- **full**—Same as **show configuration** but includes commands that set default values.
- **running**—Same as **show configuration** except that it applies to the currently running configuration, rather than the active saved configuration.
- **text files**—List text-based configuration files.



NOTE: Commands that would set something to its default are not included—so this command on a fresh configuration produces no output, except the header.

NOTE: This does not include changes that have been made but not yet written to persistent storage.

```
show running-config
show running-config [full]
```

The **show running-config** commands perform the same functions as the **show configuration** commands and are included for ease-of-use.

configuration text

Manage text-based configuration files (lists of CLI commands). These files are stored in **/config/text** on the appliance. Not all configuration is included in the **show configuration** output, so a text configuration file generated and re-applied later may not fully recreate the same configuration.

```

configuration text
  fetch <URL_or_SCP>
    apply [discard][fail-continue][verbose]
    filename <filename> [apply [fail-continue][verbose]]
  file <filename>
    apply [fail-continue][verbose]
    delete
    rename <new_filename>
    upload <URL or SCP>
  generate
    active {running | saved} [save <filename>] [upload <URL_or_SCP>]
    file {<filename> | initial | initial.bak} [save <filename>] [upload
      <URL_or_SCP>]

```

Notes:

- **fetch**—Download a text-based configuration file from the specified remote host; see [“Terminology” on page 31](#) for the **scp** URL format and requirements. Options for the fetched file:
 - **filename <filename>**—Name the fetched file; if no **filename** is specified, it is given the same name as it had on the server. All **apply** options, except **discard**, are available for fetched and named text files.
 - **apply**—If you opt to **apply** the fetched configuration text file to the running system, you have additional options:
 - **discard**—After the configuration text file is applied, discard it; cannot follow **fetch <URL_or_SCP> filename <filename> apply**.
 - **fail-continue**—If any of the commands in the text file fail, continue applying the rest of the commands.
 - **verbose**—List all commands being executed and their output instead of just those that get errors.
- **file <filename>**—Manage stored text-based configuration files:
 - **apply**—Execute the commands in the specified configuration text file. The commands execute as the present user, lack of required privilege could cause some to fail. The configuration is not reset before executing the commands, so the resulting configuration, overlaid on top of the running configuration, may be more than what is in the configuration text file.
 - **fail-continue**—If any of the commands in the text file fail, continue applying the rest of the commands.
 - **verbose**—List all commands being executed and their output instead of just those that get errors.
 - **delete**—Delete the specified configuration text file.
 - **rename <filename>**—Rename the specified configuration text file.
 - **upload**—Upload the specified configuration text file to the specified remote host.
- **generate**—Generate a new configuration text file from this system, based on either:
 - **active running**—The current active, running configuration (as in `show configuration running`).
 - **active saved**—The current active saved configuration (as in `show configuration`)
 - **file**—An inactive, saved, configuration file (as in `show configuration files`).

After the text file is generated, you have these options:

- **save**—Save the newly generated text file to the specified filename.
- **upload**—Upload the newly generated text file to the specified location.

configure

Enter configuration mode from **Enable** mode.

configure terminal

From **enable** enter **configure terminal** to go to **Configuration** mode. Use **exit** to go from **Configuration** mode to **Enable** mode; **disable** to go from **Enable** mode to **Standard** mode.

debug

Generate debugging information for Media Flow Controller functions.

debug generate dump

Generate a debugging dump (sysdump). The dump can then be manipulated using the **file debug-dump** family of commands; see [file](#) for details.



CAUTION: The system gives high priority to debugging output. For this reason, debugging commands should be turned on only for troubleshooting specific problems or during troubleshooting sessions with technical support personnel. Excessive debugging output can render the system inoperable.

- **show files debug-dump [<filename>]**—List debug dump files or, if filename is specified, list the contents of a particular debug dump file.
- **file debug-dump delete <filename>**—Delete the specified debug dump file.
- **file debug-dump upload <filename> {<URL> | <SCP>}**—Upload the specified debug dump file to the specified URL. Only FTP and TFTP URLs, as well as SCP pseudo-URLs are supported for the destination. See [“Terminology” on page 31](#) for the **scp** URL format and requirements. The uploaded file is a gnu-zipped tar file (.tgz) and can be unzipped with this command on Linux: **gunzip -c <filename>.tgz | tar tf -** or with WinZip on a Windows system.
- **file debug-dump email <filename>**—Send the specified debug dump in e-mail to the list of configured recipients for informational (**info**) events, regardless of whether they requested to receive **detailed** notifications. See [email](#) for setting notification recipients.

delivery

Set delivery options, including listen interfaces, for delivering content to the player (end-user or consumer). If not specified, default actions take place in the delivery path. This command sets global attributes; use [namespace](#) to set these attributes on a namespace basis. See [“Configuring Media Flow Controller Delivery Protocols \(CLI\)” on page 102](#) for task details. See

[“Media Flow Controller Delivery Methods” on page 42](#) and [“Caching and Origin Clustering” on page 45](#) for background information.

```
delivery protocol {http | rtsp}
  allow-req {all | method <method1> [<method2>] [<method3>]... [<method16>]}
  conn-pool
    max-arp-entry <number>
    origin
      disable
      enable
      max-conn <number>
      timeout <seconds>
  connection [close use-reset | persistence num-requests <32 - 100>]
  file-type <suffix> content-type <type>
  interface {all | <interface> <interface> <interface> ...}
  listen port {80 | <port> <port> <port> ...}
  req-length maximum <bytes>
  revalidate-get
  trace enable
  transparent <interface> <interface> <interface>...
```

Set content delivery options; choose either **http** (default port is 80), or **rtsp** (default port is 554). Two **delivery protocol** (one for **http** and one for **rtsp**) configurations are allowed. Not all options are available for **rtsp**. Use **no delivery protocol <protocol> <option>** to reset to default.

Notes:

- **allow-req {all | method <method>...}**—(**http** only) Set request method options:
 - **all**—Default. Accept any request method.
 - **method <method>...**—Media Flow Controller supports known HTTP methods (GET, POST, TRACE, CONNECT, OPTIONS, DELETE, and PUT) always. Other HTTP access methods, including custom methods, are supported via this option; up to 16 may be added.
- **conn-pool**—(**http** only). Set connection pooling options (allows many HTTP requests to multiplex over a single TCP connection to the origin):
 - **max-arp-entry**—Set the maximum number of ARP entries in the ARP cache table.
 - **origin**—Set connection pooling to origin options:
 - **disable** and **enable**—Either **enable** (default) or **disable** connection pooling to the origin server.
 - **max-conn**—Optionally, set a maximum number of connections that can be opened to the origin server concurrently. Default is **4096**; maximum allowed is **128000**.
 - **timeout**—Optionally, set a **timeout** for a single connection in the connection pool; maximum allowed is **86,400** seconds (24 hours), default is **90** seconds. This is separate from **network connection idle timeout**.
- **connection**—(**http** only) Set connection options:
 - **close use-reset**—While closing the connection, issue a reset (RST) against the final (FIN).

- **persistence num-requests**—Set the persistent number of HTTP requests allowed per connection. Default is **32**.
- **file-type to content-type match**—(**http** only) Associate certain file-types (for example, HTML, FLV, MP4, or MOV) with certain content-types (for example, text/html, video/x-flv, video/mp4, or video/quicktime) to be set in the HTTP header. When HTTP is used for access to origin, this command is needed only if a Content-Type header is not returned by the origin server; this has no effect if the origin server returns a Content-Type header. Repeat the command as needed to continue associating file types with content types. Use **no** to delete a specified entry.
- **interface**—Specify the set of interfaces on which the media delivery protocol listens for incoming requests. If not specified, Media Flow Controller listens on ALL interfaces (default); if specified, Media Flow Controller listens ONLY on those specified. You can specify a list of space-separated interfaces such as **eth2 eth3 eth4 eth5** or **eth10 eth11 eth12 eth13 eth20 eth21 eth22**, and so on. Up to 10 can be specified. Use **no** and specify the configured interfaces to reset the default (**all**), or remove a specified interface.
- **listen port**—Set the TCP port used to listen for requests; multiple entries (up to 64) are allowed. Default port for HTTP is **80**; for RTSP is **554**. Use **no** to reset the default.
- **req-length maximum**—(**http** only) Set the maximum parse size for incoming requests (request line + headers) in bytes; requests larger than this size are rejected. Default is **8192** bytes; maximum allowed value is **32768**.
- **revalidate-get**—(**http** only) Use the GET method for revalidation requests instead of the HEAD method (the default). HEAD revalidation requests are more efficient than GET revalidation requests; however, some content websites do not support HEAD requests.
- **trace**—(**http** only) Enable the Media Flow Controller HTTP delivery trace; see ["Testing a Specific Transaction" on page 223](#) for details on using the trace utility. Use **no** to disable.
- **transparent**—(**http** only) Set interfaces for transparent proxy; required configuration for transparent proxy deployments. For more information, see ["Transparent Proxy Deployments" on page 66](#).



NOTE: If you change any **delivery protocol** options, you must run the **service** command for the delivery service: **service restart mod-delivery**.

```
show delivery protocol [http | rtsp]
```

Lists delivery protocol settings.

email

Configure e-mail and event notification via e-mail. See [“Configuring Media Flow Controller Fault Notifications \(CLI\)” on page 208](#) for task details.

```

email
  auth
    enable
    password
    username
  autosupport enable [event <event_name>]
  dead-letter enable [cleanup max-age <duration>]
  domain <hostname or IP_address>
  mailhub <hostname or IP_address>
  malhub-port <port>
  notify
    event <event_name>
    recipient <email_address> [class [failure] [info]] [detail]
  return-addr <username>
  return-host
  send-test

```

Notes:

- **auth**—Set SMTP authorization parameters for e-mail notifications.
 - **enable**—Enable SMTP authentication for Media Flow Controller e-mails; default is disabled. Use **no email auth** to disable again.
 - **password**—Set a password for SMTP authentication of e-mails; if no password is set, the user is prompted for the password. As of Release 2.0.7 the only authentication method supported is "LOGIN", which sends the password in the clear (base64); so users should be aware that this involves some security risk.
 - **username**—Set a username for SMTP authentication of e-mails.
- **autosupport**—Sends e-mails to pre-configured vendor for certain failures.
 - **enable**—Enable or disable (with **no email autosupport**) the sending of e-mail to vendor support when certain failures occur. Default is **enable**.
 - **event**—Specify which events to send autosupport notification e-mails for. See [“email event name Options.”](#) for details.
- **dead-letter**—Manage undeliverable e-mails:
 - **cleanup max-age <duration>**—Set a time limit after which undeliverable e-mails are permanently deleted from the system. The form of **<duration>** is **<number>d<number>h<number>m<number>s**, so **5d4h3m2s** for 5 days, 4 hours, 3 minutes, 2 seconds.
 - **enable**—Allow (default) or stop (with **no email dead-letter enable**) the saving of undeliverable e-mails.
- **domain**—Use a **hostname** or **IP address** to set the domain name from which e-mails are to appear to come (provided that the return address is not already fully-qualified). This is used in conjunction with the system hostname to form the full name of the host from which the e-mail appears to come. Use **no email domain** to reset to default (global settings). The rules are as follows:

- a. If an e-mail domain is specified using this command, it is always used. If the **hostname** has any dots in it, everything to the right of the first dot is stripped and the e-mail domain is appended.
- b. Otherwise, if the **hostname** has dots in it, it is used as is.
- c. Otherwise, the currently-active system domain name is used. This can come either from the resolver configuration, or from state dynamically instantiated by DHCP.
- **mailhub**—Use a **hostname** or **IP address** to set the mail relay to use to send notification e-mails. Use **no email mailhub** to clear the entry. The **mailhub** option must be sent for notifications to work.
- **mailhub-port**—Set the mail port to be used to send e-mails. Default is **25**. Use **no email mailhub-port** to reset to default.
- **notify**—Set handling of events and failures via e-mail.
 - **event <event_name>**—Enable or disable (with **no**) sending e-mail notifications for the specified event type. This does not affect autosupport e-mails. Autosupport can be disabled overall, but if it is enabled, all autosupport events (**process-crash**, and **liveness-failure** only, by default) are sent as e-mails. See "[email event name Options.](#)" for details. Set thresholds for these events using **stats**. Set SNMP traps for events using **snmp-server**.
 - **recipient <email_address>**—Add or delete (with **no**) an e-mail address from the list of addresses to send e-mail notifications of (all enabled) events, and specify:
 - **class**—Set event class. Each event type is classified as either **info** or **failure**. The specified recipient receives the intersection of the set of events specified by this command, and the set of events specified overall with the **email notify event <event_name>** command. See "[email class Options.](#)" for more details.
 - **detail**—Specify whether the e-mails this recipient is sent should be detailed or summarized. Each e-mail potentially has both a detailed and summarized form, where the detailed form has a superset of the information. Default is enabled.
- **return-addr**—Set the username or fully-qualified return address from which e-mail notifications are sent. If the string provided contains an at (@) sign, it is considered fully-qualified and is used as-is. Otherwise, it is considered just the username, and Media Flow Controller appends **@<hostname>.<domain>**. The default is **do-not-reply**, but this can be changed to **admin** or as desired in case something along the line does not like fictitious addresses. Use **no email return-addr** to reset to default.
- **return-host**—Include the hostname in the return address for e-mail notifications. This only takes effect if the return address does not contain an at (@) sign. Default is include. Use **no** to exclude hostname.
- **send-test**—Send a test e-mail to all of the configured notification e-mail recipients. This is useful to make sure the configuration works without having to wait for an event to occur.

email event name Options

Email **event name** options are:

- **process-crash**—A process in the system has crashed.
- **process-exit**—A process in the system unexpectedly exited.
- **liveness-failure**—A process in the system was detected hung.
- **cpu-util-high**—CPU utilization has risen too high.

- **cpu-util-ok**—CPU utilization has fallen back to normal levels.
- **paging-high**—Paging activity has risen too high.
- **paging-ok**—Paging activity has fallen back to normal levels.
- **disk-space-low**—File system free space has fallen too low.
- **disk-space-ok**—File system free space is back in the normal range.
- **memusage-high**—Memory usage has risen too high.
- **memusage-ok**—Memory usage has fallen back to acceptable levels.
- **netusage-high**—Network utilization has risen too high.
- **netusage-ok**—Network utilization has fallen back to acceptable levels.
- **disk-io-high**—Disk I/O per second has risen too high.
- **disk-io-ok**—Disk I/O per second has fallen back to acceptable levels.
- **unexpected-shutdown**—Unexpected system shutdown.
- **interface-up**—An interface's link state has changed to UP.*
- **interface-down**—An interface's link state has changed to DOWN.*
- **cpu-util-ave-high**—Average CPU utilization has risen too high.
- **cpu-util-ave-ok**—Average CPU utilization has fallen back to normal levels.

* Can be added to **info** events class with **email notify event <event_name>**.

email class Options

Email **class** options are as follows:

- **failure** events:
 - **process-crash**—A process in the system has crashed.
 - **unexpected-shutdown**—Unexpected system shutdown.
- **info** events:
 - **liveness-failure**—A process in the system was detected as hung.
 - **process-exit**—A process in the system unexpectedly exited.
 - **cpu-util-ok**—CPU utilization has fallen back to normal levels.
 - **cpu-util-high**—CPU utilization has risen too high.
 - **disk-space-ok**—File system free space is back in the normal range.
 - **disk-space-low**—File system free space has fallen too low.

show email

Email notification settings. This does not include SNMP traps, which are under the **snmp-server** command tree.

enable

Enter **Enable** mode.

```
enable
```

From **enable** enter **configure terminal** to go to **Configuration** mode. Use **disable** to go from **Enable** mode back to **Standard** mode.

errorlog

Configure error log options. See [“Reading the Error Log \(errorlog\)” on page 191](#) for usage information. See [“Configuring Media Flow Controller Service Logs \(CLI\)” on page 188](#) for task details, including information on log rotation, **level** options, and **module** options. See [“Reading the Error Log \(errorlog\)” on page 191](#) for usage information, status codes, and sub-codes.



NOTE: This log is mainly used for debugging purposes by Juniper Networks Support.

```
errorlog
  copy <SCP>
  filename <name>
  level {1 | 2 | 3 | 4 | 5-7}
  module <module>
  on-the-hour {disable | enable}
  rotate {filesize <integer> | time-interval <hours>}
  syslog replicate {disable | enable}
```

Notes:

- **copy**—Auto-upload (when the set **rotate** criteria is reached) the errorlog using the secure channel protocol (SCP), to the server specified using **hostname**. If **username** and **password** are provided, Media Flow Controller uses that for authentication of the SCP session. The **no** variant disallows auto-upload. See [“Terminology” on page 31](#) for the **scp** URL format); you must have an SCP server installed in order to send files to your machine.
- **filename**—Configure the name of the file where the error log is stored. Default is **errorlog.<num>.yyyymmdd_hour:min:sec** (numbered sequentially).
- **level**—Set the errorlog severity level. Each higher number **level** setting includes more messages. See [“Error Log Options.”](#) for details. **Severe** messages are always sent regardless of the chosen module.
- **module <module>**—Set errorlog module; default is **all**. Setting modules restricts error logging to only those modules. Enter modules on separate lines. See [“Error Log Module Options” on page 187.](#) for module names and descriptions.
- **on-the-hour**—Set hourly log rotation. Default is **no** (disabled).
- **rotate**—Media Flow Controller allows error log rotation based on file size or time.

- **filesize**—Set rotation based on file size. Media Flow Controller creates "error.log.1," "error.log.2," and so on up to "error.log.10," after which it wraps around. By default, **rotate filesize** is **100 MB**. We highly recommend not increasing the size; huge file transfers take a lot of time, and if there is a system reset, large volumes of data are at risk.
- **time-interval**—Set rotation based on time. Specify a time in hours after which the error log is rotated.
- **syslog replicate**—Specify whether (**enable**) or not error log messages are seen as part of syslog; default is **no** (disabled), error log is not seen as part of syslog.

```
show errorlog [continuous | last]
```

List error log settings; or use **last** to see the last few lines of the log and **continuous** to view the log as it is written.

exit

Leave configuration mode or log out of the system.

```
exit
```

Exit the current mode. From configuration mode, go to **enable** mode. From **enable** or **standard** mode, log out. Use **disable** to go from **enable** mode to **standard** mode.

file

Use these commands to manage stats, and tcpdump reports.

```
file
```

```
  debug-dump
```

```
    delete <filename>
```

```
    email <filename>
```

```
    upload <filename> {<URL> | <SCP>}
```

```
  stats
```

```
    delete <filename>
```

```
    move <source_filename> <dest_filename>
```

```
    upload <filename> {<URL> | <SCP>}
```

```
  tcpdump
```

```
    delete <filename>
```

```
    upload <filename> {<URL> | <SCP>}
```

Notes:

- **debug-dump**—Generate files useful for system debugging.
 - **delete**—Delete the specified debug dump file.
 - **email**—Send the specified debug dump in e-mail to the list of configured recipients for informational (**info**) events, regardless of whether they have requested to receive **detailed** notifications or not.
 - **upload**—Upload the specified debug dump file to the specified URL*.
- **stats**—Manipulate statistics report files. See **stats** for details.
 - **delete**—Delete a statistics report file by name.

- **move**—Rename (to a new location) a statistics report file.
- **upload**—Upload a statistics report file*.
- **tcpdump**—Manipulate tcpdump output files.
 - **delete**—Delete the specified tcpdump file.
 - **upload**—Upload the specified tcpdump file to the specified URL*.

* Only FTP and TFTP URLs, as well as SCP pseudo-URLs are supported for the destination. See [“Terminology” on page 31](#) for the **scp** URL format and requirements.

```
show files
  debug-dump [<filename>]
  stats [<filename>]
  tcpdump [<filename>]
```

Notes:

- **debug-dump**—List debug dump files. Use the **filename** option to list a summary of the contents of a particular debug dump file.
- **stats**—List statistics report files. Use the **filename** option to list the contents of a particular statistics report file.
- **tcpdump**—List tcpdump files. Use the **filename** option to list a summary of the contents of a particular tcpdump file.

fmsaccesslog

Configure Flash Media Server (FMS) access log options; this log is generated by the FMS server. In the Web-based interface, this log is displayed as FMSAccess Log under the **Logs** tab. This log is written to `/nkn/adobe/fms/logs/access.<nn>.log`, by default (you can access this using **application fms shell**, which takes you to the **fms** directory).

The `fmsaccesslog` lists all FMS server command executions. Streaming `fmsaccesslog` events include play, pause, seek, and stop events; session `fmsaccesslog` events include connect, disconnect, and connect-pending events, by default. For each of these events, the `fmsaccesslog` has at least some of the following fields logged:

- **date**—Date the event is logged
- **time**—Time the event is logged
- **c-ip**—Client IP address
- **c-proto**—Client connection protocol, either **rtmp** or **rtmpd**
- **s-uri**—URI of the FMS application

After FMS is installed (see [“Installing and Using FMS in Media Flow Controller \(CLI\)” on page 110](#)), you can access the `Logger.xml` file to read more about the FMS access log and make some configuration changes. This file is located in `/nkn/adobe/fms/conf`; you can access this using **application fms shell**, which takes you to the **fms** directory.

See [“Configuring Media Flow Controller Service Logs \(CLI\)” on page 188](#) for task details, including information on log rotation. See [“Reading the FMSAccess Log \(fmsaccesslog\)” on page 192](#) for usage information.

```
fmsaccesslog
  copy <SCP>
  filename <name>
  on-the-hour {disable | enable}
  rotate {filesize <integer> | time-interval <hours>}
  syslog replicate {disable | enable}
```

Notes:

- **copy**—Auto-upload (when the set **rotate** criteria is reached) the FMS access log using SCP (secure channel protocol), to the server specified using **hostname**. If **username** and **password** are provided, Media Flow Controller uses that for authentication of the SCP session. The **no** variant disallows auto-upload. See [“Terminology” on page 31](#) for the **scp** URL format); you must have an SCP server installed in order to send files to your machine.
- **filename**—Configure the name of the file where the FMS access log is stored. Default is **fmsaccesslog.<num>.yyyymmdd_hour:min:sec** (numbered sequentially).
- **on-the-hour**—Set hourly log rotation. Default is **no** (disabled).
- **rotate**—Media Flow Controller allows FMS access log rotation based on file size or time.
 - **filesize**—Set rotation based on file size. Media Flow Controller creates "fmsaccess.log.1," "fmsaccess.log.2," and so on up to "fmsaccess.log.10," after which it wraps around. By default, **rotate filesize** is **100 MB**. We highly recommend not increasing the size; huge file transfers take a lot of time, and if there is a system reset, large volumes of data are at risk.
 - **time-interval**—Set rotation based on time. Specify a time in hours after which the FMS access log is rotated.
- **syslog replicate**—Specify whether (**enable**) or not FMS access log messages are seen as part of syslog; default is **no** (disabled); FMS access log is not in syslog.

fmsedgelog

Configure Flash Media Server (FMS) edge log options; this log is generated by the FMS server. In the Web-based interface, this log is displayed as FMSEdge Log under the **Logs** tab. The **fmsedgelog** lists transactions to the FMS edge server; for example, “Connection rejected by server,” “Edge disconnected from core,” and “Listener started for clients.” This log is written to `/nkn/adobe/fms/logs/edge.<nn>.log`, by default (you can access this using **application fms shell**). This log is used for diagnostic purposes.

See [“Configuring Media Flow Controller Service Logs \(CLI\)” on page 188](#) for task details, including information on log rotation. See [“Reading the FMSEdge Log \(fmsedgelog\)” on page 193](#) for usage information.

```
fmsedgelog
  copy <SCP>
  filename <name>
  on-the-hour {disable | enable}
  rotate {filesize <integer> | time-interval <hours>}
  syslog replicate {disable | enable}
```

Notes:

- **copy**—Auto-upload (when the set **rotate** criteria is reached) the FMS edge log using SCP, to the server specified using **hostname**. If **username** and **password** are provided, Media Flow Controller uses that for authentication of the SCP session. The **no** variant disallows auto-upload. See [“Terminology” on page 31](#) for the **scp** URL format); you must have an SCP server installed in order to send files to your machine.
- **filename**—Configure the name of the file where the FMS edge log is stored. Default is **fmsedgelog.<num>.yyyymmdd_hour:min:sec** (numbered sequentially).
- **on-the-hour**—Set hourly log rotation. Default is **no** (disabled).
- **rotate**—Media Flow Controller allows FMS edge log rotation based on file size or time.
 - **filesize**—Set rotation based on file size. Media Flow Controller creates "fmsedge.log.1," "fmsedge.log.2," and so on up to "fmsedge.log.10," after which it wraps around. By default, **rotate filesize** is **100 MB**. We highly recommend not increasing the size; huge file transfers take a lot of time, and if there is a system reset, large volumes of data are at risk.
 - **time-interval**—Set rotation based on time. Specify a time in hours after which the FMS edge log is rotated.
- **syslog replicate**—Specify whether (**enable**) or not FMS edge log messages are seen as part of syslog; default is **no** (disabled); FMS edge log is not seen in syslog.

fuselog

Configure FMSConnector Log or fuselog options. While the streamlog records RTMP transactions, the fuselog records RTMP transaction details, including what URIs are accessed and how many bytes are returned by the FUSE module. This log is generated by Media Flow Controller.

See [“Configuring Media Flow Controller Service Logs \(CLI\)” on page 188](#) for task details including information on log rotation. See [“Reading the FMSConnector Log / fuselog” on page 194](#) for usage information.

```
fuselog
  copy <SCP>
  filename <name>
  on-the-hour {disable | enable}
  rotate {filesize <integer> | time-interval <hours>}
  syslog replicate {disable | enable}
```

Notes:

- **copy**—Auto-upload (when the set **rotate** criteria is reached) the fuse log using SCP, to the server specified using **hostname**. If **username** and **password** are provided, Media Flow Controller uses that for authentication of the SCP session. The **no** variant disallows auto-upload. See [“Terminology” on page 31](#) for the **scp** URL format); you must have an SCP server installed to send files to your machine.
- **filename**—Configure the name of the file where the fuse log is stored. Default is **fuselog.<num>.yyyymmdd_hour:min:sec** (numbered sequentially).
- **on-the-hour**—Set hourly log rotation. Default is **no** (disabled).

- **rotate**—Media Flow Controller allows fuse log rotation based on file size or time.
 - **filesize**—Set rotation based on file size. Media Flow Controller creates "fuse.log.1," "fuse.log.2," and so on up to "fuse.log.10," after which it wraps around. By default, **rotate filesize** is **100 MB**. We highly recommend not increasing the size; huge file transfers take a lot of time, and if there is a system reset, large volumes of data are at risk.
 - **time-interval**—Set rotation based on time. Specify a time in hours after which the fuse log is rotated.
- **syslog replicate**—Specify whether (**enable**) or not fuse log messages are seen as part of syslog; default is **no** (disabled); fuse log is not seen as part of syslog.

```
show fuselog [continuous | last]
```

List log settings; or use **last** to see the last few lines of the log, and **continuous** to view the log as it is written.

hostname

Set system hostname. See [“Configuring Interfaces, Hostname, Domain List, DNS, and Default Gateway \(CLI\)” on page 88](#) for task details.

```
hostname <hostname>
```

Set the system hostname. Use **no hostname** to clear the setting.

```
show hosts
```

List system hostname, DNS configuration, and static host mappings.

image

Manipulate system software images.

```
image
  boot {location <location_ID> | next}
  delete <image_name>
  fetch {<URL> | <SCP>} [filename <name>]
  install <image_filename>
    location <partition_number>
    progress {no-track | track}
    verify {ignore-sig | require-sig}]
  move <source_image_name> <dest_image_name>
  options require-sig
```

Notes:

- **boot**—Specify from which location the image should boot by default; there are only two locations to choose from so the options are **1** and **2** for location ID. If **next** is used, set the boot location to be the next one after the one currently booted from. This does not mean the next one after the one Media Flow Controller is currently set to boot from; thus the command is idempotent, and does not continue to cycle through all of the available locations.
- **delete**—Delete the specified image file.

- **fetch**—Download an image from the specified URL. If a filename is specified, the file is given that name; otherwise it is given whatever name it had on the server (the part after the last slash in the URL). See [“Terminology” on page 31](#) for the **scp** URL format).
- **install**—Install the specified image file. Arguments:
 - **location**—If no **location** is specified, the next one after the one currently booted from is chosen. You cannot install to the "active" location (the one which is currently booted). On a two-location system, this is never necessary, as there is only one legal choice, which is chosen automatically if no location is specified.
 - **progress**—override the CLI default for displaying progress (**cli default progress enable**): **no-track** overrides the set default value enabling tracking, and blocks progress tracking; **track** overrides the set value disabling tracking, and tracks installation progress. If the options are not specified the CLI default is in effect.
 - **verify**—The **require-sig** and **ignore-sig** options override the system-wide defaults for signature verification; **require-sig** causes the image install to fail if a valid signature is not found on the image, **ignore-sig** forces unsigned or invalidly signed images to be installed.
- **move**—Rename the specified image file. The destination image may not already exist.
- **options require-sig**—Cause all image installs to require a valid signature. If not set, a signature is not required, but if one is present it must be valid. Use **no image options require-sig** to disable.

show images

List all image files on the system, as well as what images are installed in the two locations, the active location (which was most recently booted from), and the default location (which is the default to boot from in the future). There may be overlap between these two lists.

interface

Configure network interfaces. See [“Configuring Interfaces, Hostname, Domain List, DNS, and Default Gateway \(CLI\)” on page 88](#) for task details. See also **bridge** for details on setting up bridge groups.

```
interface <interface_name>
  alias <alias_index> ip address <IP_address> <netmask>
  arp {enable | disable}
  bond <bonded_interface>
  bridge-group <bridge-group_name> [path-cost <cost>] [priority <integer>]
  comment <comment>
  dhcp [renew]
  duplex {half | full | auto}
  identify <number>
  ip address <IP_address> <netmask>
  mtu <mtu_size_in_bytes>
  shutdown
  speed {<speed> | 10 | 100 | 1000 | 10000 | auto}
  txqueuelen <number>
```

Create an interface with a name; use **no interface <interface_name>** to delete. Notes:

- **alias**—Add or delete (with **no**) a secondary address to the specified interface. The **alias_index** creates a pseudo interface; its address appears in the output of **show interface** under the primary interface's data.
- **arp**—Either **disable** or **enable** (default) Address Resolution Protocol (ARP) broadcasts on a specified interface. This option is specific to broadcast networks such as Ethernet or packet radio. It enables (or disables) the use of the ARP to detect the physical addresses of hosts attached to the network. Disabling ARP on an interface supports Direct Server Return, using the Server Load Balancer (SLB) to load-balance incoming requests to the server farm, and letting the return traffic from the server bypass SLB by being sent directly to the client. See [“Load Balancing with Direct Server Return” on page 159](#) for more information.
- **bond**—Add or delete (with **no**) the named interface from the specified bonding interface. See **bond** for more link aggregation details.
- **bridge-group**—Add or delete (with **no**) the named interface from the specified bridge group, and set options. See **bridge** for details on creating bridge groups.
 - **path-cost**—Set the path cost to the interface, used by the spanning tree protocol (STP) to determine the "best" path. Default is **10**.
 - **priority**—Set the priority of the interface in comparison with others going to the same subnet. Default is **128**.
- **comment**—Add or delete (with **no**) a comment for the specified interface.
- **dhcp**—Enable or disable (with **no**) use of DHCP on the specified interface. This gets the IP address and netmask via DHCP so those settings are ignored. Setting the IP address and netmask disables DHCP implicitly. Use **renew** to force a restart on the DHCP client for the specified interface. Default is **disabled**.
- **duplex**—Set or clear (with **no**) the interface duplex. Default is **auto**, which also sets interface **speed** to **auto**. Setting one of the manual settings, **half** or **full**, also sets the **speed** to a manual setting that is determined by querying the interface to find out its current auto-detected state. We recommend keeping the default (**auto**). *Changing the **duplex** setting can interfere with auto-configuration operations and should be avoided.*
- **identify**—Select an interface and set **identify <number>** to flash the LED of that interface that number of seconds.
- **IP_address**—Set or clear (with **no**) this interface's IP address and netmask. If you change the IP address of an interface, run the **service restart** command for the **mod_delivery** service on that interface.
- **mtu**—Set the interface MTU. Default is **1500**. We recommend keeping the default. Use **no interface <interface_name> mtu** to reset default.
- **shutdown**—Enable or disable (with **no**) the specified interface.
- **speed**—Set or clear (with **no**) the interface speed. Default is **auto**, which also sets duplex to **auto**. Setting one of the manual settings (generally **10**, **100**, or **1000**) also sets the interface **duplex** to a manual setting that is determined by querying the interface to find out its current auto-detected state. *Changing the **speed** setting can interfere with auto-configuration operations and should be avoided.*
- **txqueue1en**—Transmit Queue Length. Control the amount of data queued up for transmission. Use for performance tuning.

```
show interface [<interface_name> [configured] [brief]]
```

List information about the specified interface, or all interfaces if one is not named. Includes this information:

- **Admin up**—Whether or not the interface is enabled; **up** means it is enabled. Disable an interface with **interface <interface_name> shutdown** and re-enable an interface with **no interface <interface_name> shutdown**. Interfaces are enabled by default.
- **Link up**—Whether or not there is a cable plugged into that interface and it is “live,” being connected to something which is turned on at the other side; for example, a switch, router, or another computer.
- **Additional information**—The configured **IP address** and **Netmask** of the interface, the **Speed**, **Duplex**, **Interface type**, **Interface source**, **MTU**, **HW address** (hardware address), a **Comment** (if one is configured), plus current **RX** (transmissions received) and **TX** (transmissions out) statistics. Detailed information is given by default; use the subcommands to modify that.

Notes:

- **brief**—List abbreviated runtime state, the interface statistics are excluded.
- **configured**—List configuration of the interface, rather than its runtime state.

ip

Configure Internet protocol (IP) settings. See [“Configuring Interfaces, Hostname, Domain List, DNS, and Default Gateway \(CLI\)” on page 88](#) for task details.



NOTE: This command is intended for restricting access to management ports (CLI, SNMP GETs, Media Flow Activate, and so on) to specific IP address range/subnet for security. It is not meant to be used for the media delivery path, as it internally uses a heavy duty tool (Linux IP tables).

```
ip
  default-gateway {<next_hop_IP_address_or_Interface> [<interface>]}
  dhcp
    hostname <DHCP_hostname>
    primary-intf <interface_name>
    default-gateway yeild-to-static
    send-hostname <DHCP_hostname>
  domain-list <domain_name_for_resolving_hostnames> ...
  filter
    chain {FORWARD | INPUT | OUTPUT}
      clear
      policy {ACCEPT | DROP}
      rule {move | append | insert | set | modify} target <target>
        [<rule_arguments>]
    enable
  host <hostname> <IP_address>
  map-hostname
  name-server <IP_address>
  route <network_prefix> {<netmask> | <mask_length>} {<next_hop_IP_address>
```

```
| <interface>}
```

Notes:

- **default-gateway**—Set or delete (with **no**) the default route to the Internet.
- **dhcp**—Dynamic Host Configuration Protocol settings. An interface is only eligible to be the DHCP primary if it is "admin up", has DHCP enabled, and has gotten a DHCP lease. Ineligibility does not prevent an interface from being configured as the DHCP primary interface, but prevents it from actually being used as such. If the configured primary is eligible, it is chosen as the acting primary; otherwise, one of the eligible interfaces is chosen (the first, in alphabetical order).
 - **hostname <DHCP_hostname>**—Configure an alternate hostname for the DHCP client to send to the server during DHCP negotiation. This only applies when **ip dhcp send-hostname** has been enabled. The hostname may be unqualified or fully qualified. This is a global configuration command that affects all DHCP interfaces. Changing this setting forces a renewal of DHCP on all interfaces.
 - **default-gateway yield-to-static**—Whether or not a DHCP default gateway yields to a statically configured default gateway. Default is not to yield, so the DHCP default gateway is added to the statically configured one. If **yield-to-static** is set, DHCP's default gateway is not installed if there is already one statically configured.
 - **primary-intf**—Set or delete (with **no**) the interface from which non-interface-specific configurations (resolver and routes) are accepted via DHCP.
 - **send-hostname**—Configure whether the DHCP client should send a hostname to the server. By default, no hostname is sent during DHCP negotiation. The server may use and honor the hostname supplied by the client. By default, the client sends the system's hostname. This may be overridden with the command **ip dhcp hostname <dhcp-hostname>**. This is a global configuration command that affects all DHCP interfaces. Changing this setting forces a renewal of DHCP on all interfaces.
- **domain-list**—Add or delete (with **no**) domains to try unqualified hostnames in.
- **filter**—Configure IP filtering. The **ip filter** commands operate on nodes that are implemented using IP tables, they implicitly operate only on the filter table.
 - **chain <chain_name>**—Choose a chain name, FORWARD, INPUT, or OUTPUT and take an action:
 - **clear**—Delete all rules from the specified chain.
 - **policy <policy>**—Set the policy (default **target**) for a specified chain, or reset to its default (with **no**). All of the rules on this chain are overrides of this default. Only these **targets** are allowed:
 - **ACCEPT**—Accept all traffic by default for this chain. Default policy for OUTPUT chain.
 - **DROP**—Drop all traffic by default for this chain. Default policy for FORWARD and INPUT chains.
 - **rule <rule_number>**—Add a filtering rule to the specified chain (in the "filter" table), overwrite an existing rule, or delete an existing rule. If you specify a chain and rule using the **no** variant (**no ip filter chain <chain_name> rule <rule_number>**), just that rule is deleted and higher number rules are moved to close the gap. If you specify a chain and just the keyword **rule** using the **no** variant, all the rules for that chain are deleted and the chain's policy and baseline

are reset to their defaults. See [“ip filter chain rule arguments” on page 397](#) for **rule** arguments.

- **move** `<rule_number>` to `<rule_number>`—IP filter rules are applied in the order in which they are placed in the IP filter table. Use this command to move a rule from one place in the IP table to another. Surrounding rules are re-numbered to keep the numbers consecutive.
- **append tail** `target<target>` `<rule_argument>`—Add a new rule after all the existing rules.
- **insert** `<rule_number>` `target<target>` `<rule_argument>`—Insert a new rule before the existing rule with the specified number; the number specified must be that of an existing rule. The rule specified and all above it are re-numbered to make room for the new rule.
- **set** `<rule_number>` `target<target>` `<rule_argument>`—Specify the rule number of an existing rule to overwrite with the new rule. No attempt is made to merge the old rule and new; the old one is obliterated.
- **modify** `<rule_number>` `target<target>` `<rule_argument>`— Specify the rule number of an existing rule to modify. Only the parameters specified in this invocation are changed; everything else is left untouched, except that setting a criterion removes any corresponding **not-** criterion, and vice-versa.
- **enable**—Enable or disable (with **no**) IP filtering of network traffic. Default is disabled.
- **host**—Add or delete (with **no**) hostname/IP mappings for /etc/hosts.
- **map-hostname**—Set or delete (with **no**) a static host mapping for the current hostname.
- **name-server**—Add or delete (with **no**) DNS servers.
- **route**—Set or delete (with **no**) a static route. If it is called with only a network prefix and mask, this deletes all routes for that prefix.



NOTE: If you change the IP address for a host, you must run the **service restart** command for the **mod_delivery** service for that host.

```
show ip
  default-gateway [static]
  filter [all] [configured]
  route [static]
```

Notes:

- **default-gateway**—List the currently active default route, or the configured one, if **static** is used. This is redundant with **show ip route [static]**, provided for ease of use.
- **filter**—List network filtering rules currently installed on the system, omitting ones that were not configured on this system.
 - **all**—Show ALL rules currently installed, even those that did not come from your configuration. Those rules do not have numbers next to them, so the numbers correspond to the entries in configuration, NOT to IPtables' internal rule numbers.
 - **configured**—Show the current set of rules in this configuration.

- **route**—List the routing table in the system, which includes dynamic routes as well as any active static routes. Use **static** to list the static routes. The **ip route** command only works on devices that already have an IP address assigned.



NOTE: In the **show ip filter all** output, some rules may be shown without rule numbers. Numbers are shown only next to rules that were installed through configuration, and match the numbers of the corresponding rules in configuration. In some cases other components install their own rules, outside the scope of the configuration, these non-configured rules are left alone.

ip filter chain rule arguments

The following arguments are available for **ip filter chain <chain> rule <rule> target <target>**. All of the arguments after the target are optional. The following conditions apply:

- There must always be at least one rule.
- Each of the **not-...** arguments is mutually exclusive with the corresponding positive argument.
- If a given kind of restriction is not specified, there is assumed to be no restriction for that type of criteria.
- All criteria specified are “ANDed” together. There is no way to specify for criteria to be “ORed” within a single rule; rather, you would need to have multiple rules with the same target.



NOTE: An **<IPv4 netmask>** may be given either as a netmask (for example, 255.255.255.0); or as a mask length preceded by a forward slash (for example, /24).

- **comment <string>**—Specify a comment for the specified rule.
- **dest-addr <IPv4 prefix><IPv4 netmask>**—Match a specific destination address range.
- **dest-port <port_or_port_range>**—Match a specific destination port or port range.
- **dup-delete**—After adding or modifying the rule, delete all other pre-existing rules that are duplicates of it. By default, there is no duplicate detection, and creation of duplicates is freely permitted.
- **in-intf <interface_name>**—Match a specific (single) inbound interface.
- **not-dest-addr <IPv4 prefix><IPv4 netmask>**—Do not match a specific destination address range.
- **not-dest-port <port_or_port_range>**—Do not match a specific destination port or port range.
- **not-in-intf <interface_name>**—Do not match a specific inbound interface.
- **not-out-intf <interface_name>**—Do not match a specific outbound interface.
- **not-protocol <protocol>**—Do not match a specific protocol. See **protocol** target description, for details.

- **not-source-addr** <IPv4 prefix><IPv4 netmask>—Do not match a specific source address range
- **not-source-port** <port_or_port_range>—Do not match a specific source port or port range.
- **out-intf** <interface_name>—Match a specific (single) outbound interface.
- **protocol** <protocol>—Match a specific protocol. The available protocols are **tcp**, **udp**, **icmp**, and **all**. Not specifying a protocol is the same as specifying **protocol all**. Specifying **not-protocol all** will not match anything. If **tcp** or **udp** are selected for the protocol, you may specify source and destination ports as well (if **icmp** is selected, these options are either ignored, or produce an error.) The source or destination port may each be either a single number, or a range specified as <low>-<high>; for example, **10-20** would specify ports 10 through 20, inclusive. Only one port or port range may be specified per type; that is, one for source, and one for destination.
- **source-addr** <IPv4 prefix><IPv4 netmask>—Match a specific source address range.
- **source-port** <port_or_port_range>—Match a specific source port or port range.
- **state**—Match packets in a particular **state**. The **state** criteria has to do with the classification of the packet relative to existing connections. If there are more than one state, they should be separated by commas; for example, **ESTABLISHED,RELATED**. A packet can be in one of three states:
 - **ESTABLISHED**—It is associated with an existing connection which has seen traffic in both directions.
 - **RELATED**—It opens a new connection, but one which is related to an established connection.
 - **NEW**—It opens a new, unrelated connection.

ldap

```

ldap
  base-dn <string>
  bind-dn <string>
  bind-password <string>
  host <IP_address> order {last | <order_number>}
  login-attribute {<string> | uid | sAMAccountName}
  port <port>
  scope {one-level | subtree}
  timeout <seconds>
  version {2 | 3}

```

Notes:

- **base-dn**—Set the base distinguished name (location) of the user information in the LDAP server's schema. This is a string like **ou=users,dc=example,dc=com**, with no spaces.
- **bind-dn**—Enter the distinguished name to bind to the LDAP server. This can be left empty for anonymous login (the default).

- **bind-password**—Enter the password used when binding to the LDAP server. With anonymous login (**bind-dn** is ""), also let this be empty (the default).
- **host <IP_address>**—Add an LDAP server to the set of servers used for authentication; servers are tried in the order they appear in the server list. New servers are added to the end of the list of servers by default. You can use the **<order_number>** argument for control over server placement in the list. If **no ldap host <ip-address>** is specified, the host is removed from the list. The special keyword "last" moves the specified server to be last to be tried
 - **last**—Move the server to the end of the server list.
 - **<order_number>**—Move or add an LDAP server such that the specified server has the given order number. Other LDAP servers as moved as required. The order numbering starts at **1**.
- **login-attribute**—Set the attribute name that contains the login name of the user. The **no** variant resets to the default, **sAMAccountName**.
 - **<string>**—Enter a string for the attribute name that contains the user login name.
 - **uid**—Specify that the **uid** LDAP attribute contains the user login name.
 - **sAMAccountName**—Specify that the **sAMAccountName** attribute contains the user login name. This is the default **login-attribute** value.
- **port**—Set the port on the LDAP server to connect to for authentication. The **no** variant resets it to the default, port **389**.
- **scope**—Set the search scope for the user under the **base-dn**. The **no** variant resets to the default, **subtree**.
 - **one-level**—Search the immediate children of the **base-dn**.
 - **subtree**—Search at the **base-dn** and all its children.
- **timeout <seconds>**—Set (or reset to the default with **no**) a global communication timeout for all LDAP servers. Default is **5**. Range is **1-60**.
- **version**—Set the LDAP protocol version number to use. Version **3** (default) is the current standard. Version **2** is used by some older servers.

show ldap

List LDAP configuration settings.

license

Activate features with license keys.

```
license [delete <license_number>] [install <license_key>]
```

Install with a license key or **delete** a license number (automatically assigned in the order the licenses are added). If the key is invalid (for example, it could never have been a valid license), an error message is printed and it is not added. If the license is valid but there is something else wrong with it (it names a nonexistent feature, it is expired, and so forth) a warning message is printed but it is added.

show licenses

List all installed licenses and list all licensable features which are currently activated by a license. For each, show:

- A unique ID which is a small integer.
- The text of the license key as it was added.
- Whether or not it is valid and active.
- Which features it is activating.

logging

Configure event logging. See [“Configuring Media Flow Controller System Log” on page 198](#) for more information.

These settings configure the system log (syslog) that records all system activity such as user logins, configuration changes, and system condition changes. It does not record service activity or errors. The Media Flow Controller errorlog records service related errors but is mostly useful for debugging by Juniper Networks Support. Media Flow Controller provides several service-specific logs, detailed in [“Chapter 10, “Troubleshooting Media Flow Controller.”](#)

logging

```

fields seconds enable [fractional-digits<integer> | whole-
  digits<integer>]
files
  delete {current | oldest [<number_of_files_to_delete>]}
  rotation
    criteria {frequency<frequency>| size<megabytes>| size-
      pct<percentage>}
    force
    max-num <max_number_of_files_to_keep>
  upload {current | 1 | 2} {<URL | <SCP>} [</filename>]
format
  standard
  welf [fw-name <firewall_name>]
level cli commands {<severity_level> | none}
local
  override [class <class> priority {<severity_level> | none}]
    {<severity_level> | none}
receive
trap {<severity_level> | none}
<IP_address>
  trap
    override [class <class> priority {<severity_level> | none}]
      <severity_level>

```

Notes:

- **fields seconds**—Include an additional field in each log message that shows the number of seconds since the Epoch; default is disabled. This is independent of the standard syslog datetime at the beginning of each message in the format "Feb 25 18:00:00". Aside from indicating the year at full precision, its main purpose is to provide

subsecond precision. The precision can be controlled with the two **-digits** commands described. Except for the year, all of these digits are redundant with syslog's own datetime.

- **enable**—Enable the **seconds** field. Use **no logging fields enable** to disable.
- **fractional-digits**—Control the number of digits to the right of the decimal point; options are **1**, **2**, **3**, and **6**; which specify how many digits to the right of the decimal point to use. Truncation is done from the right, so you always get the <n> most significant digits.
- **whole-digits**—Control the number of digits to the left of the decimal point; options are **1**, **6**, and **all** (do not limit the number of digits to the left of the decimal point). Truncation is done from the left, so you always have the <n> least significant digits. Except for the year, all of these digits are redundant with syslog's own datetime.
- **files**—Manage log files.
 - **delete**—Delete log files:
 - **current**—Delete all current log files.
 - **oldest**—Force immediate deletion of the specified number of oldest local log files.
 - **rotation**—Configure automatic rotation of local logging files.
 - **criteria**—Configure what criteria decides when to automatically rotate local log files on local persistent storage. There are three mutually exclusive options:
 - **frequency**—Rotate based on time: **daily** (at midnight), **weekly**, or **monthly** (first day, at midnight). Default is **daily**.
 - **size**—Rotate log files that pass the specified **size** threshold.
 - **size-pct**—Rotate logs that pass the specified disk percentage.If a size criteria is chosen, the file size is checked hourly, so if it passes the threshold in the middle of the hour it is not rotated right away.
 - **force**—Force an immediate rotation of the local log files. This does not affect the schedule of auto-rotation if it was done based on time: the next automatic rotation still occurs at the same time it was previously scheduled. Naturally, if the auto-rotation was based on size, this delays it somewhat as it reduces the size of the active log file to zero.
 - **max-num**—Configure how many old local log files to keep. If the number of log files ever exceeds this number (either at rotation time, or when this setting is lowered), the system deletes as many as necessary, starting with the oldest, to bring it down to this number. Default is **10**.
 - **upload**—Upload a local log file to a remote host (specified with **URL** or **scp** path); specify which (available) log file first:
 - **current**—The current log file.
 - **1**—Archived, compressed log file "messages.1.gz".
 - **2**—Archived, compressed log file "messages.2.gz".To specify an archived log file, give its number as displayed by **show log files**. The current log file has the name "messages" if you do not specify a new name for it in the upload URL. The archived log files have the name "messages.<n>.gz" if you do not specify a new name in the URL, and are compressed with gzip regardless. See [“Terminology” on page 31](#) for the **scp** URL format and requirements.

- **format**—Set log messages format. The **no** variant resets the format to default (**standard**), whether or not **welf** is used with it. Arguments:
 - **standard** (default)—Squid standard format.
 - **welf**—Web trends Enhanced Log Format. Use **fw-name** to specify the firewall name that should be associated with each message logged in WELF format. If no firewall name is set, the hostname is used by default. Use **no logging format welf fw-name** to delete.
- **level cli commands**—Set the severity level at which user-executed CLI commands are logged. Default is notice. See [logging severity level](#), for details.
- **local**—Set local logging options.
 - **override**—Enable, and add or delete (with **no**) a per-class override on the logging level. All classes that do not have an override use the global logging level set with **logging local <severity_level>**. Use **no logging local override** to disable all class-specific overrides to the local log level. Default is enabled. The **no** variant that disables them leaves them in configuration, but disables them so the logging level for all classes is determined by the global setting. Use the **class** argument to divide log messages according to their origin. The default classes are **mgmt-core** (for mgmtd alone), **mgmt-back** (for other back end components), and **mgmt-front** (for front end components, utilities, and tests).
 - **<severity_level>**—Set the minimum severity of log messages to be saved on local persistent storage, also applies to log messages originating from other hosts; or use **none** to disable local logging (you can also use **no logging local**). See [logging severity level](#), for details. Default is **notice**.
- **receive** —Allow this system to receive log message from another host. Default is **disabled**. If enabled, only log messages matching or exceeding the severity specified with **logging local <severity_level>** are logged, regardless of what is sent from the remote host. Use **no logging receive** to disable.
- **trap <severity_level>**—Set minimum severity of log messages sent to syslog servers. This sets both the default for new servers, as well as the setting for all existing servers. The **no** variant sets the level to **none**, disabling logging to remote servers altogether (though the list of servers is not erased). This command does not affect console or local logging. See [logging severity level](#), for details.
- **<IP_address>**—Send syslog messages to the specified remote syslog server. Hostnames are not allowed. Use **no logging <IP_address>** to stop. Use the **trap** option (set minimum severity of log messages sent to the specified server) arguments, **override**, and **<severity_level>**, as described for **logging local**.

logging severity level

Logging **severity-level** options are described in [Table 39](#).

Table 39 Logging Severity Levels

| Level | Description |
|--------------|--|
| alert | Action must be taken immediately for functioning to continue. |
| crit | An unexpected error-causing condition or response for unknown reasons. |

Table 39 Logging Severity Levels (Continued)

| Level | Description |
|----------------|---|
| debug | Messages generated by the system debugging utility. |
| emerg | System is unusable or cannot recover. |
| err | Error conditions. |
| info | Normal but significant condition or response that does not affect operations. |
| none | Disable logging (nothing is logged for this class) |
| notice | Normal but significant condition or response that could affect operations (default). |
| warning | An anomalous condition that can be ignored and functioning continue, but may affect operations. |

show logging

List logging configuration settings.

show log [files <file_number>] [[not] matching <regex>]

View a local system log file. Arguments:

- If **files <file_number>** is specified, view an archived log file, where the number is from **1** up to the number of archived log files.
- If **[not] matching <regex>** is specified, the file is piped through the grep utility to only include lines either matching, or not matching, the provided regular expression.

show log continuous [[not] matching <regex>]

List the last few lines of the current log file, and then continue to list new lines as they come in, until the user hits Ctrl+C. This is done using the tail utility. If **[not] matching <regex>** is specified, only log lines matching, or not matching, the provided regular expression are printed.

Enclose all regex entries in double quotes; for example, "**^.*\example\.com**".

show log files

List local log files.

management

Configure which interface is named **eth0**. The naming of the interfaces is done during installation. For Juniper Networks VXA Series appliances, the interface names are set automatically and **eth0** will always be set properly. When installing on other machines, you normally configure the naming of the interfaces during installation, either interactively or by specifying the MAC address of the interfaces to be named **eth0** and **eth1**. Use this command if you did not configure **eth0** properly during installation, and re-installing is not an option.

After using this command, the machine must be rebooted immediately.

management interface <MAC_address>



CAUTION: Using this command to set **eth0** renames some or all of the Ethernet interfaces.

CAUTION: Management interface configuration is not required for Juniper Networks VXA Series appliances; it is set in the factory.

media-cache

Configure media caching parameters.

Tip! Find the names of various media cache using **show media-cache** (see description). See [“Managing the Media Flow Controller Disk Cache \(CLI\)” on page 103](#) for task details, including troubleshooting. See [“Caching and Origin Clustering” on page 45](#) for background information on Media Flow Controller caching.

media-cache

disk

```

cache-tier [sas | sata | ssd] admission-threshold <integer>
dictionary pre-load {disable | enable | stop}
group-read [sas | sata | ssd] {enable | disable}
mount-new
<string>
  cache {enable | disable}
  disk-type {sata | sas | ssd}
  format
  repair
  status {active | inactive}
external-eviction {sas | sata | ssd} watermark <high> <low>
internal-eviction {sas | sata | ssd} watermark <high> <low>

```

Notes:

- **disk**—Set media cache options.
 - **mount-new**—**EXEC**. Discover a newly inserted disk, mount it, name it, and make it part of the disk cache system.
 - **<string>**—Specify a media cache disk by name (use **show media-cache disk list** to get cache names), and set options:
 - **cache**—Enable or disable the disk for caching. When the system boots and starts detecting disks, it auto-calibrates the disk, puts it in static mode, and enables the disk for caching. Use **disable** as needed; for example, if disk contents are rendered not-to-be-cached, or to reformat the disk; see the **reformat** and **repair** commands. A newly inserted disk must be formatted before being enabled. The **init-cache** option, used during installation only, formats all the disks.
 - **disk-type**—Set the disk type; options are **SATA**, **SAS**, or **SSD**. This command is normally not needed; but in some cases (for example, HP P410 smart array

controller) the disks may not be assigned the correct cache-type. Default is correct auto-assignment.

- **format**—**EXEC** command. Use when you insert a disk into a running Media Flow Controller with the intent of not using the contents in the disk.
- **repair**—Not supported in Release 2.0.7.
- **status**—Make the media-cache **active** or **inactive**. Use **inactive** if you need to pull the disk for any maintenance purposes; for example, to upgrade to a higher capacity disk, replace a SATA disk with a SAS disk, or replace a failed disk. Media Flow Controller allows On-line Insertion and Removal (OIR) of hard disk drives (HDD). However, **the HDD MUST be made inactive to be removed**. When a new HDD is in the disk, it must be made active and (if necessary) enabled for caching.
- **cache-tier {sas|sata|ssd} admission-threshold <integer>**—Select a cache tier—**sas**, **sata**, or **ssd**—and set a threshold for session admission. Default value is **0** (zero) for all tiers. This command affects the queuing behavior of disk read requests. Providing a non-zero value for this enables the session admission control for the selected tier. If the number of queued requests to the tier exceeds the value of “threshold value x 4 x number of disks,” new requests to the tier are forwarded to the next lower tier or, if that tier is the lowest tier, to the origin where the object is available.
- **dictionary pre-load**—Allow (with **enable**, the default), or disallow (with **disable**), or cease (with **stop**) Media Flow Controller pre-loading—taking from disk and putting to RAM—the cache dictionary (metadata about all objects in cache).
- **group-read**—Select a tier, **sas**, **sata**, or **ssd** and choose to read either the entire block (with **enable**), or one object (URI) (with **disable**). Default is **disable** for SSD and **enable** for SAS and SATA.
- **external-eviction | internal-eviction**—Select a tier—**SAS**, **SATA**, or **SSD**—and set **watermark** high and low values. Watermarks are used to control how long data stays in cache before being written to disk. The **internal-eviction** thresholds must be set higher than the **external-eviction** thresholds, or external eviction will never process. Default values are:
 - **sas**—High watermark default = 90%; Low watermark default = 85%
 - **sata**—High watermark default = 90%; Low watermark default = 85%
 - **ssd**—High watermark default = 95%; Low watermark default = 90%

The **internal-eviction** command is more frequently run. It looks at only the hottest objects in the cache and makes sure that these objects are placed in the correct storage media. The **external eviction** command runs less frequently. It looks at hot and cold objects (larger number of objects) and makes sure that the coldest of objects are evicted from the cache.

To purge all objects when a disk cache is removed, reboot Media Flow Controller with the **reload** command.

```
show media-cache {cache-tier | controller 3ware | disk {list | <cache_name>}
                 | free-block | generic-config | group-read | watermark-internal}
```

List various **media-cache** characteristics.

Notes:

- **cache-tier**—List the cache tiers and their admission threshold setting.
- **controller 3ware**—List 3ware controller information on the disk.

- **disk**—List information on the disk.
 - **list**—List all disk drives, their names, information on the physical location, serial number, type, and capacity. Before you activate or enable a cache, run **show media-cache disk list** and get the name assigned to the disk to use in configuration.
 - **<cache_name>**—List information on the specified cache.
- **free-block**—Display the free-block threshold of the disk caches.
- **generic-config**—Display the current settings for **media-cache disk dictionary preload** and **media-cache disk rate-limit**.
- **group-read**—Display the **group-read** setting for each tier.
- **watermark-internal**—Display the high and low watermark settings for each tier.

Examples:

show media-cache cache-tier

```
SAS cache-tier admission threshold :20
SATA cache-tier admission threshold :12
SSD cache-tier admission threshold :1250
```

show media-cache controller 3ware

| Ctl | Model | (V)Ports | Drives | Units | NotOpt | RRate | VRate | BBU |
|-----|-------------|----------|--------|-------|--------|-------|-------|-----|
| c0 | 9690SA-4I4E | 1 | 1 | 1 | 0 | 1 | 1 | - |

show media-cache disk list

| Device | Type | Tier | Active | Cache | Free Space | State |
|--------|------|--------|--------|-------|------------|---------------|
| dc_1 | SATA | Tier-3 | yes | yes | 38890 MiB | cache running |
| dc_2 | SATA | Tier-3 | yes | yes | 68668 MiB | cache running |

show media-cache disk dc_2

```
Disk Cache Configuration & Status:
Device Name/Type: dc_2/SATA
Cache Tier: Tier-3
Activated: yes
Cache Enabled: yes

Free Space: 68596 MiB
Disk State : cache running
```

show media-cache generic-config

```
media-cache disk dictionary preload enabled : no
media-cache disk rate-limit enabled       : yes
```

show media-cache group-read

```
SAS group read enabled : yes
SATA group read enabled : yes
SSD group read enabled  : no
```

show media-cache watermark-internal

```
Tier  Hi Lo
SSD   9090
```

SAS 9085

mfdlog

Configure accesslog and errorlog ports and interfaces.

mfdlog

field-length <number>

interface <interface_name> **tcp port** <port_number>

Notes:

- **field-length**—Number of lines allowed per log field.
- **interface** <interface_name> **tcp port**—Interface and port on which mfdlog listens; default port is 7957.

namespace

A namespace is a named collection of parameters that set delivery policies in a granular manner; you can configure up to 256 namespaces. To set global delivery policies, see [delivery](#) for CLI details. See [“Media Flow Controller Namespaces” on page 50](#) for background information; see [Chapter 6, “Configuring Namespaces \(CLI\)”](#) for task details and implementation particulars.

```
namespace <name>
  cache-inherit <namespace_UID>
  cluster-hash {base-url | complete-url}
  delivery protocol
    http
      cache-index domain-name {exclude | include}
      client-request
        cache-control max-age <number> action serve-from-origin
        cache-hit action revalidate-always
        cache-index url-match <regex> map-to <map_string> [no-match-tunnel]
        cookie action {cache | no-cache}
        query-string action {cache [exclude-query-string] | no-cache}
        tunnel-all
        tunnel-override {auth-header | cache-control | cookie}
      client-response header <name> [<value>] action {add | delete}
      origin-fetch
        cache object-size range <minimum_KB_size> <maximum_KB_size>
        cache-age {content-type<string><secs> | content-type-any <secs>}
        cache-age-default <seconds>
        cache-fill {aggressive | client-driven}
        content-store media [cache-age-threshold<secs>] [object-size<bytes>]
        header set-cookie {any | <header>} cache {deny | permit}
        tunnel-override {cache-control-no-transform | object-expired}
      origin-request
        cache-revalidation {deny | permit} [method {get | head} [validate-header <header>]] [use-date-header]
        connect timeout <seconds> retry-delay <seconds>
        header <name> [<value>] action add ...
        host-header inherit incoming-req {deny | permit}
        read interval-timeout <seconds> retry-delay <seconds>
        x-forwarded-for {disable | enable}
      rtsp origin-fetch cache-age-default <seconds>
  domain {any | regex <regex> | <FQDN>}
  live-pub-point <name>
    caching {enable | disable}
    receive-mode on-demand
    status {active | inactive}
  match
    header {regex <regex> | <name> {any | <value>}} [precedence <number>]
    query-string {regex <regex> | <name> {any | <value>}} [precedence <number>]
```

```

uri {regex <regex> | <uri-prefix>} [precedence <number>]
virtual-host <IP_address> [<port>] [precedence <number>]
object [delete {all [pinned] | <URI> | <pattern>}] [list {all | <URI> |
<pattern>}] [revalidate all] [validity-begin-header <header>]
origin-server
  http {absolute-url | follow {dest-ip [use-client-ip] | header<header>
[use-client-ip bind-to <client_traffic>_NIC]} | server-map
<map_name> | <FQDN/path> [<port>]}
  nfs {server-map <name> | <FQDN:export_path> [<port>]}
  rtsp {follow dest-ip [use-client-ip] | <FQDN> [rtsp-rtsp | rtp-udp]}
pre-stage
  ftp user <username> password <password>}
  password {RADIUS | TACACS | <password>}
status {active | inactive}
virtual-player <name>

```

Use **no namespace <name>** to delete. In prefix mode, use **namespace <name> no...** to make changes to configurations.

Tip! At a minimum, a **namespace** configuration requires a **name**, **domain**, **origin-server**, **match** setting, and **status** activation. Other settings can be left at their defaults. Example:

```

namespace example
  domain www.example.com
  origin-server http sv05
  match uri /vod
  status active

```



NOTE: What you configure for **origin-server** also sets a proxy mode and, in some cases, certain defaults. See [“Using namespace for Proxy Configurations” on page 148](#) for details.

NOTE: Multiple HTTP or NFS origin servers can be configured with the **server-map** option. See [server-map](#) for CLI details, and [Chapter 8, “Configuring Media Flow Controller Server Maps”](#) for task details.

Notes:

- **cache-inherit**—Add the cache of the specified namespace to this namespace; see [“Using namespace cache-inherit” on page 142](#) for details.
- **cluster**—Not Supported.
- **cluster-hash**—Not Supported.
- **delivery protocol**—Set delivery protocol options; two delivery protocols and options can be configured. Entering the delivery protocol puts you into namespace delivery protocol configure mode; use **exit** when finished.
 - **http**—Configure options for HTTP delivery:
 - **cache-index**—Choose to **exclude** or **include** (default) the domain name from the cache-index when creating the cache-index name for an object.
 - **client-request**—Manipulate incoming client requests; see [“\(namespace\) delivery protocol http client-request”](#) for CLI details.
 - **client-response header**—Delete headers in, or add headers to, outgoing responses to client requests. Up to 16 **headers**, including the custom header X-

Accel-Cache-Control, can be configured with an **action** value (either **add** or **delete**). If you only enter a header **<name>**, the only action allowed is **delete**; if you enter a header **<name>** and **<value>**, the only action allowed is **add**.

- **origin-fetch**—Set policies for content fetched from origin; see [“\(namespace\) delivery protocol http origin-fetch”](#) for CLI details.
- **origin-request**—Set policies for content requested from origin; see [“\(namespace\) delivery protocol http origin-request”](#) for CLI details.
- **rtsp**—Configure options for RTSP delivery:
 - **cache-index**—Choose to **exclude** or **include** (default) the domain name from the cache-index when creating the cache-index name for an object.
 - **origin-fetch cache-age-default**—Specify a cache age value in case it is not specified in the data fetched from the origin server. Can be specified in addition to four **content-type** specifications. Default value is **28800** seconds (8 hours). See [“Using namespace delivery protocol {http | rtsp} origin-fetch cache-age” on page 143](#) for implementation details.
- **domain**—The defined fully qualified domain name (FQDN) or regular expression (REGEX) is matched with the incoming HOST header. If there is a match, the request is refined with a **match** value. See [“Using namespace domain <FQDN:Port>” on page 144](#) and [“Using namespace domain regex” on page 144](#) for task details.
 - **any** (default)—Allow all domains to use this namespace.
 - **regex**—Set the domain for this namespace with a regular expression. Enclose all **regex** entries in double quotes; for example, a regex for **www.example.com** plus **example.com** could be: **“^.*\example\.com”**.
 - **FQDN**—Enter a FQDN. The domain you enter should match whatever you have configured as HOST header; you can append a port number as well, if needed (and used in HOST header).
- **live-pub-point**—Add a live stream publishing service.
 - **caching**—Either **enable** or **disable** (default) caching for this **live-pub-point**.
 - **receive-mode**—Set a method for receiving live streaming:
 - **on-demand**—Contact the origin/publishing server when a request from the client is received.
 - **status**—Make **active** or **inactive** (default) the live-pub-point.
- **match**—Refine the path of incoming requests (enclose all regex entries in double quotes).
 - **header**—A header **<name>** and **<value>** or use **any** for any value of that header; can also be a **regex**. Optionally, set a **precedence**.
 - **query-string**—A defined query param **<name>** and **<value>** or use **any** for any value of that query param; can also be a **regex**. Optionally, set a **precedence**.
 - **uri**—A **<uri-prefix>**; can also be a **regex**. See [“uri-prefix” on page 34](#) for **uri-prefix** definition and usage. Optionally, set a **precedence**.
 - **virtual-host**—The IP address must be a /32 address; it can take a special value of **0.0.0.0**, which means any IP address. The **<port number>** specification is optional. To map requests by TCP port number only, set the IP address to 0.0.0.0 and configure the port number. If you set the domain to **any**, configure **virtual-host** IP to **0.0.0.0**, then requests can be assigned to a namespace based solely on the port number on which the request comes in to Media Flow Controller. Optionally, set a **precedence**.

- **precedence**—All **match** options can use the **precedence** argument to break ties between namespaces defined with the same **match** criteria. The lower the number, the higher the preference for that namespace; **0** (zero) is the default and highest. See [“Using namespace match <criteria> precedence” on page 145](#).
- **object**—**EXEC** command. Either **delete**, or view (**list**), the contents of a namespace specified with either a **uri** (can include a filename), a **<pattern>**, or **all** to delete or list the contents of all configured namespaces; use the argument **pinned** to delete all pinned objects. Use the **domain** and **port** arguments for mid-tier or virtual namespaces. Use **revalidate all** to validate the timestamp of each object in cache, even if not expired. See [“\(namespace\) object list | delete | revalidate.”](#) for CLI details. Use **validity-begin-header** to set a Header name that tells when the object should start being delivered.
- **origin-server**—Specify how Media Flow Controller determines and accesses origin for cache misses; see [“\(namespace\) origin-server.”](#) for CLI details.
- **pre-stage**—Set authentication options for pre-staging content:
 - **ftp user <username> password <password>**—Set a password or authentication credentials for the default, auto-created namespace FTP user, **<namespace>_ftpuser**, for FTP pre-staging content to Media Flow Controller. Only this user can do FTP pre-staging for that namespace. When opening the FTP session, log in as the auto-configured user (**<namespace>_ftpuser**); in this way, every FTP session transfers content only for a single namespace. See [radius-server](#) and [tacacs-server](#) for CLI details on setting authentication schemes.
 - **password**—Set a password, or use a configured RADIUS or TACACS+ password.
- **status**—Make **active** or **inactive** the namespace. You must deactivate the namespace to make large changes, such as the **match** value. A newly created namespace is inactive by default; you must explicitly activate it.
- **virtual-player <name>**—Associate a defined virtual player with a namespace; in this way, that namespace uses the specified virtual-player settings instead of the default Media Flow Controller settings configured with the **network connection** commands. Use **no virtual-player** to remove. See [virtual-player](#) for CLI details.

```
show namespace {list | <namespace_name> [counters | object list]}
```

Notes:

- **list**—List the namespaces in the system.
- **<namespace_name>**—List the settings of the specified namespace.
 - **counters**—Show the counters, with current values, for the specified namespace. These counters are mostly self-explanatory, a few that might not be are:
 - **HTTP Resource Monitoring**
 - **Client Active Sessions**—Number of active connections per namespace.
 - **Current Bandwidth**—Bandwidth consumed per namespace, in Mbps.
 - **Served Bytes**—Bytes of data served towards client per namespace.
 - **Transaction Per sec**—Transactions (number of requests) handled by namespace per second.
 - **Cache Hit Ratio (Bytes)**—Ratio between number of bytes served from origin to those served from cache.

– **Cache Hit Ratio (Req)**—Ratio between number of requests handled from origin to those served from cache.

- **object list**—See [“\(namespace\) object list | delete | revalidate” on page 418](#).

List the namespaces in the system, or details of a given namespace. When used with **<namespace_name>**, the output relevant to the namespace, such as the Active status, Precedence, URI Prefix, Delivery Protocol, Domain, Origin Server and options, Virtual Player (if any), Origin Fetch configuration, and Pre-stage FTP configuration, is given.



NOTE: Configuration changes, including a **namespace** deletion, might not be updated for up to 30 seconds. This is due to a deferred update scheme that requires an HTTP request. An internal probe ensures that such a request occurs at least every 30 seconds.

(namespace) delivery protocol http client-request

(Optional) Specify a list of policies for incoming client requests.

```
namespace <name> delivery protocol http client-request
  cache-control max-age <number> action serve-from-origin
  cache-index url-match <regex> map-to <map_string> [no-match-tunnel]
  cache-hit action revalidate-always
  cookie action {cache | no-cache}
  query-string action {cache [exclude-query-string] | no-cache}
  tunnel-all
  tunnel-override {auth-header | cache-control | cookie}
```

Notes:

- **cache-control max-age <number> action serve-from-origin**—Set a Max-Age value for the incoming request Cache-Control header. Default is **0** (zero) which invalidates the incoming Max-Age value causing a data fetch from origin.
- **cache-hit action revalidate-always**—Specify that requested objects in cache always trigger a timestamp revalidation. Default is disabled; when enabled, there is performance impact as every transaction is then revalidated. Additionally, the **namespace <name> object revalidate all** command is not executable when **revalidate-always** is configured, as it is configurable only for T-proxy namespaces.
- **cache-index**—Configure cache-index parameters.
 - **url-match <regex>**—Configure a regex expression to match on the request URL. The **<regex>** has a maximum character limit of **1024** characters (including NULL); if the URI exceeds this limit, the request is tunneled. Only one **url-match <regex>** expression per namespace is allowed. No PCRE regex is allowed, only GNU regex is allowed. The default value is NULL.
 - **map-to <map_string>**—Configure a **map-string** value (a string to map, or rewrite, the URL portion of the request) when a match is found. The **<map_string>** has a maximum character limit of **2048** characters (including NULL). The default value is NULL.
 - **no-match-tunnel**—Tunnel the request when no regex match is found.

- **cookie action**—Set an **action** for objects with a cookie (cookies are typically dynamic and often not appropriate for caching); either **cache** or **no-cache** (default).
- **query-string action**—Set an **action** for objects with a query-string (such objects are typically dynamic and often not appropriate for caching); either **cache** (and, optionally, **exclude-query-string** itself from the cache) or **no-cache** (default).
- **tunnel-all**—Configure this **namespace** to tunnel all incoming client requests directly to the origin server without further processing. Default is disabled.
- **tunnel-override**—Configure this **namespace** to increase the cache-hit ratio by overriding some tunneled transactions' reasons codes. All of the **tunnel-override** options are **disabled** by default.
 - **auth-header**—Cache requests containing the “Auth” header.
 - **cache-control**—Cache requests containing either “Cache-Control: No-Cache,” or “PRAGMA: No-Cache” headers.
 - **cookie**—Cache requests containing the “Cookie” header.

(namespace) delivery protocol http origin-fetch

(Optional) Specify a list of policies for data fetched from origin (**delivery protocol http** or **rtsp**); puts you in **namespace <name> delivery protocol {http | rtsp} origin-fetch** prefix mode; use **exit** to leave. Use **no namespace <name> delivery protocol {http | rtsp} origin-fetch** to reset configured values to their defaults.

```
namespace <name> delivery protocol {http | rtsp} origin-fetch
  cache object-size range <minimum_KB_size> <maximum_KB_size>
  cache-age {content-type <string> <seconds> | content-type-any <seconds>}
  cache-age-default <seconds>]
  cache-fill {aggressive | client-driven}
  content-store media [cache-age-threshold <seconds>] [object-size <bytes>]
  header set-cookie {any | <header>} cache {deny | permit}
  tunnel-override {cache-control-no-transform | object-expired}
```

Notes:

- **cache object-size range <minimum_KB_size> <maximum_KB_size>**—(**delivery protocol http** only) Specify an object size range, maximum and minimum, in KBs. If specified, only objects within the range are cached (if cacheable); objects outside the range are tunneled. The defaults for both values is **0** (zero); there is no size-based check for a cache or no-cache decision. Valid values are any positive integer or **0**.
- **cache-age content-type <string><seconds>**—(**delivery protocol http** only) Set cache aging by **content-type <string>** or the keyword **content-type-any**, and set a time value in **seconds**. Re-enter the command as needed. Example **cache-age content-type** entries: **application/flv 28800**, **application/mov 2880**, **application/3gp 288**, **application/f4v 28**. If the **cache-age content-type-any** argument is unspecified, the **cache-age-default** value is used. See [“Using namespace delivery protocol {http | rtsp} origin-fetch cache-age” on page 143fa](#) for implementation details.
- **cache-age-default**—Specify a cache age value in case it is not specified in the data fetched from the origin server. May be specified in addition to four **content-type** specifications. Default value is **0** seconds (objects are cached only in RAM). See [“Using](#)

[namespace delivery protocol {http | rtsp} origin-fetch cache-age](#) on page 143 for implementation details.

- **cache-fill**—(**delivery protocol http** only) Specify how data is fetched.
 - **aggressive**—Get all the data irrespective of how much the client requested; useful for reverse proxy namespace.
 - **client-driven**—(default) Fetch only as much data as the client requested; useful for transparent proxy namespace.
- **content-store media**—(**delivery protocol http** only) Set caching options.
 - **cache-age-threshold**—Set a time threshold for newly fetched content stored in **media** cache (nonvolatile) instead of RAM cache. Default is **60** seconds: new content is stored in RAM cache for only 59 seconds (the expectation being that the content will not be served for too long and is not worth storing in media cache). To have new content always stored in media cache, set this value to **0** (zero).
 - **object-size**—Set a size limit, in bytes, for storing objects in disk cache; default is **4096**. For example, a value of **4** means store in disk cache all fetched objects larger than 4 bytes; a value of **0** (zero) means every object irrespective of size is cached in disk (if not marked non-cacheable in the “Cache-Control” header). If the object size is less than this threshold, it is served from the Media Flow Controller RAM cache. Setting **object-size** can improve disk-cache performance since small objects need not be written into disk and can be served directly from the RAM cache.
- **header set-cookie**—(**delivery protocol http** only) Allows caching, with **permit**, or no caching (with **deny**, the default), if the specified **set-cookie <header>** exists.
- **tunnel-override**—(**delivery protocol http** only) Configure the tunnel reason code for responses that Media Flow Controller should cache.
 - **cache-control no-transform**—Cache responses with the “Cache Control: No-Transform” header.
 - **object-expired**—Cache responses that indicate an object has expired, and set a new expiry date as the current date plus the configured **cache_age**. This is discussed in more detail in the Media Flow Controller Administration Guide, in the section “Using namespace delivery protocol <protocol> origin-fetch cache-age”.

(namespace) delivery protocol http origin-request

(Optional) Specify a list of policies for data requested from origin with **delivery protocol http**; puts you in **namespace <name> origin-request** mode, use **exit** to leave. Use **no namespace <name> origin-request** to reset configured values to their defaults.

```
namespace <name> delivery protocol http origin-request
  cache-revalidation {deny | permit} [method {get | head} [validate-header
    <header>]] [use-date-header]
  connect timeout <seconds> retry-delay <seconds>
  header <name> [<value>] action add ...
  host-header inherit incoming-req {deny | permit}
  read interval-timeout <seconds> retry-delay <seconds>
  x-forwarded-for {disable | enable}
```

Notes:

- **cache-revalidation**—Specify that Media Flow Controller ask origin for cache age revalidation prior to deleting expired data by choosing **permit**, the default. This allows Media Flow Controller to proactively talk to the origin when the content is close to its expiry (20 percent of life left) due to a pre-set cache age, and ask for revalidation. Doing this can save network bandwidth toward origin as the entire content is not fetched, and only the headers need revalidation. To have Media Flow Controller delete content that has attained its set cache age, choose **deny**. Optionally set:
 - **method**—Set the HTTP method to use for a revalidation request. Default is **head**.
 - **get**—Use GET method on a cache-revalidation request.
 - **head**—Use HEAD method on a cache-revalidation request.
 Optionally, for either **get** or **head**, set:
 - **validate-header <header_name>**—Configure the HTTP response header to use to validate whether object was modified or not. Only allowed value for **<header_name>** is **Etag** (case sensitive).
 - **use-date-header**—Use the Date header for revalidation if the Last-Modified header does not exist.
- **connect**—Set options for connections to the target origin server; applicable only with an **origin-server server-map cluster-map** or **origin-escalation-map** configuration.
 - **retry-delay**—The amount of time, in milliseconds, the request is delayed and retried after a socket connect timeout. Default is **100** ms.
 - **timeout**—The socket connect timeout, in milliseconds. Default is **100** ms.
- **header action add**—Specify a header **<name>** and, optionally, **<value>** to be added to the incoming request; up to four headers can be added this way.
- **host-header inherit incoming-req**—Allow (with **permit**) or disallow (with **deny**) Media Flow Controller to set the HOST: header in the Media Flow Controller-to-origin HTTP REQUEST to the value found in the HOST: header in the incoming URL to Media Flow Controller. Default is **deny**.
- **read**—Set read options for connections to the target origin server; applicable only with an **origin-server server-map cluster-map** or **origin-escalation-map** configuration.
 - **interval-timeout**—The socket plus internal processing timeout, in milliseconds. Default is **100** ms.
 - **retry-delay**—The amount of time, in milliseconds, the request is delayed and retried after a read timeout. Default is **100** ms.
- **x-forwarded-for**—Allow (with **enable**) or disallow (with **disable**) Media Flow Controller setting the X-Forwarded-For header to the value of the client IP address when requests are sent from Media Flow Controller to origin upon a cache miss. Default is **enable**.

(namespace) origin-server

Specify where Media Flow Controller should go to fetch content from origin upon a cache miss (required). Use **no namespace <name> origin-server** to make changes. In Release 2.0.7, you can configure only one origin server; however, you can configure multiple HTTP or NFS

origin servers via the **server-map** option (described). See [Chapter 8, “Configuring Media Flow Controller Server Maps”](#) for implementation details.

```
namespace <name> origin-server
  http
    absolute-url
    follow {dest-ip [use-client-ip] | header <header> [use-client-ip bind-
      to <client_traffic>_NIC]}
    server-map <map_name>
    <FQDN/path> [<port>]}
  nfs
    server-map <name>
    <FQDN:export_path> [<port>]
  rtsp
    follow dest-ip [use-client-ip] |
    <FQDN> [<port#>] [rtsp-udp | rtp-rtsp]
```

Notes:

- **http**—Use HTTP for origin-server delivery; multiple origin servers can be specified using the **server-map** option.
 - **absolute-url**—Derive the origin server from the absolute URL set against the HTTP access method (GET, HEAD, and so forth). Use **absolute-url** to configure Media Flow Controller as a mid-tier proxy (if set, **show namespace** output has Proxy Mode: forward). Media Flow Controller uses the absolute URL to contact the origin-server. If **absolute-url** is configured and the incoming request (REQ header) does *not* have the absolute URL, then the request is rejected with the appropriate error code.
 - **follow**—Define alternate methods of determining which origin server to use:
 - **header**—Use a defined **<header_name>** in the request as the origin server. For example, you could set follow header HOST, and the value of the HOST header in the incoming request is used as the origin server. If the configured header does not exist, the request is rejected. The **<header name>** can be any of the well-defined headers OR a custom header. If set, show namespace output has Proxy Mode: virtual. In Release 2.0.7, the only allowed value for **header** is **host**. Optionally, set **use-client-ip bind-to <client_traffic_NIC>** to use a Media Flow Controller IP address when fetching an object from the origin server. This setting is used to create a transparent proxy deployment; for more information see [“Transparent Proxy Deployments” on page 66](#).
 - **dest-ip**—Use the destination IP address of the incoming request as the origin server. Optionally, set **use-client-ip** to use the client IP address in place of the origin server’s IP address in the request.
 - **server-map**—A defined **server-map**. If set, **show namespace** output has Proxy Mode: reverse. You can specify up to three server maps if they are either **cluster-map** or **origin-escalation-map** types; the order in which the maps are added to the namespace determines the order in which they are read.
 - **<FQDN/path>**—**hostname/IP address/path**, and (optional) **port**; default is **80**. If set, **show namespace** output has Proxy Mode: reverse.
- **nfs**—Use NFS for origin-server delivery; multiple origin servers can be specified using the **server-map** option.

- `server-map`—A defined `server-map`. You can specify more than one `server-map` only if they both are either `cluster-map` or `origin-escalation-map` types; the order the maps are added to the namespace is the order in which they are read.
- `<FQDN:export_path>`—`hostname/IP address:path`, and (optional) `port`; default is `2049`.
- `rtstp`—Use RTP/RTSP for origin-server delivery.
 - `follow dest-ip`—Use the destination IP address of the incoming request as the origin server. Optionally, set `use-client-ip` to use the client IP address in place of the origin server's IP address in the request.
 - `FQDN`—A `hostname` or `IP address`; and (optional) a `port`, default is `554`. Optionally choose an RTP transport mechanism for MFD to use when fetching media data from the origin streaming server, either `rtp-udp` or `rtp-rtsp` (interleaved); the default is to use what the client specifies.

(namespace) object list | delete | revalidate

EXEC command. Manipulate **objects** in the cache, on a namespace basis.

```
namespace <name> object {list | delete | revalidate} {all [pinned] | <URI> |
  pattern} [<domain>] [<port>]
```

Notes:

- `list`—Perform a **list** operation on the contents in a namespace.
- `delete`—Use **delete all pinned** to remove pinned objects from the namespace's cache. All objects stored by Media Flow Controller, RAM cache and disk cache, are stored as **UUID:/uri/filename**; this command derives the UUID from the specified **namespace**. Object deletion is tracked separately for each namespace. Object deletes issued for different namespaces occur in parallel. If **namespace <name> object delete** is issued for a namespace that is already processing an object delete, an error is displayed indicating that a purge is already in progress. Object deletion is done at a metered rate to minimize the impact on content delivery.
- `revalidate`—Use **revalidate all** to validate the contents of the cache present under the namespace. From the time when issued, Media Flow Controller serves the content to the client only after revalidating the content with the origin.



NOTE: The **domain** and **port** options only apply to namespaces with **proxy-mode mid-tier** or **proxy-mode virtual**. This way, objects cached for a given domain or domain and port are listed instead of all objects irrespective of domain.

Example **namespace** configuration and **object list | delete** actions:

```
namespace ns1
domain example.com
match uri /abc
```

Suppose your URL is **http://example.com/abc/def/file.flv**. To list an object and get its characteristics, issue the following command:

```
namespace ns1 object list /abc/def/file.flv
```

To delete an object with the same URL:

```
namespace ns1 object delete /abc/def/file.flv
```

To delete all the objects in that namespace's disk cache with the same URL:

```
namespace ns1 object delete all
```

To list all objects in a disk cache and create a file named with the UUID of the namespace, use this command. In the example, if the namespace had a UUID of 80213A2C, the file containing the list is 80213A2C.lst. Use the **upload** command to view the file.

```
namespace ns1 object list all
```

You can also list and delete based on patterns. For example, you can specify ***.flv** as a pattern. Media Flow Controller does not support a full regular expression for deleting or listing. The command **namespace ns1 object list all** is equivalent to **namespace ns1 object list /abc/def/***.



NOTE: For Release 2.0.7, only the asterisk (*) wildcard is available for pattern use. Asterisk (*) matches zero or more characters of any kind as indicated.

Example output for **namespace test_ns object list all**:

Objects in cache for namespace : test_ns

```
-----
*Loc  Size(KB)          Expiry              URL
*-----
RAM   512      Tue Nov 24 01:59:13 2009  example.local:80/tmp/ram/data/test.0.flv
RAM   512      Tue Nov 24 01:59:14 2009  example.local:80/tmp/ram/data/test.4.flv
RAM   512      Tue Nov 24 01:59:13 2009  example.local:80/tmp/ram/data/test.3.flv
RAM   512      Tue Nov 24 01:59:14 2009  example.local:80/tmp/ram/data/test.5.flv
*Loc  Size(KB)          Expiry              URL
*-----
dc_1  512      Tue Nov 24 01:59:14 2009  example.local:80/tmp/ram/data/test.4.flv
dc_1  512      Tue Nov 24 01:59:13 2009  example.local:80/tmp/ram/data/test.3.flv
dc_1  512      Tue Nov 24 01:59:14 2009  example.local:80/tmp/ram/data/test.5.flv
dc_1  512      Tue Nov 24 01:59:14 2009  example.local:80/tmp/ram/data/test.6.flv
```

network

Configure the network layer for Media Flow Controller; see [“Setting Network Connection Options \(CLI\)” on page 100](#) for task details. Many of these commands are also available in virtual players, see [virtual-player](#) for details. When set in a virtual player assigned a namespace, the virtual player values override these values.

network

```
connection
  assured-flow-rate {0 | <kbps>}
  concurrent session {64000 | <integer>}
  idle timeout {60 | <seconds>}
  max-bandwidth {0 | <kbps>}
  origin queue [max-delay] [max-num]
  socket interface-bind {disable | enable}
  resolver [asynchronous] [cache-timeout {auto | random | <seconds>}]
```

```

tcp
  fin-timeout <seconds>
  max-orphans <positive_number>
  max-tw-buckets <positive_number>
  memory low <KB> high <KB> maxpage <KB>
  path-mtu discovery {off | on}
  read-memory min <KB> default <KB> max <KB>
  syn-cookie [half-open-conn <positive_number>]
  write-memory min <KB> default <KB> max <KB>
  tunnel-only {disable | enable}

```

Notes:

- **connection**—Configure network parameters for connections. Use **no network connection <parameter>** to reset the default (except where indicated).
 - **assured-flow-rate**—Set the assured flow rate (AFR) for any connection. AFR is the minimum rate at which a connection is allowed to exist in the system. Connections usually get a bandwidth between this and the **network connection max-bandwidth**. Default is **0** (zero) = Media Flow Controller best effort. AFR is disabled by default; if needed, enable it with these configurations or with a **virtual-player** configuration. See [“Using Network Connection Assured Flow” on page 100.](#) for more details.
 - **concurrent session**—Define the maximum number of global client-side connections to accept, including connections accepted by all namespaces or all resource pools. Default is **64000** with Media Flow Controller license, **10** without it; maximum allowed is **250,000** in Release 2.1. You must have the Media Flow Controller license installed to change the default from 10.
 - **idle timeout**—Set socket idled-out time, in seconds; default is **60**. This is the time the network waits before closing a session when there is no data transfer in it.
 - **max-bandwidth**—Set the maximum bandwidth for a session. The actual session bandwidth is between the **network connection assured-flow-rate** and this value. Even if there is available bandwidth in the link, Media Flow Controller does not allocate more than this value for a session. When there is a full download, Media Flow Controller tries to allocate this value to the session. Default is **0** kbps (unbounded) with Media Flow Controller license, **200** kbps without it. Use **no network connection max-bandwidth** to reset default (unbounded). You must have the Media Flow Controller license installed to change the unlicensed default.
 - **origin queue**—Set connection parameters for the origin server:
 - **max-delay**—Set a maximum delay for the origin server queue. Default is **3**.
 - **max-num**—Set a maximum number of connections for the origin server queue. Default is **2**.
 - **socket interface-bind**—Either **enable** or **disable** socket bind on a physical interface. The default is **no (disable)**.
- **resolver**—Set DNS resolver options. Use the **no** form to reset a value to its default.
 - **asynchronous**—Enable asynchronous (non-blocking) DNS resolution.
 - **cache-timeout**—Configure cache timeout for entries in the DNS cache:

- `auto`—The internal algorithm sets a reasonable value given the amount of physical memory found on the system.
- `random`—Allows Media Flow Controller to set a randomly selected value.
- `<seconds>`—Set a value, in seconds.
- `tcp`—Set TCP parameters. Use the **no** form to reset a value to its default.
 - `fin-timeout`—Define the time that must elapse before TCP can release a closed connection and reuse its resources. Acceptable values are 5 to 120. The default value is **20** seconds.
 - `max-orphans`—Define the maximum allowable number of sockets not attached to any user file handle; use this to prevent simple DOS attacks. If this number is exceeded, orphaned connections are reset immediately and a warning is displayed. Acceptable values are 1024 to 524288, the default value is **131072**.
 - `max-tw-buckets`—Define the maximum allowed number of TIME_WAIT sockets held by the system simultaneously. If this number is exceeded, TIME-WAIT sockets are immediately destroyed and warning is displayed. Acceptable values are 1024 to 1048576. The default value is **180000**.
 - `memory [low] [high] [maxpage]`—Define the TCP stack value for memory usage, in pages of 4 KB.
 - `low`—When the amount of memory allocated by TCP is below this number of pages, TCP is memory allocation is acceptable. Acceptable values are 98304 to 393216, the default is **196608**.
 - `high`—When the amount of memory allocated by TCP exceeds this number of pages, TCP moderates its memory consumption and enters memory pressure mode, which is exited when memory consumption falls below the configured **low** value. Acceptable values are 131072 to 524288, the default is **262144**.
 - `maxpage`—Define the number of pages allowed for queuing by all TCP sockets. Acceptable values are 196608 to 6291456, the default is **6291456**.
 - `path-mtu discovery {off | on}`—Enable (with **on**) or disable (with **off**) the discovery of the network maximum transmission unit (MTU).
 - `read-memory [min] [default] [max]`—Define the memory usage values for the READ buffer.
 - `min`—Define the minimum size of the TCP socket RECEIVE buffer. Acceptable values are 4096 to 32768, the default is **4096**.
 - `default`—Define the default size of the TCP socket RECEIVE buffer. This value overrides **net.core.rmem_default** used by other protocols. Acceptable values are 32768 to 1048576, the default is **87380**.
 - `max`—Define the maximum size of automatically selected TCP socket RECEIVE buffers. This value does not override **net.core.rmem_max**. Acceptable values are 65536 to 16777216, the default is **16777216**.
 - `syn-cookie half-open-conn`—Define the maximum number of remembered connection requests (those that still had not received an acknowledgement from a connecting client); use this to prevent a SYN flood attack. After Media Flow Controller starts listening on the server socket, the value cannot be changed unless the LISTEN socket of the delivery module is re-initialized (**service restart of mod-delivery**). The default value is **1024**.

- `write-memory [min] [default] [max]`—Define the memory usage values for the SEND buffer.
 - `min`—Define the amount of memory reserved for TCP socket SEND buffers. Acceptable values are 4096 to 32768, the default is **4096**.
 - `default`—Define the default amount of memory allowed for TCP socket SEND buffers. This value overrides the `net.core.wmem_default` used by other protocols; it is usually lower than the `net.core.wmem_default`. Acceptable values are 32768 to 1048576, the default is **65536**.
 - `max`—Define the maximum amount of memory allowed for automatically selected TCP socket SEND buffers. This value does not override `net.core.wmem_max`. Acceptable values are 65536 to 16777216, the default is **16777216**.
- `tunnel-only`—Either **enable** or **disable** (default) tunneling all traffic to Media Flow Controller.

`show network [internal]`

List current **network** settings.

ntp

Configure a Network Time Protocol (NTP) server—Media Flow Controller has no configured default NTP servers.

```
ntp
  disable
  enable
  peer <IP_address> [disable] [version <number>]
  server <hostname | IP_address> [disable] [version <number>]
```

Notes:

- `disable` and `enable`—Enable or disable NTP overall. Default is **enabled**.
- `peer`—Add or delete (with **no**) an NTP peer.
 - `disable`—Temporarily disable this NTP peer. Use **no ntp peer <NTP_per_IP_address> disable** to re-enable.
 - `version number`—Allowable version numbers are **3** and **4**; default is **4**. If unspecified, default is used.
- `server`—Add or delete (with **no**) an NTP server.
 - `disable`—Temporarily disable this NTP server. Use **no ntp server <NTP_server_IP_address> disable** to re-enable.
 - `version number`—Allowable version numbers are **3** and **4**; default is **4**. If unspecified, default is used.



NOTE: Servers and peers start enabled; disabling is just a way of making them temporarily inactive without losing their configuration.

ntpdate

```
ntpdate <IP_address>
```

Set the system clock using the specified NTP (network time protocol) server. This is a one-time operation and does not cause the clock to be kept in sync on an ongoing basis. It generates an error if NTP is enabled, as the socket it requires is already in use.

```
show ntp
```

List NTP settings.

ping

```
ping [<options>] <hostname>
```

EXEC command. Network diagnostic tool **ping**. Invokes standard binary, passing command line arguments straight through.

radius-server

Remote Authentication Dial In User Service (RADIUS) is a networking protocol that provides centralized access, authorization and accounting management for people or computers to connect and use a network service. Use these commands to configure RADIUS server settings, either globally or by specified host. Use **no radius <parameter>** to clear settings.

```
radius-server
  host <IP_address>
    auth-port <port>
    key <string> | prompt-key
    login-lat-group
    retransmit <retries>
    timeout <seconds>
  key [<key_string>]
  login-lat-group <string>
  retransmit <retries>
  timeout <seconds>
```

Notes:

- **host**—Configure hosts to receive RADIUS authentication requests; overrides the set global defaults for all RADIUS hosts. Use **no radius host <IP_address>** to delete all RADIUS-specific settings for this host. To refine which host is deleted, use **no radius host <IP_address> auth-port <port>**. RADIUS hosts are tried in the order they are set.
 - **auth-port**—Set a port for RADIUS; default is **1812**. You can use the same IP address in more than one **radius host** as long as the **auth-port** for each is different. A UDP port number, specify **auth-port** immediately after the **host** option (if present).
 - **key | prompt-key**—For this host, set or clear, (with **no**) a shared secret text **string** or choose **prompt-key** and the user is prompted for the key; entries echo the asterisk (*) character, for greater security. If **no key** is set, the user is prompted for the key. Mutually exclusive with **prompt-key** argument.

- **login-lat-group**—For this host, set or clear (with **no**), the string that identifies the groups that the user is authorized to use when Login-service is defined as LAT (local area transport). If none is set, the user is prompted for the string; entries echo the asterisk (*) character, and the user must enter the same string.
- **retransmit**—For this host, set or reset to zero (with **no**), the number of times the client attempts to authenticate with any RADIUS server. Range is **0-5**, default is **1**. To disable retransmissions set it to **0** (zero).
- **timeout**—For this host, set or reset default (with **no**), the timeout for retransmitting a request to any RADIUS server. Range is **1-60**, default is **3**.
- **key**—Set or clear (with **no**) a global shared secret text string (key) used to communicate with any RADIUS host. If no key is set, the user is prompted for the key. Entries made at this prompt echo the asterisk (*) character, and the user must enter the same string twice.
- **login-lat-group**—Set or clear (with **no**), a global shared secret text string (key) used to communicate with any RADIUS server. If no key is set, the user is prompted for the key; entries echo the asterisk (*) character, and the user must enter the same string twice.
- **retransmit**—Set or reset to zero (with **no**), a (global) number of times the client attempts to authenticate with any RADIUS server. Default is **1**. To disable retransmissions set it to **0** (zero). Range is **0-5**.
- **timeout**—Set or reset to default (with **no**), the (global) timeout for retransmitting a request to any RADIUS server. Default is **3**. Sets. Range is **1-60**.

`show radius`

RADIUS settings.

ram-cache

Configure RAM cache options.

```
ram-cache
  cache-size-MB {0 | <number>}
  dict-size-MB {0 | <number>}
  revalidate-minsize <kilobytes>
  small-buffers scale-factor <number>
  sync-interval
```

Notes:

- **cache-size-MB**—Configure RAM cache size for Media Flow Controller processes; default **0** (zero) means AUTO (30 percent of memory is reserved for other processes). At least 512 MB is always reserved, regardless of the configured **cache-size-MB**.
- **dict-size-MB**—Set the dictionary size, in megabytes, or set **0** (zero) to use the internal algorithm to set the dictionary cache size.
- **revalidate-minsize**—Configure revalidation object size, in kilobytes; the default is **1024** kilobytes.
- **small-buffers scale-factor**—Configure the ratio between 32 KB buffers and 4 KB buffers in the RAM cache. Default is **2**. Acceptable values are between **0** and **8**; if $n = 1 - 8$, the ratio is $1:2^n$. For Release 2.0.5, Juniper Networks recommends a scale-factor of **3**; for Release 2.1, the recommendation is **4** or **5**.

- **sync-interval**—Control the time interval between which hotness data is written to disk from RAM. Since disk writes are expensive, this interval should not be very small. This value must be set smaller for transparent proxy caching than in reverse proxy because the number of objects and the rate of change of those objects is much higher in the transparent proxy mode. Default is **14400** seconds.

```
show ram-cache
```

Current **ram-cache** settings.

reload

Reboot or shut down the system.

```
reload
  force
  halt [noconfirm]
  noconfirm
```

Notes:

- **reload**—Reboot the system. If there are unsaved changes to the configuration, you may be prompted to save these changes (do a **write memory**) first before rebooting. The prompt is suppressed if confirmation of losing unsaved changes is disabled (with the **no cli default prompt confirm-unsaved** command). You may also be prompted to confirm the reload regardless of whether there are unsaved changes or not. This prompt is contingent on a separate configuration setting, controlled with the **[no] cli default prompt confirm-reload** command. If both prompts are enabled, and the configuration was unsaved, you are prompted twice.
 - **force**—Reboot the system immediately. This reboots the system, and there is no **halt** variant. There is also never any confirmation, whether or not there are any unsaved changes to the configuration
 - **halt**—Shut down the system if **reload halt**. If the system is busy performing another operation (requiring the management subsystem, which is almost any management operation), the regular **reload [halt]** command blocks until it is finished.
 - **noconfirm**—Suppresses the confirmations.

reset

Reset configuration, delete logs, and all other data.

```
reset factory [keep-all-config] [keep-basic] [reboot]
```

Notes:

- **reset factory**—‘Scrub’ the system clean, resetting it entirely to its factory state. This does not just involve configuration (done with the **configuration revert** command instead), but everything on the system: logs, stats, CLI command history, system image files (not the image installations), as well as resetting the configuration to factory defaults and deleting all other configuration files. The system halts after this process, unless the **reboot** option is set, in which case it reboots.

- **keep-all-config**—Preserve everything in the active configuration file, and also do not delete any other configuration files. You are prompted for confirmation before honoring this command, unless confirmation is disabled with the **no cli default prompt confirm-reset** command.
- **keep-basic**—Preserve licenses in the active configuration file.
- **reboot**—Reboot system after reset (instead of halting).

scheduler

Set scheduler options. Requires **service restart mod-delivery**.

```

scheduler
  tasks deadline <integer>
  threads {deadline | realtime} <number_of_threads>

```

Notes:

- **tasks**—Configure the number of **deadline** tasks. Default is **256**. Valid Input values are **256, 512, or 1024**.
- **threads**—Configure the number of **deadline** and **realtime** scheduler threads (glob_rtsched_num_core_threads). For deadline, the valid input is **1**. For realtime, the valid input range is **1** through **8**; default is **2**.

server-map

```

server-map <name>
  file-url <URL> refresh-interval <seconds>
  format-type <type> {host-origin-map | cluster-map | nfs-map | origin-
    escalation-map}
  node-monitoring heartbeat
    allowed-fails <integer>
    connect-timeout<ms>
    interval<ms>
    read-timeout<ms>
  refresh-force

```

Media Flow Controller can use an XML file to resolve incoming client requests to the correct origin server when Media Flow Controller encounters a cache-miss. Create the XML file following the conventions outlined in [Chapter 8, “Configuring Media Flow Controller Server Maps.”](#) and save it; then use these commands to **name** the server map, reference it with a **file-url**, provide a **format-type**, and set other parameters. After it is configured, assign the **server-map** to a **namespace origin-server**. See [“Chapter 8, “Configuring Media Flow Controller Server Maps.”](#) for information on creating and configuring **server-map** XML files.

Notes:

- **node-monitoring**—Set cluster and origin escalation node monitoring options; only allowed with **format-type cluster-map** or **format-type origin-escalation-map**.
- **allowed-fails**—Specify how many request failures are allowed before the node is declared down. Default is **3**, minimum allowed value is **0** (zero); maximum value is **32**.

- **connect-timeout**—Specify the allowable time, in milliseconds, for the socket connect to complete.
- **heartbeat-interval**—Specify the time, in milliseconds, for nodes to wait before sending a “heartbeat” signal to the other nodes indicating availability status.
- **read-timeout**—Specify the allowable time, in milliseconds, for the socket read to complete after the connection is established.
- **file-url**—Specify the file that contains the table of server maps. The mapping file is refreshed according to the specified **refresh-interval** time. Permitted values for time are **0** (no refresh) or **300** seconds (minimum) to **86400** seconds (maximum). Default is **0**.
- **format-type**—Specify the DTD type of XML file defining the **server-map**. Use **no server-map <name> format-type <type>** to delete.
 - **cluster-map**—Allows use of consistent hashing. This **format-type** can be used in conjunction with a defined **origin-escalation-map**.
 - **host-origin-map**—Maps the incoming HOST header to a defined origin.
 - **nfs-map**—Allows use of NFS publishing points for origin.
 - **origin-escalation-map**—Allows multiple, defined origins to be used, sequentially based on defined **weight**, should an origin fail. This **format-type** can be used in conjunction with a defined **cluster-map**.
- **refresh-force**—Force a refresh of the server map pointed to by the specified **file-url**; can also be set in the NFS server map XML file (the Media Flow Controller CLI setting overrides the NFS XML file setting), this feature is not yet available for the HTTP server map XML file.

```
show server-map [<name>]
```

Display the list of, and settings for, all configured server-maps or the specified server-map.

Example:

```
show server-map
```

```
Server-map: smNFS1
Format-Type:
Map File: http://mapfile.example.com/nfs/path/filemapA.xml
Refresh Interval: 60
```

server-map Example for NFS Origin

For an incoming URL of http://www.example.com/nfs_mnt1/video1.flv, and a **server-map** configuration of:

```
MFC (config) # server-map smNFS1
MFC (config server-map smHTTP1) # format-type nfs-map
MFC (config server-map NFS1) # file-url http://mapfile.example.com/nfs/path/
filemapA.xml refresh-interval 300
MFC (config server-map NFS1) #
```

Media Flow Controller would scan the URL, extract **nfs_mnt1** and use this defined **namespace** to index into the **server-map** file fetched from the set **file-url**. The fetched file (**filemapA.xml**) would have an entry (or multiple entries) with **nfs_mnt1** indicating the NFS mount point.

See ["Chapter 8, “Configuring Media Flow Controller Server Maps.”](#) for additional examples.

server-map Example for HTTP Origin

For an incoming URL of **http://www.example.com/vod/video1.flv**, and a **server-map** configuration of:

```
MFC (config) # server-map smHTTP1
MFC (config server-map smHTTP1) # format-type host-origin-map
MFC (config server-map smHTTP1) # file-url http://mapfile.example.com/vod/
path/filemapB.xml refresh-interval 300
MFC (config server-map smHTTP1) #
```

Media Flow Controller would scan the URL, extract **www.example.com**, and use this defined **namespace** to index into the **server-map** file fetched from the set **file-url**. The fetched file (**filemapB.xml**) would have an entry (or multiple entries) indicating the location of the defined origin-servers and corresponding HOST headers; the **format-type** tells Media Flow Controller that this type of XML file is for HTTP origin.

See "[Chapter 8, "Configuring Media Flow Controller Server Maps."](#)" for additional examples.

service

Media Flow Controller service options.

```
service
  flash-led ethernet [<count>]
  restart <service_name>
  snapshot mod-delivery {disable | enable}
  stop
```

Notes:

- **flash-led ethernet**—Flash the Ethernet LEDs in sequence. Optionally specify how many loops of flashing with **<count>**; up to 25 loops allowed.
- **restart**—Media Flow Controller requires some services to be restarted after an IP address or delivery protocol port change. For **<service_name>** you can use the following:
 - **mod_delivery** (Delivery Engine)
 - **mod_file-publisher** (File Publisher)
 - **mod_ftp** (FTP module)
 - **mod_live-publisher** (Live Streaming Publisher)
 - **mod-log** (Media Flow Controller logging)
 - **mod_oom** (Offline Origin Manager)
 - **mod_rtmp-admin** (RTMP Administration Service)
 - **mod_rtmp-fms** (FMS RTMP Service)
- **snapshot mod-delivery**—Choose to either **enable** or **disable** (default) a snapshot (coredump) to be automatically generated whenever the **mod-delivery** service crashes.
- **stop**—Stop a service. Currently, you can stop the **mod-file-publisher** and **mod-live-publisher** services.

```
show service [<name>]
```

List status of the of all services, or the specified service.

show

View information; see [“Viewing Information Using Show Commands” on page 211](#) for details. Most are **EXEC** commands. There are many **show** commands described at the end of the appropriate sections. Additionally, there are special **show** arguments:

- **bootvar**—Installed system images and boot parameters.
- **counter**—System statistics.
- **hosts**—Hostname, DNS configuration, and static host mappings.
- **interfaces configured**—Interface configurations.
- **memory**—System memory usage.
- **media-cache controller**—List media cache controllers; currently only **3ware**.
- **ram-cache**—The current size of the RAM cache and default RAM cache configurations.
- **running-config**—Commands to recreate current running configuration.
- **service mod-rtmp-fms** and **service mod-rtmp-admin**—Lists the status of the FMS service; either **running** or **stopped**. Use **service restart** command if needed.
- **stats cpu**—CPU statistics.
- **users**—Information about user logins.
- **version**—Version information for current system image.
- **whoami**—The identity and capabilities of the current user.

```
show version [concise]
```

List version information for the currently running system; shows each field with a description. The **concise** variant fits it onto one line, in a form suitable for a bug report, and so forth.

```
show ram-cache
```

List current **ram-cache** settings.

```
show configuration
  audit
  files [<filename>]
  full
  running [full]
```

List the CLI commands needed to bring the state of a fresh system up to match the current persistent state of this system. A short header is included, containing the name and version number of the configuration, in a comment.

Notes:

- **audit**—List settings for configuration change auditing.
- **files**—If no **filename** is specified, list the configuration files in persistent storage. If **filename** is specified, list the commands to recreate the configuration in that file; only non-default commands are shown.
- **full**—Same as **show configuration** but includes commands that set default values.
- **running**—Same as **show configuration** except that it applies to the currently running configuration, rather than the active saved configuration.



NOTE: Commands that would set something to its default are not included—so this command on a fresh configuration produces no output, except the header.

NOTE: This does not include changes that have not yet been written to persistent storage.

```
show running-config [full]
```

The **show running-config** commands perform the same functions as the **show configuration** commands and are included for ease-of-use.

slogin

```
slogin [<options>] <hostname>
```

EXEC command. Invokes the SSH client. You are returned to the CLI when SSH finishes.

snmp-server

Configure SNMP server options. To set trap event notification recipients, use the **email** commands.

At this time, Media Flow Controller does not support provisioning over SNMP V3.

```
snmp-server
  community <community_name> [ro]
  contact <contact_name>
  enable [communities] [traps]
  host <IP_address>
    disable
      traps [version <trap_version> | 1 | 2c] [<community_string>]
  listen [enable] [interface <interface_name>]
  location <system_location>
  traps event <event_name>
  user {<user_name> | admin} v3
```

Notes:

- **community**—Set a community name for either read-only (**ro**) or read/write (**rw**) SNMP requests. Default, and if unspecified, is **ro**. The **read-only** community means only queries are performed. Use **no** to reset default. In Release 2.0.7 only SNMP **ro** (Read-Only) is supported.
- **contact**—Set (remove with **no**) the **syscontact** variable served from the System MIB in MIB-II.
- **enable**—Enable the SNMP server. Use **no snmp-server enable** to disable; this stops serving SNMP variables and the sending of SNMP traps.
 - **communities**—Enable or disable (with **no**) community-based authentication on this system. If disabled, the community configured is ignored.
 - **traps**—Enable or disable (with **no**) sending SNMP traps from this system. The SNMP server must be enabled first. See "[snmp traps.](#)" for details. Traps are only sent if there are trap sinks configured and enabled with **snmp-server host**.

- **host <IP_address>**—Add or delete (with **no**) hosts to receive SNMP traps.
 - **disable**—Temporarily disable sending traps to this host. All trap sinks are created enabled. Use **no snmp host <IP_address> disable** to re-enable.
 - **traps**—Send traps to the specified host. This setting is only meaningful if traps are enabled, though the list of hosts may still be edited if traps are disabled.
 - **version**—Specify the SNMP version of traps to send to this host.
 - **community string**—Set a password for the reading and writing of SNMP traps; this is an alternate option to the default community string, which is "public". If you set this, it must be set on the server and the client.
- **listen**—Configure SNMP server interface access restrictions.
 - **enable**—Enable or disable (with **no**) the listen interface restricted list for SNMP. If enabled and at least one non-DHCP interface is specified in the list, the SNMP connections are only accepted on those specified interfaces. When disabled, SNMP connections are accepted on any interface. Default is enabled.
 - **interface**—Add or delete (with **no**) interfaces to the **listen** list. If the interface is also running as a DHCP client, it is as if the interface was not added. If DHCP is later turned off on this interface, it is as if the interface was then added to the listen list.
- **location**—Set the **syslocation** variable served from the System MIB in MIB-II.
- **traps event <event_name>**—Specify which types of events to send as SNMP traps. By default, the entire list of notify-able events are sent as SNMP traps to any declared trap sinks. Use **no** to delete. See "[snmp traps events.](#)" for details.

user <username> v3—Not supported in Release 2.0.7. **show snmp [user] [engineID]**

Notes:

- **snmp**—All SNMP settings, except for what is displayed by the other **show snmp** commands described.
- **user**—Not supported in Release 2.0.7.
- **engineID**—SNMP engine ID of this system.

snmp traps

The traps sent by the SNMP agent are:

- Cold boot (may include SNMP configuration having been changed)
- Link up/down
- CPU load too high
- CPU load no longer too high
- Paging activity too high
- A process has crashed
- A process has exited unexpectedly

snmp traps events

Media Flow Controller generates a number of traps to notify you about critical system events. You can configure Media Flow Controller to send SNMP traps/alarms to a 3rd party network management system. SNMP traps notify-able events are described in [Table 40](#).

Table 40 SNMP Traps Notify-able Events

| Trap | Description |
|----------------------------------|---|
| <code>cpu-util-high</code> | CPU utilization has risen too high. |
| <code>cpu-util-ave-ok</code> | CPU utilization has fallen back to acceptable levels. |
| <code>disk-io-high</code> | Disk I/O per second has risen too high. |
| <code>disk-space-low</code> | Filesystem free space has fallen too low. |
| <code>interface-down</code> | An interface's link state has changed to down. |
| <code>interface-up</code> | An interface's link state has changed to up. |
| <code>liveness-failure</code> | A process in the system was detected as hung. |
| <code>memusage-high</code> | Memory usage has risen too high. |
| <code>netusage-high</code> | Network utilization has risen too high. |
| <code>paging-high</code> | Paging activity has risen too high. |
| <code>process-crash</code> | A process in the system has crashed. |
| <code>process-exit</code> | A process in the system unexpectedly exited. |
| <code>smart-warning</code> | Smartd warnings. |
| <code>unexpected-shutdown</code> | Unexpected shutdown. |

ssh

Configure your SSH client and server. See [ssh client](#) and [ssh server](#).

ssh client

Configure SSH (secure sockets shell) client options.

```
ssh client
  global
    host-key-check {ask | no | yes}
    known-host <known_host_entry>
  user {<username> | admin | cmcrendv | monitor}
    authorized-key sshv2 <public_key>
    identity {<key_type> | rsa2 | dsa2}
      generate
        private-key [<key>]
        public-key <key>
      known-host <IP_address> [remove]
```

Notes:

- `global`—Configure global SSH client settings.
 - `host-key-check`—Set SSH client configuration to control how host key checking is done. Use **no** to disable host key checking. Arguments:

- **ask**—Prompt the user to accept new host keys, but disallow connection if there is already a known host entry that does not match the one presented by the host.
- **no**—Accept unknown keys and add them to the relevant known hosts file.
- **yes**—Only permit connections if a matching host key is in the known hosts file.
- **known-host <known_host_entry>**—Add or delete (with **no**) an entry to the global known-hosts configuration file.
- **user <username>**—Configure an SSH user.
 - **authorized-key sshv2 <public_key>**—The specified key is added to the list of authorized SSHv2 RSA or DSA public keys for this user account. These keys can be used to log into the user's account. The specified user must be a valid account on the system. As keys are added, an implicit ID is associated with the key; this is to make key deletion easier. If a key is pasted from a cut buffer and displayed with a paging program, it is likely that newline characters have been inserted, even if the output was not long enough to require paging; most **show** command output is displayed this way, as paging is enabled by default in the CLI. Specify **no cli session paging enable** before doing the **show** command to prevent the newlines from being inserted. Use **no ssh client user <username> authorized-key sshv2 <key_id>** to delete a public key from the specified user's authorized key list. The key identifier can be found with **show ssh client**.
 - **identity <key_type>**—Set SSH client identity options for the specified user.
 - **generate**—Generate a new identity (private and public keys) for the specified user name. The given user name must correspond to a valid local user account. When the keys are generated, the private key is written to the user's **.ssh** directory in an appropriately named file (for example, **id_dsa**). This identity can be used when the user connects from the system to another host with **slogin**. DSA and RSA v2 keys for SSHv2 can be generated using **dsa2** or **rsa2** as the **key-type**.
 - **private-key**—Set private key SSH client identity for the specified user. An optional passphrase may be specified for the private key.
 - **public-key**—Set public key SSH client identity for the specified user.
Set the public or private key of specified type for the specified user name. This is an alternative to generating the key with the **generate** command and is also used for reverse mapping generated keys. If the **private-key** command is used with no **key** given, the user is prompted for the key. Entries made at this prompt echo the asterisk (*) character, and the user must enter the same string twice. Use **no ssh client user <username> identity <key_type>** to delete the public/private keys for the specified user; any private key file in a valid user **.ssh** directory is deleted.
 - **known-host <IP_address> remove**—Delete a known host from the specified user's **.ssh** **known_hosts** file.

show ssh client

SSH client identities (public/private keys) and the per user list of authorized keys for the users.

ssh server

Enable or disable, and configure SSH (secure sockets shell) server options.

```
ssh server
  enable
  host-key
    generate
      <key_type> {private-key <key> | public-key <key>}
  listen [enable] [interface <interface_name>]
  min-version {1 | 2}
  ports <port> [<port2> ...]
  x11-forwarding enable
```

Notes:

- **enable**—Enable (default) or disable (with **no**) the SSH server. If the SSH server is disabled, the CLI is only accessible over the serial console; this does not terminate existing SSH sessions; it only prevents new ones from being established.
- **host-key**—Manipulate host keys for SSH:
 - **generate**—Regenerate new host keys for the SSH server. This generates three keys: **RSAv1**, **RSAv2**, and **DSAv2**. The system automatically generates the host keys on its first boot, so this only needs to be done if a security breach is suspected and the keys need to be changed.
 - **<key_type>**—Manually set the **host-key** (either private or public, but should be both if changing) of the specified key type; options are **rsa1**, **rsa2**, and **dsa2** and either **private-key** or **public-key**. If the positive form of the **private-key** command is used with **no key** given, the user is prompted for the key. Entries made at this prompt echo the asterisk (*) character, and the user must enter the same string twice.
- **listen**—Configure SSH server interface access restrictions.
 - **enable**—Enable (default) or disable (with **no**) the listen interface-restricted list for SSHD. If enabled and at least one non-DHCP interface is specified in the list, the SSH connections are only accepted on those specified interfaces. When disabled, SSH connections are accepted on any interface.
 - **interface <interface_name>**—Add interfaces to the **listen** list; default is **eth0**. If the interface is also running as a DHCP client, it is as if the interface was not added. If DHCP is later turned off on this interface, it is as if the interface was then added to the listen list. Use **no ssh server listen interface <interface_name>** to delete entries.
- **min-version**—Set the minimum version of the SSH protocol that the server supports. Default is **2**. Use **no ssh server min-version** to reset to default.
- **ports**—Specify on which ports the SSH server listens; default is **22**. Multiple ports can be specified by repeating the **<port>** entry. There must be at least one port specified on the system for the SSH server. Removes any previous ports if not listed in the command.
- **x11-forwarding enable**—Enable or disable (with **no**) x11 forwarding for this SSH server. Default is disabled. X11 forwarding can provide a secure, encrypted link.

```
show ssh server [host-keys]
```

SSH server information, including whether or not it is enabled, and the host key fingerprints. Use the **host-keys** option to list information about the SSH server, including whether or not it is enabled, the host key fingerprints, and the full host keys.

stats

Configure statistics and alarms. To see default threshold values, enter **show stats alarm <alarm_ID>**. Set e-mail notifications for these stats using **email** commands. See [“Configuring Media Flow Controller Log Statistics Thresholds \(CLI\)” on page 202](#) for task details.

```
stats
  alarm <alarm_ID>
    clear
    enable
    falling [clear-threshold <value>] [error-threshold <value>]
    rate-limit
      count {long <count> | medium <count> | short <count>}
      reset
      window {long<duration> | medium<duration> | short<duration>}
    rising [clear-threshold <value>] [error-threshold <value>]
  chd <CHD_ID>
    clear
    compute time [interval <seconds>] [range <seconds>]
    enable
  clear-all
  export {<format> | csv} {<report_name>| cpu_util | memory | paging}
    after <date><time> [before <date><time>]
    before <date><time> [after <date><time>]
    filename <filename>
      after <date><time> [before <date><time>]
      before <date><time> [after <date><time>]
  sample <sample_ID>
    clear
    enable
    interval <number_of_seconds>
```

Notes:

- **alarm <alarm_ID>**—Configure alarms based on sampled or computed statistics. See [“stats alarms.”](#) for the list of supported alarms. Some statistics are of concern when they fall below a certain point, others when they rise above a certain point.
 - **clear**—Clear all state for this alarm. Clearing an alarm resets it to a non-error state, clears the watermarks, and forgets the event history. “Watermark” for rising alarm = max value since reset; for falling alarm = min value since reset.
 - **enable**—Enable this alarm. Use **no stats alarm <alarm_ID> enable** to disable.
 - **falling**—Set alarm for when specified statistic fall too low.
 - **clear-threshold <value>**—This value terminates the alarm.
 - **error-threshold <value>**—This value initiates the alarm.
 - **rate-limit**—Set alarm event rate-limits:

- **count**—Set the alarm event rate-limit maximum counts for the three types of counts (**short, medium, long**) for alarms; defaults are short=5, medium=20, long=50. See [“stats alarm rate-limit count.”](#) for more information.
 - **reset**—Reset the rate-limit counters and time for the specified alarm.
 - **window**—Set the alarm event rate-limit duration windows for the three types of durations (**short, medium, long**) for alarms; defaults are short=3600 (1 hour), medium=86400 (1 day), long=604800 (1 week).
- **rising**—Set alarm for when specified statistic rises too high.
 - **clear-threshold**—This value terminates the alarm.
 - **error-threshold**—This value initiates the alarm.
- **chd <CHD_ID>**—Configure computed historical datapoints (CHDs). See [“stats CHDs.”](#) for the list of supported CHDs.
 - **clear**—Clear all data from this CHD series.
 - **compute time**—Set parameters for when this CHD is computed, and which data points are used in each calculation.
 - **interval**—Specify calculation interval (how often to do a new calculation) in number of seconds.
 - **range**—Specify calculation range (the data points to use) in number of seconds.
 - **enable**—Enable this CHD. Use **no stats chd <CHD_ID> enable** to disable the CHD. See [Table 42 on page 439](#) for which CHDs are enabled by default.
- **clear-all**—Clear data for all samples and CHDs, and status for all alarms.
- **export <format> <report_name>**—Export statistics to file. Currently the only supported value for **<format>** is **csv** (comma-separated value). The dataset to be exported is determined by the **report_name** value. Options for **report_name** are: **memory, paging,** and **cpu_util**. All alarm statistics associated with the specified report are exported. Use **file stats upload** to access the file.
 - **after**—Only include stats collected after the specified time.
 - **before**—Only include stats collected before the specified time.
Either one, both, or neither of the **after** and **before** arguments may be specified. These place boundaries on the timestamps of the instances to be exported. When one of these arguments is specified, two values must follow, one for the date (**yyyy/mm/dd**) and one for the time (**hh:mm:ss**); in 24-hour time. Dash (-) may be used in the **<time>** field as an abbreviation for midnight. The date and time specified are interpreted as local time according to the currently set timezone.
 - **filename <filename>**—If a **filename** is specified, the stats are exported to a file of that name; otherwise a name is chosen automatically and contains the name of the report and the time and date of the export. Any automatically-chosen name is given a **.csv** extension. If the user specifies a name, **.csv** is added if it is not already part of the name. The word **custom** should be a reserved report name (no reports should be named that) to leave room in the command set for later allowing you to specify manually which series to export. If the filename is specified, it must come just after the report name. If the **after** or **before** arguments are specified, they may come in either order relative to each other.
- **sample <sample_ID>**—Configure sampled statistics. See [“Measurement Counters \(stats samples\)” on page 204.](#) for a list of supported samples.

- **clear**—Clear all data from this sample series.
- **enable**—Enable this sample. Use **no stats sample <sample_ID> enable** to disable the sample. See [Table 43 on page 441](#) for which samples are enabled by default.
- **interval**—Set the amount of time in seconds (1 - 2147483647) between sample polling for the specified group of sample data.

```
show stats
  alarm [<alarm_ID>]
  chd [<CHD_ID>]
  cpu
  sample [<sample_ID>]
```

Notes:

- **alarm**—Status of all alarms or the specified alarm, whether or not it is in an error state.
- **chd**—Statistics CHDs settings.
- **cpu**—Basic statistics about CPU utilization: the current level, the peak over the past hour, and the average over the last hour.
- **sample**—Sampling interval for all samples, or the specified one.

stats alarms

Enable or disable the specified alarm with **stats alarm <alarm_ID> enable**; the **no** variant disables the specified alarm. Set a threshold value for each alarm using the **stats alarm <alarm_ID> rising** or **stats alarm <alarm_ID> falling** commands; the alarm is triggered when the specified threshold is reached. Alarms that can be enabled or disabled are described in [Table 41](#).

stats alarm States

Alarms can be in one of two states:

- **OK**—The alarm is in a normal state.
- **ERROR**—The alarm is already triggered and it is in the error state.

You can specify the **error-threshold** and **clear-threshold** levels for alarms using the **stats** commands; for example:

```
stats alarm total_byte_rate rising error-threshold 10
stats alarm total_byte_rate rising clear-threshold 1
```

In this example, after the **total_byte_rate** stat alarm goes beyond **10**, the alarm state changes to ERROR. The state changes to OK only when the **total_byte_rate** stat alarm comes to less than or equal to **1**.

stats alarm rate-limit count

Specify three time limits for tracking and, potentially, limiting how many of this type of alarm to send. The duration and count of each bucket for **short**, **medium**, and **long** are meant to be ordered such that the **short** bucket has the smallest time duration (**window**) and smallest maximum allowed **count**. The **rate-limit** applies to all three buckets simultaneously.

Separate counts of alarms are kept for **error** alarm events and **clear** alarm events. For each alarm type, a single skip count is kept and is reported when the alarm event is later sent.

When an alarm event count is exceeded for any bucket, only the skipped count is incremented (not any of the alarm event counts).

For example, if the **short** bucket allows 5 alarm events per 5 minutes, the sixth alarm and above, that occur during the 5 minute window, only increment the skip count. When one of the windows expires, the alarms can be sent again (assuming no other window and count limits are exceeded) and the tracking starts again.

Table 41 Stats Alarms

| Stats Alarm | Description |
|--|---|
| Alarms | |
| (unless otherwise noted, default rising error threshold is 200000000 Bps; rising clear threshold is 100000000 Bps) | |
| <code>avg_cache_byte_rate *</code> | Total number of bytes served divided by system up time. |
| <code>avg_disk_byte_rate *</code> | Total number of bytes served from all disks divided by system up time. |
| <code>avg_origin_byte_rate *</code> | Total number of bytes fetched from origin divided by system up time. |
| <code>cache_byte_rate *</code> | Total data bandwidth being served from RAM/buffer cache. |
| <code>connection_rate *</code> | Incoming connections per second, arrived by summing up all accepted connections and dividing by system up time. Default rising error threshold is 20000 per sec; default rising clear threshold is 10000 per sec. |
| <code>cpu_util_indiv</code> | Average CPU utilization. The units for the <code>cpu_util_indiv</code> alarm are hundredths of a point of the one-minute load average. For example, setting it to 100 causes an alarm if the one-minute load average is ever over 1.0 when it is sampled. Default rising error threshold is 90% ; default rising clear threshold is 70% . |
| <code>disk_byte_rate *</code> | Current total data bandwidth being served from disk in the system. |
| <code>disk_io *</code> | Disk I/O (input/output) in kilobytes per second. Default rising error threshold is 5120 kilobytes per sec; default rising clear threshold is 4608 kilobytes per sec. |
| <code>fs_mnt</code> | Percent free filesystem space. Default falling error threshold is 7% of disk space free; default falling clear threshold is 10% of disk space free. |
| <code>http_transaction_rate *</code> | Number of HTTP transactions (GET requests) per second; calculated as number of GET requests received so far divided by system up time. Default rising error threshold is 40000 per sec; default rising clear threshold is 20000 per sec. |
| <code>intf_util *</code> | Network utilization (in Bps). Default rising error threshold is 10485760 bytes per sec; default rising clear threshold is 9437184 bytes per sec. |
| <code>memory_pct_used *</code> | Percent of physical memory in current in use. Default rising error threshold is 90% of physical memory used; default rising clear threshold is 87% of physical memory used. |
| <code>nkn_cpu_util_ave</code> | CPU utilization across all cores too high. |
| <code>origin_byte_rate *</code> | Current total data bandwidth being served from origin. |
| * Disabled by default; all others are enabled by default. | |

Table 41 Stats Alarms (Continued)

| Stats Alarm | Description |
|---|--|
| <code>paging</code> | Paging activity (page faults). The units for the paging alarm are number of pages read from, or written to, the swap partition. The alarm is on the amount of paging activity that has occurred over the past 20 seconds. Default rising error threshold is 2000 page faults; default rising clear threshold is 1000 page faults. |
| <code>perdiskbyte_rate *</code> | Total number of bytes served from each disk drive divided by system up time on a per-disk basis; cumulative (not per-disk). |
| <code>peroriginbyte_rate *</code> | Total amount of data fetched per origin server divided by system up time; cumulative (not per-origin). |
| <code>perportbyte_rate *</code> | By-port I/O (input/output) in kilobytes, per second. |
| <code>rp_global_pool_client_sessions</code> | Alarm for Global (default) resource pool. Raised when the <code>global_pool</code> (default resource pool) client session count exceeds a configured threshold limit. An alarm raised may result in an e-mail notification being sent out, if configured. |
| <code>rp_global_pool_max_bandwidth</code> | Alarm for Global (default) resource pool. Raised when the <code>global_pool</code> (default resource pool) bandwidth exceeds a configured threshold limit. An alarm raised may result in an e-mail notification being sent out, if configured. |
| <code>total_byte_rate</code> | Total data bandwidth being served in the system; does not include management traffic on Ethernet port. |
| * Disabled by default; all others are enabled by default. | |

stats CHDs

Stats CHDs are shown in [Table 42](#); unless otherwise noted, default **interval** and **range** for all CHDs are **1 data point**.

Table 42 Stats CHDs

| Stat CHD | Description |
|-----------------------------------|--|
| <code>avg_cache_byte_rate</code> | Total number of bytes served from RAM/buffer cache divided by system up time. |
| <code>avg_disk_byte_rate</code> | Total number of bytes served from all disks divided by system up time. |
| <code>avg_origin_byte_rate</code> | Total number of bytes fetched from origin divided by system up time. |
| <code>bandwidth_day_avg</code> | Amount of data fetched averaged over 24 hours. Default interval is 900 seconds , default range is 900 seconds . |
| <code>bandwidth_day_peak</code> | High point of data fetched averaged over 24 hours. Default interval is 900 seconds , default range is 900 seconds . |
| <code>bandwidth_month_avg</code> | Amount of data fetched averaged over 30 days. Default interval is 14400 seconds , default range is 14400 seconds . |
| <code>bandwidth_month_peak</code> | High point of data fetched averaged over 30 days. Default interval is 14400 seconds , default range is 14400 seconds . |

Table 42 Stats CHDs (Continued)

| Stat CHD | Description |
|--------------------------------------|--|
| <code>bandwidth_week_avg</code> | Amount of data fetched averaged over 7 days. Default interval is 3600 seconds , default range is 3600 seconds . |
| <code>bandwidth_week_peak</code> | High point of data fetched averaged over 7 days. Default interval is 3600 seconds , default range is 3600 seconds . |
| <code>cache_bandwidth_day</code> | Bandwidth served from RAM/buffer cache over last 24 hours. |
| <code>cache_bandwidth_week</code> | Bandwidth served from RAM/buffer cache over last 7 days. |
| <code>cache_byte_rate</code> | Total data bandwidth being served from RAM/buffer cache. |
| <code>connection_day_avg</code> | Incoming connections average for 24 hours. Default interval is 900 seconds , default range is 900 seconds . |
| <code>connection_day_peak</code> | Incoming connections high point in last 24 hours. Default interval is 900 seconds , default range is 900 seconds . |
| <code>connection_month_avg</code> | Incoming connections average for last 30 days. Default interval is 14400 seconds , default range is 14400 seconds . |
| <code>connection_month_peak</code> | Incoming connections high point for last 30 days. Default interval is 14400 seconds , default range is 14400 seconds . |
| <code>connection_rate</code> | Incoming connections per second, arrived by summing up all accepted connections and dividing by system up time. |
| <code>connection_week_avg</code> | Incoming connections average for 7 days. Default interval is 3600 seconds , default range is 3600 seconds . |
| <code>connection_week_peak</code> | Incoming connections high point in 7 days. Default interval is 3600 seconds , default range is 3600 seconds . |
| <code>cpu_util</code> | CPU utilization: percentage of time spent. |
| <code>cpu_util_ave</code> | CPU utilization average: percentage of time spent. Default interval is 15 seconds , default range is 60 seconds . |
| <code>cpu_util_day</code> | CPU utilization that day: percentage of time spent. Default interval and range are 1800 seconds . |
| <code>current_bw_serv_rate_tx</code> | Current transmitted bandwidth. This is a sample variable and holds the instantaneous value of the current transmit throughput. |
| <code>disk_bandwidth_day</code> | Bandwidth being served from disk in last 24 hours. |
| <code>disk_bandwidth_week</code> | Bandwidth being served from disk in last 7 days. |
| <code>disk_byte_rate</code> | Total data bandwidth served from disk. |
| <code>disk_io</code> | Disk I/O (input/output) in kilobytes per second. |
| <code>fs_mnt_day</code> | Filesystem usage average that day: bytes. Default interval and range are 300 seconds . |
| <code>fs_mnt_month</code> | Filesystem usage for that month: bytes. Default interval and range are 7200 seconds . |
| <code>fs_mnt_week</code> | Filesystem usage for that week: bytes. Default interval and range are 1800 seconds . |

Table 42 Stats CHDs (Continued)

| Stat CHD | Description |
|------------------------------------|--|
| <code>http_transaction_rate</code> | Number of HTTP transactions (GET requests) per second; calculated as number of GET requests received so far divided by system up time. |
| <code>intf_day</code> | Network interface statistics aggregation: bytes. Default interval and range are 300 seconds . |
| <code>intf_hour</code> | Network interface statistics by hour. Default interval and range are 30 seconds . |
| <code>intf_month</code> | Network interface statistics by month. |
| <code>intf_util</code> | Aggregate network utilization across all interfaces. |
| <code>memory_day</code> | Average physical memory usage: bytes. Default interval and range are 1800 seconds . |
| <code>memory_pct</code> | Average physical memory usage. |
| <code>origin_bandwidth_day</code> | Bandwidth served from origin in last 24 hours. |
| <code>origin_bandwidth_week</code> | Bandwidth served from origin in last 7 days. |
| <code>origin_byte_rate</code> | Current total data bandwidth being served from origin in system. |
| <code>paging</code> | Average paging activity (page faults). Default interval is 20 seconds , default range is 300 seconds . |
| <code>paging_day</code> | Paging activity (page faults) by day. Default interval and range are 1800 seconds . |
| <code>perdiskbyte_rate</code> | Total number of bytes served from each disk drive divided by system up time on a per-disk basis; cumulative (not per-disk). |
| <code>peroriginbyte_rate</code> | Total amount of data fetched per origin server divided by system up time; cumulative (not per-origin). |
| <code>perportbyte_rate</code> | Total amount of data fetched per port divided by system up time; cumulative (not per-origin). |
| <code>tot_bandwidth_day</code> | Total data bandwidth served in last 24 hours. |
| <code>tot_bandwidth_week</code> | Total data bandwidth served in last 7 days. |
| <code>total_byte_rate</code> | Total amount of data fetched. |

stats samples

The options for stats **sample ID** are shown in [Table 43](#); sampling interval defaults shown in **bold**. All are enabled by default unless otherwise indicated.

Table 43 Stats Samples

| Stat Sample | Description |
|------------------------------------|--|
| <code>cache_byte_count</code> | Bandwidth being served from RAM/buffer cache; default = 10 second . |
| <code>cache_byte_count_day</code> | Bandwidth served from RAM/buffer cache for last 24 hours; default = 5 minutes . |
| <code>cache_byte_count_week</code> | Bandwidth served from RAM/buffer cache for last 7 days; default = 30 minutes . |
| <code>connection_count</code> | Total active connections; default = 10 seconds . |

Table 43 Stats Samples (Continued)

| Stat Sample | Description |
|-------------------------------------|--|
| <code>cpu_util</code> | CPU utilization: milliseconds of time spent; default = 15 seconds . |
| <code>current_bw_serv_tx</code> | Current transmitted bandwidth; default = 1 seconds . |
| <code>disk_byte_count</code> | Bandwidth being served from disk; default = 10 seconds . |
| <code>disk_byte_count_day</code> | Bandwidth being served from disk for the last 24 hours; default = 5 minutes . |
| <code>disk_byte_count_week</code> | Bandwidth being served from disk for the last 7 days; default = 30 minutes . |
| <code>disk_io</code> | Disk I/O (input/output, in kilobytes); default = 15 seconds . |
| <code>fs_mnt_bytes</code> | Filesystem usage, in bytes; default = 1 minute . |
| <code>fs_mnt_inodes</code> | Filesystem usage, in inodes; default = 1 minute . |
| <code>http_transaction_count</code> | Number of HTTP transactions (GET requests) per second; default = 10 seconds . |
| <code>interface</code> | Network interface statistics; default = 30 seconds . |
| <code>intf_day</code> | Network interface statistics for the last 24 hours; default = 5 minutes . |
| <code>intf_month</code> | Network interface statistics for the last 30 days; default = 4 hours . |
| <code>intf_util</code> | Network interface utilization, in bytes; default = 5 seconds . |
| <code>intf_week</code> | Network interface statistics for the last 7 days; default = 30 minutes . |
| <code>memory</code> | System memory utilization, in bytes; default = 20 seconds . |
| <code>ns_served_bytes</code> | Number of bytes transmitted by a namespace; default = 5 minutes . |
| <code>ns_transactions</code> | Number of transactions handled by a namespace; default = 5 minutes . |
| <code>num_of_connections</code> | Current number of HTTP connections; default = 10 seconds . |
| <code>origin_byte_count</code> | Bandwidth being served from origin; default = 10 seconds . |
| <code>origin_byte_count_day</code> | Bandwidth served from origin for the last 24 hours; default = 5 minutes . |
| <code>origin_byte_count_week</code> | Bandwidth served from origin for the last 7 days; default = 30 minutes . |
| <code>paging</code> | Paging activity: page faults; default = 20 seconds . |
| <code>perdiskbytes</code> | Number of bytes being served from each disk drive; default = 10 seconds . |
| <code>peroriginbytes</code> | Number of bytes being served from origin; default = 10 seconds . |
| <code>perportbytes</code> | By-port I/O (input/output), in kilobytes per second; default = 10 seconds . |
| <code>proc_mem</code> | Memory used by Media Flow Controller processes; default = 30 seconds . Disabled by default. |
| <code>resource_pool</code> | Resource pool; default = 10 seconds . |
| <code>total_bytes</code> | Total data bandwidth being served in the system; default = 10 seconds . |
| <code>total_bytes_day</code> | Total data bandwidth served in the last 24 hours; default = 5 minutes . |
| <code>total_bytes_week</code> | Total data bandwidth served in the last 7 days; default = 30 minutes . |
| <code>total_cache_byte_count</code> | Total data bandwidth being served from cache; default = 10 seconds . |
| <code>total_disk_byte_count</code> | Total data bandwidth being served from disk; default = 10 seconds . |

Table 43 Stats Samples (Continued)

| Stat Sample | Description |
|-------------------------|--|
| total_origin_byte_count | Total data bandwidth being served from origin; default = 10 seconds . |
| virt-cpu | Virtual CPU utilization: milliseconds of time spent. |

streamlog

Configure the Media Flow Controller RTP/RTSP streaming events log. See [“Configuring Media Flow Controller Service Logs \(CLI\)” on page 188](#) for task details, including information on log rotation and **format** options. See [“Reading the Stream Log \(streamlog\)” on page 194](#) for usage information.

```
streamlog
  copy <SCP>
  filename <filename>
  format <field1 field2 ...>
  on-the-hour {disable | enable}
  rotate {filesize <integer> | time-interval <integer>}
  syslog replicate {disable | enable}
```

Notes:

- **copy**—Auto-upload (when the set **rotate** criteria is reached) the stream log using secure channel protocol (SCP), to the server specified using **hostname**. If **username** and **password** are provided, Media Flow Controller uses that for authentication of the SCP session. The **no** variant disallows auto-upload. See [“Terminology” on page 31](#) for the **scp** URL format; you must have an SCP server installed in order to send files to your machine.
- **filename**—Configure the name of the file where the streaming log is stored. Default is **streamlog.<num>.yyyymmdd_hour:min:sec** (numbered sequentially).
- **format**—Specify a format for the stream log. Use **no** to reset to default: **“%h %c %t %x “%r” %s %l %O**. See [“Stream Log Format Options” on page 185](#), for field option details.
 - **<field1 field2 ...>**—Choose available field options, described in [“Stream Log Format Options” on page 185](#).
 - **clf**—Common Log Format: **%h %V %u %t %r %s %b**
 - **display**—Either **enable** (default) or **disable** the display of the format in the log.
- **on-the-hour**—Set hourly log rotation. Default is **no** (disabled).
- **rotate**—Media Flow Controller allows streaming log rotation based on file size or time.
 - **filesize**—Set rotation based on file size. Media Flow Controller creates ten streamlogs, numbered sequentially, after which it wraps around. By default, **rotate filesize** is **100 MB**. We highly recommend not increasing the size; huge file transfers take a lot of time, and if there is a system reset, large volumes of data are at risk.
 - **time-interval**—Set rotation based on time. Specify a time in minutes after which the streamlog is rotated. Default is **0 hours** (disabled).
- **syslog replicate**—Specify whether (**enable**) or not streamlog messages are seen as part of syslog; default is **disable** (streaming log is not seen as part of syslog).

```
show streamlog [continuous | last]
```

List log settings; or use **last** to see the last few lines of the log and **continuous** to view the log as it is written.

tacacs-server

Terminal Access Controller Access-Control System Plus (TACACS+) is a protocol that provides access control for routers, network access servers and other networked computing devices via one or more centralized servers. TACACS+ provides separate authentication, authorization and accounting services. TACACS+ servers are tried in the order they are configured.

```
tacacs-server
  host <IP_address>
    auth-port <port>
    auth-type {ascii | pap}
    key <string>
    prompt-key
    retransmit <retries>
    timeout <seconds>
  key [<key_string>]
  retransmit <retries>
  timeout <seconds>
```

Notes:

- **host <IP_address>**—Add a TACACS+ server to the set of servers used for authentication. Some of the arguments given may override the configured global defaults for all TACACS+ servers. Use **no tacacs-server host <IP_address>** to delete all TACACS+ servers with the specified IP address. To refine which host is deleted, **no tacacs-server host <IP_address> auth-port <port>** may be specified.
 - **auth-port**—For this host, sets or clears (with **no**) the port for TACACS+. The same IP address can be used in more than one **tacacs-server host** command as long as the **auth-port** is different for each. A UDP port number, **auth-port** must be specified immediately after the host option (if present). Default is **49**.
 - **auth-type**—For this host, specify which of the two currently supported authentication methods (**ascii** or **pap**) to use. Default is **pap**.
 - **key**—For this host, set, or clear (with **no**), the shared secret text string used to communicate with any TACACS+ server. If unspecified, the user is prompted for it.
 - **prompt-key**—Mutually exclusive with **key <string>**. It requests to be prompted for the key, with the entry echoed as asterisk (*) characters, for greater security.
 - **retransmit**—For this host, set or reset to **0** (zero) (with **no**), the number of times the client attempts to authenticate with any TACACS+ server. Range is **0-5**, default is **1**. Set to **0** to disable retransmissions.
 - **timeout**—For this host, set or reset to default (with **no**), the wait time for retransmitting a request to any TACACS+ server. Range is **1-60**, default is **3**.
- **key**—Sets, or clears (with **no**), a global communication value for all TACACS+ servers. Can be overridden in a **tacacs-server host** command. Sets the shared secret text string used to communicate with any TACACS+ server. If the positive form of the private key

command is used with no key, the user is prompted for the key. Entries made at this prompt echo the asterisk (*) character, and the user must enter the same string twice.

- **retransmit**—Sets, or resets to 0 (zero) (with **no**), a global communication value for all TACACS+ servers. Can be overridden in a **tacacs-server host** command. Range is **0-5**, default is **1**. Sets the number of times the client attempts to authenticate with any TACACS+ server. To disable retransmissions set it to 0 (zero).
- **timeout**—Sets, or resets to the default (with **no**), a global communication value for all TACACS+ servers. Can be overridden in a **tacacs-server host** command. Range is **1-60**, default is **3**. Sets the wait time for retransmitting a request to any TACACS+ server.

show tacacs

TACACS+ settings.

tcpdump

tcpdump [<options>]

Network diagnostic tool. Invokes standard binary, passing command line arguments straight through. Runs in foreground, printing packets as they arrive, until you press Ctrl+C.

tech-support

EXEC command. Use this command to collect system information.

tech-support {<URL> | <SCP>}

Upload the tech-support file to the specified URL. Only FTP and TFTP URLs, as well as SCP pseudo-URLs are supported for the destination. See [“Terminology” on page 31](#) for the **scp** URL format and requirements.

The file is created with the name “nkn_tech-support”. We highly recommend that, if you have multiple Media Flow Controllers, each Media Flow Controller be configured with a different path variable (use a different directory) so that **tech-support** data from one Media Flow Controller does not overwrite data from another. If different hosts are used for each Media Flow Controller, then the same path name can be used.

telnet

EXEC command. Use TELNET; or enable or disable, and configure telnet settings.

telnet [<telnet_client_options>]

Invokes the telnet client. The user is returned to the CLI when telnet finishes.

telnet-server

Manage the TELNET server.

telnet-server enable

Enable or disable (with **no**) the telnet server.

```
show telnet-server
```

Telnet server settings.

terminal

EXEC command. Set parameters for the CLI terminal: terminal length, type, and width.

```
terminal [length <integer>] [resize] [type <type>] [width <width>]
```

See [“CLI Options” on page 84](#), `cli session terminal` arguments notes for details.

tracelog

Configure trace log options. See [“Configuring Media Flow Controller Service Logs \(CLI\)” on page 188](#) for task details, including information on log rotation. See [“Reading the Trace Log \(tracelog\)” on page 195](#) for usage information.

```
tracelog
  copy <SCP>
  filename <name>
  on-the-hour {disable | enable}
  rotate {filesize <integer> | time-interval <integer>}
  syslog replicate {disable | enable}
```

Notes:

- **copy**—Auto-upload (when the set **filesize** is reached) the tracelog using secure channel protocol (SCP), to the server specified using **hostname**. If **username** and **password** are provided, Media Flow Controller uses that for authentication of the SCP session. The **no** variant disallows auto-upload. See [“Terminology” on page 31](#) for the **scp** URL format. You must have an SCP server installed in order to send files to your machine.
- **filename**—Configure the name of the file where the trace log is stored. Default is **tracelog.<num>.yyyymmdd_hour:min:sec** (numbered sequentially).
- **on-the-hour**—Set hourly log rotation. Default is **no** (disabled).
- **rotate**—Media Flow Controller allows trace log rotation based on file size or time.
 - **filesize**—Set rotation based on file size. Media Flow Controller creates "trace.log.1," "trace.log.2," and so on all the way to "trace.log.10," after which it wraps around. By default, **rotate filesize** is **100** MB. We highly recommend not increasing the size; huge file transfers take a lot of time, and if there is a system reset, large volumes of data are at risk.
 - **time-interval**—Set rotation based on time. Specify a time in minutes after which the trace log is rotated.
- **syslog replicate**—Specify whether (**enable**) or not trace log messages are seen as part of syslog; default is **disable** (trace log is not seen as part of syslog).

```
show tracelog [continuous | last]
```

List log settings; or use **last** to see the last few lines of the log and **continuous** to view the log as it is written.

traceroute

EXEC command. Network diagnostic tool **traceroute**. Invokes standard binary, passing command line arguments straight through.

```
traceroute [<options>] {<hostname>}
```

upload

Upload files.

```
upload
```

```
accesslog {current | all} {<SCP> | <SFTP>}
cachelog {current | all} {<URL> | <SCP>}
errorlog {current | all} {<URL> | <SCP>}
fmsaccesslog all {<URL> | <SCP>}
fmsedgelog all {<URL> | <SCP>}
fuselog {current | all} {<URL> | <SCP>}
object list <namespace> {<URL> | <SCP>}
streamlog {current | all} {<URL> | <SCP>}
tracelog {current | all} {<URL> | <SCP>}
```

Upload either the current or all **accesslog**, **cachelog**, **errorlog**, **streamlog**, delivery **tracelog**, or **namespace <name> object list** files, to the specified SCP path. With the accesslog, you can also use SFTP. With the errorlog, you can also use an HTTP URL. See [“Terminology” on page 31](#) for the **scp** and **sftp** URL format and requirements. To set up SSH, used by SFTP, see [“Using SSH in Automated Scripts \(CLI\)” on page 85](#).

username

Configure user accounts and privileges. See [“Understanding Authentication, Authorization, and User Options” on page 95](#) for task details.

```
username <username>
  capability <capability>
  disable [password]
  full-name <name>
  nopassword
  password [ 0 <cleartext_password> | 7 <encrypted_password> |
    <cleartext_password>]
```

Add or delete (with **no**) a user account. New users are created initially with **admin** privileges and you must manually change those privileges, if desired. To enable a user account, just set a password on it (or use the **nopassword** command to enable it with no password required for login). Removing a user account does not terminate any current sessions that user has open; it just prevents new sessions from being established.

Notes:

- **capability <capability>**—Change the capabilities for this user account. Creates the account if it does not exist. Use **no** to revert the specified user to the default capability, which is **admin** privileges. There are three pre-defined capabilities:

- **admin**—Full privileges (default); in **Enable** mode all **EXEC** commands are available. If no password is set, can log in as **admin**.
- **monitor**—Privileges for reading configuration data (not logs) and performing actions, but not for changing any configuration. If no password is set, can log in as **monitor**.
- **unpriv**—Unprivileged.
- **ftpruser**—Privileged for FTP transactions only; FTP users auto-created with a **namespace** creation have this capability. See **namespace pre-stage** for details.
- **disable**—Disable means of logging in to this account. Disabling a user account does not terminate any current sessions that user has open; it just prevents new sessions from being established. Optionally, specify a **password**; this leaves the account as a whole the same, but forbids login with a password; it is assumed that SSH key access is used instead. To re-enable the account, the user must un-disable it, and put a password on it. The **no username <userid> disable** command prints a message to this effect; it only exists to avoid stumping users with an apparently irreversible command.
- **full-name**—Set or reset to empty string (with **no**) the full name (referred to in some circles as the "gecos") on this account.
- **nopassword**—Allow login to this account without a password.
- **password**—Set the login password for this user. Enter * (asterisk) to disable login; enter nothing (and confirm) to set no password for a user.
 - **0 <cleartext_password>**—Allows the password to be specified in cleartext, whereby the system encrypts it using the DES algorithm. This password shows in the encrypted form with **show configuration**.
 - **7 <encrypted_password>**—Allows the password to be provided in the same encrypted form in which it is stored in the shadow password file (/etc/shadow). Useful for **show configuration**, since the cleartext password cannot be recovered (in cleartext) after it is set.
 - **<cleartext_password>**—Enter a cleartext password; if none is specified, the user is prompted for the password, with entries obscured, requiring the same string to be entered twice for confirmation.

Tip! Use **password 7 <encrypted_password>** when setting user passwords; in case you ever need to re-apply a saved configuration, the encrypted user passwords are saved (not so for cleartext passwords) and can be re-applied with the saved configuration.

```
show
  usernames
  users [history [username <username>]]
  whoami
```

Notes:

- **usernames**—List of all user accounts and the capabilities of each.
- **users**—List of all currently logged-in users, and related information such as idle time and what host they have connected from. Optionally, choose **history** to view the history of user logins, past and present. You can also optionally specify a **username** and only the history of that particular user is displayed.
- **whoami**—The username and capabilities of the currently logged-in user.

virtual-player

Create a named virtual player in Media Flow Controller with policies for delivery. The virtual player can then be used in a namespace; see [namespace](#) for details. The different virtual player types correspond to those types of videos; for example, if you intend to deliver YouTube® videos, create a **type youtube** virtual player for the corresponding namespace.

When you create a virtual player of any type with **virtual-player <name> type <type>**, you enter **virtual-player** prefix mode. This makes configuration easier if you have a set of command values to enter with copy-and-paste. To leave the **virtual-player** prefix mode, use **exit**. Use **no** to negate or disable settings.

These are the general **virtual-player** commands; not all options are available for all types:

```
virtual-player <name>
  assured-flow {auto | query-string-parm <string> | rate <kbps>}
  cache-name video-id query-string-parm <string> format-tag query-string-
    parm <string>
  connection max-bandwidth <kbps>
  control-point {player | server}
  fast-start {query-string-parm <string> | size <size-kB> | time <seconds>}
  fragment-tag <tag>}
  full-download
    always
    match <string> {header <name> | query-string-parm <string>}
  hash-verify
    digest <digest_type>
    shared-secret <string> {append | prefix}
    match query-string-parm <string>
    expiry-time-verify query-string-parm <string>
    url-type {absolute-url | relative-url | object-name}
  health-probe query-string-parm <string> match <string>
  quality-tag <string>
  rate-map match <string> rate <kbps>
  req-auth digest md-5 stream-id query-string-parm <string> auth-id query-
    string-parm <string> shared-secret <string> time-interval <seconds>
    match query-string-parm <string>
  seek-config query-string-parm <string>
    enable-tunnel
    seek-length query-string-parm <string>
  seg-frag-delimiter <string> [enable-tunnel] [seek-length query-string-
    parm <string>]
  signals
    session-id query-string-parm <string> state query-string-parm <string>
    profile query-string-parm <string>
  type [generic | break | qss-streamlet | yahoo | smoothflow | youtube |
    smoothstream-pub | flashstream-pub]
```



NOTE: If no virtual player is assigned to a namespace, that namespace uses the **network connection** values for **assured-flow** and **connection max-bandwidth**. If a virtual player is assigned to a namespace, the virtual player values override the network connection values.

See [“Media Flow Controller Virtual Players” on page 51](#) for an overview. See [Chapter 5, “Configuring Virtual Players, Media Fetch and Pre-Staging \(CLI\)”](#) for task details, including details on using various virtual player parameters.

Currently Media Flow Controller supports six types of virtual players:

- `virtual-player type generic`—For caching most Web video content. Delivery options include **assured-flow**, **connection max-bandwidth**, **fast-start**, **full-download**, **hash verification**, and **seek**.
- `virtual-player type break`—For Break® video delivery (no **full-download** option).
- `virtual-player type qss-streamlet`—Fine-grained list of delivery rate-maps for assured flow.
- `virtual-player type yahoo`—For Yahoo video delivery, includes hash digests and healthcheck probes.
- `virtual-player type smoothflow`—For SmoothFlow function.
- `virtual-player type youtube`—For YouTube video delivery.
- `virtual-player type smoothstream-pub`—For Smoothstream videos.

```
show virtual-player {list | <name>}
```

List the configured virtual players in the system or details of a specified virtual player.



NOTE: In Release 2.0.7, the show options command ? (question mark), lists all virtual player options no matter what **virtual player type** you are configuring; however, if you try to set an option that does not apply to that player type, an error is displayed.

NOTE: The virtual player **query-string-parm** values you configure in your Media Flow Controller origin must match the corresponding **query-string-parm** values configured in your Media Flow Controller edge.

virtual-player type generic

Use **type generic** (formerly type 0) virtual players for caching most Web video content. Supports seek/scrub for MP4 and FLV videos using time offsets, fast-start, assured flow, connection max-bandwidth, full-download, and authentication via hash. Use **no virtual-player <name>** to delete. See [Chapter 5, “Configuring Virtual Players, Media Fetch and Pre-Staging \(CLI\)”](#) for implementation details.

```
virtual-player <name> type generic
  assured-flow {auto | query-string-parm <string> | rate <kbps>}
  connection max-bandwidth <kbps>
  fast-start {query-string-parm <string> | size <KB>}
  full-download {always | match <string> {query-string-parm <string> |
    header <header_name>}}
  hash-verify
    digest <digest_type>
    shared-secret <string> {append | prefix}
    match query-string-parm <string>
    expiry-time-verify query-string-parm <string>
    url-type {absolute-url | relative-url | object-name}
```

```
seek-config query-string-parm <string>
enable-tunnel
seek-length query-string-parm <string>
```

Create a **virtual-player <name> type generic**. Notes:

- **assured-flow**—Set AFR for this virtual player. This assures that content is delivered at least at the specified rate (but no more than the configured **connection max-bandwidth**) for the session. In prefix mode, use **no assured-flow** to disable again, if needed. By default, AFR is disabled (no delivery rate is assured). See [“Using Network Connection Assured Flow” on page 100](#) for examples.
 - **auto**—Not supported in Release 2.0.7.
 - **query-string-parm**—Specify a string; referenced value must be in kilobytes.
 - **rate**—Define a static value, in kbps. A value of **0** (zero) means no throughput at all.
- **connection max-bandwidth**—Set the maximum bandwidth for a session. The actual session bandwidth is between the AFR and this value. Even if there is available bandwidth in the link, Media Flow Controller does not allocate more than this value for a session. When it is a full download, Media Flow Controller tries to allocate the max-bandwidth to the session. Default is **0** (unbounded) with the Media Flow Controller license, **200** kbps without it; you must have the license to change the unlicensed default. In prefix mode, use **no connection** to reset the default.
- **fast-start**—Deliver the first set of kilobytes at either the maximum session speed or the available bandwidth; in prefix mode, use **no fast-start** to disable (default is disabled).
 - **query-string-parm**—Specify a string; referenced value must be in kilobytes.
 - **size**—Define how many kilobytes should be expedited.
- **full-download**—Allow the delivery to download content at the fastest possible speed, limited by the set **connection max-bandwidth** and possibly exceeding the set **assured-flow rate**.
 - **always**—Downloads are always delivered at the fastest possible speed.
 - **match**—Downloads are only delivered at the fastest possible speed when a **match** is found for the specified **query-string-parm** or **header** name.
- **hash-verify**—Verify the hash value specified in the URL (see [“Using hash-verify” on page 127](#) for details); in prefix mode, use **no hash-verify** to disable.
 - **digest**—Set the type of hash digest to use. Only **md-5** is supported in Release 2.0.7.
 - **expiry-time-verify query-string-parm**—Reject incoming requests that have passed the current system time; coupled with hash verification, this helps prevent bandwidth stealing. The value for **query-string-parm** must be an expiration time specified as a standard POSIX timestamp (seconds since January 1 1970 00:00:00 UTC). The timestamp value in the URL is generated by the player issuing the request. Media Flow Controller compares this timestamp with the current time to determine if the URL request has expired. An expiry time of **0** (zero) (for example, in cases where the player does not give a timestamp) indicates that this request “does not expire.” Sample URL request form where “e” indicates the **expiry-time-verify query-string-parm** value:

```
http://www.example.com/media/
foo.flv?e=3312665958&h=ec41f550878f45d9724776761d6ac416
```

- **match query-string-parm**—Specify a string indicating the provided hash value; the default for this virtual-player type is **h**.
- **shared-secret {append | prefix}**—Enter a secret key that is then appended or prefixed (as specified) to the URI to calculate the hash which is then "matched" with the **match query-string-parm** hash value.
- **url-type**—Tell Media Flow Controller what part of the request URL to use for the hash calculation:
 - **absolute-url**—Current and default behavior. The hash calculation should use the entire request URL (including query string up to configured **match query-string-parm** value).
 - **relative-uri**—The hash calculation should use only the URI part of the request URL, excluding the domain, and access method (but including query string up to configured **match query-string-parm** value).
 - **object-name**—The hash calculation should use only the object name part of the request URL (and query string up to configured **match query-string-parm** value).
- **seek-config query-string-parm <string>**—Specify a string to implement MP4 or FLV seek (allows the client player to seek to a specific location of the URL). In prefix mode, use **no seek** to disable. Optionally:
 - **enable-tunnel**—If enabled, all **seek** requests to the origin server are tunneled; typically this option needs to be enabled only when the origin site changes their **seek** mechanism. Default is disabled.
 - **seek-length query-string-parm**—Specify a string to signal the number of bytes of data to send from the **seek** start position; referenced value must be in bytes. For FLV support of this option, **seek-flv-type** must be set to **byte-offset**.

virtual-player type break

Use **type break** (formerly type 1) virtual players for Break® video delivery. Supports seek/scrub for FLV videos using byte offsets and MP4 videos using time-offsets, fast-start, assured flow, connection max-bandwidth, and authentication via hash. Use **no virtual-player <name>** to delete. See [“Creating and Configuring Virtual Players \(CLI\)” on page 123](#) for implementation details.

```
virtual-player <name> type break
  assured-flow {auto | query-string-parm <string> | rate <kbps>}
  connection max-bandwidth <kbps>
  fast-start {query-string-parm <string> | size <KB> | time <integer>}
  hash-verify digest <digest_type> shared-secret <string>{append | prefix}
    match query-string-parm <string>
  seek query-string-parm <string> [enable-tunnel] [seek-length query-
    string-parm <string>]
```

Notes:

- **assured-flow**—Set AFR for this virtual player, this assures that content is delivered at least at the specified rate (but no more than the configured **connection max-bandwidth**) for the session. Use **virtual player <name> type break no assured-flow** to re-disable, if needed. By default, AFR is disabled (no delivery rate is assured). See [“Using Network Connection Assured Flow” on page 100](#) for example.

- **auto**—Not supported in Release 2.0.7.
- **query-string-parm**—Specify a string; referenced value must be in kilobytes.
- **rate**—Define a static value, in kbps. A value of **0** (zero) means no throughput at all.
- **connection max-bandwidth**—Set the maximum bandwidth for a session. The actual session bandwidth is between the AFR (Assured Flow Rate) and this value. Even if there is available bandwidth in the link, Media Flow Controller does not allocate more than this value for a session. When it is a full download, Media Flow Controller tries to allocate the max-bandwidth to the session. Default is **0** (unbounded) with the Media Flow Controller license, **200** kbps without it; you must have the license to change the unlicensed default. In prefix mode, use **no connection** to reset default.
- **fast-start**—Deliver the 1st set of kilobytes at either the maximum session speed or the available bandwidth; in prefix mode, use **no fast-start** to disable (default is disabled).
 - **query-string-parm**—Specify a string; referenced value must be in kilobytes.
 - **size**—Define how many kilobytes should be expedited.
 - **time**—Define how many seconds should be expedited.
- **hash-verify**—Verify the hash value specified in the URL (see [“Using hash-verify” on page 127](#) for details); in prefix mode, use **no hash-verify** to disable.
 - **digest**—Only **md-5** is supported in Release 2.0.7.
 - **match query-string-parm**—Specify a string indicating the provided hash value.
 - **shared-secret {append | prefix}**—Enter a secret key that is then appended or prefixed (as specified) to the URI, to calculate the hash which is then "matched" with the corresponding configured **match query-string-parm** hash value.
- **seek**—Specify a string to implement FLV seek (allows the client player to seek to a specific location of the URL); default is **ec_seek**. Use **no seek** to disable. Optionally:
 - **seek-length query-string-parm**—Specify a string to signal the number of bytes of data to send from the **seek** start position; referenced value must be in bytes.
 - **enable-tunnel**—If set, all **seek** requests to the origin server are tunneled; typically this option needs to be selected only when the origin site changes their seek mechanism. Default is disabled.

virtual-player type qss-streamlet

Use **type qss-streamlet** (formerly type 2) virtual players to enforce **assured-flow** via rate maps. Useful for Adaptive Bit Rate video delivery where the URI format signals the assured flow rate. Use **no virtual-player <name>** to delete. In prefix mode, use **no rate-map match <string>** to delete one particular entry. See [Chapter 5, “Configuring Virtual Players, Media Fetch and Pre-Staging \(CLI\)”](#) and [“Using virtual-player type qss-streamlet rate-map” on page 128](#) for implementation details.

```
virtual-player <name> type qss-streamlet
  connection max-bandwidth <kbps>
  rate-map match <string> rate <kbps>
```

Notes:

- **connection max-bandwidth**—Set the maximum bandwidth for a session. The actual session bandwidth is between the AFR for a given URL as determined by its configured

rate-map **rate**, and this value. Even if there is available bandwidth in the link, Media Flow Controller does not allocate more than this value for a session. Default is **0** (unbounded) with the Media Flow Controller license, **200** kbps without it; you must have the license to change the unlicensed default. Use **no connection** to reset default.

- **rate-map**—The configured rate-map **rate** extracts the value from the URL to calculate the AFR for each HTTP request. By default, the **match** string (length 2 bytes; for example, **01**) is extracted by going to the end of the URL and skipping 12 bytes from the end. The value in that location is mapped to the configured **rate**, in kbps. For more details, see [“Using virtual-player type gss-streamlet rate-map” on page 128](#).

virtual-player type yahoo

Use **type yahoo** (formerly type 3) virtual players to allow caching and authentication of Yahoo® videos. Supports special hash digests, healthcheck probes, and seek and AFR. Use **no virtual-player <name>** to delete. See [Chapter 5, “Configuring Virtual Players, Media Fetch and Pre-Staging \(CLI\)”](#) for task details.

This type of virtual player requires a special license to create; see Juniper Networks Support for details.

```
virtual-player <name> type yahoo
  assured-flow {auto | query-string-parm <string> | rate <kbps>}
  connection max-bandwidth <kbps>
  health-probe query-string-parm <string> match <string>
  req-auth digest md-5 stream-id query-string-parm <string> auth-id query-
    string-parm <string> shared-secret <string> time-interval <seconds>
    match query-string-parm <string>
  seek-config query-string-parm <string>
  seek-flv-type {byte-offset | time-secs | time-msec}
  seek-length query-string-parm <string>
  seek-mp4-type {time-secs | times-msec}
  tunnel-mode {enable | disable}
```

Notes:

- **assured-flow**—Set AFR for this virtual player, this assures that content is delivered at least at the specified rate (but no more than the configured **connection max-bandwidth**) for the session. In prefix mode, use **no assured-flow** to disable again, if needed. By default, AFR is disabled (no delivery rate is assured). See [“Using Network Connection Assured Flow” on page 100](#) for an example.
 - **auto**—Not supported in Release 2.0.7.
 - **query-string-parm**—Specify a string; referenced value must be in kilobytes.
 - **rate**—Define a static value, in kbps. A value of **0** (zero) means no throughput at all.
- **connection max-bandwidth**—Set the maximum bandwidth for a session. The actual session bandwidth is between the AFR and this value. Even if there is available bandwidth in the link, Media Flow Controller does not allocate more than this value for a session. When it is a full download, Media Flow Controller tries to allocate the max-bandwidth to the session. Default is **0** (unbounded) with the Media Flow Controller license, **200** kbps without it; you must have the license to change the unlicensed default. Use **no connection** to reset default.

- **health-probe**—Configure an external server to do health checks by making Media Flow Controller fetch data from origin and play it to the server initiating the health check. The signal that a given HTTP request is for a health probe is the configured **health-probe query-string-param <string>**; if that value matches the following **match <string>** value, the GET request is treated as a health probe. When servicing health probes, Media Flow Controller does not cache the data into disk or buffer. Use **no health-probe** to disable.
- **req-auth**—Compute MD-5 hash of query string parameters representing **stream-id** (default is **streamid**), **auth-id** (default is **authid**), a configured **shared-secret** (default is **ysecret**), and **time-interval** (default is **15** seconds); and match the computed value with the specified **match query-string-param <string>** (default is **ticket**). All arguments must be configured. The session proceeds if the computed MD-5 hash matches; if there is no match, the session is rejected. In prefix mode, use **no req-auth** to disable.
- **seek-config query-string-param <string>**—Specify a string to implement MP4 or FLV seek (allows the client player to seek to a specific location of the URL). In prefix mode, use **no seek** to disable. Optionally:
 - **seek-flv-type**—Configure the FLV seek type:
 - **byte-offset**—The value of the seek **query-string-param** sent by the client will be in bytes. This option must be set for **seek-length** for FLV.
 - **time-secs**—The value of the seek **query-string-param** sent by the client will be in seconds.
 - **time-msec**—The value of the seek **query-string-param** sent by the client will be in milliseconds.
 - **seek-length query-string-param**—Specify a string to signal the number of bytes of data to send from the **seek** start position; referenced value must be in bytes. For FLV support of this option, **seek-flv-type** must be set to **byte-offset**.
 - **seek-mp4-type**—Configure the MP4 seek type:
 - **time-secs**—The value of the seek **query-string-param** sent by the client will be in seconds.
 - **time-msec**—The value of the seek **query-string-param** sent by the client will be in milliseconds. This is the recommended setting for YouTube players.
 - **tunnel-mode {disable | enable}**—If enabled, all **seek** requests to the origin server are tunneled; typically this option needs to be enabled only when the origin site changes their **seek** mechanism. Default is disabled.

virtual-player type smoothflow

Use **type smoothflow** (formerly type 4) virtual players to enable Adaptive Bit Rate delivery of video to Adobe Flash players; set only those parameters you need. Use **no virtual-player <name>** to delete. See [“Creating and Configuring Virtual Players \(CLI\)” on page 123](#) for implementation details.



NOTE: Video content must be prepared in a manner described in [Chapter 13, “Deploying SmoothFlow for Media Flow Controller”](#) to enable delivery using this player.

```

virtual-player <name> type smoothflow
  connection max-bandwidth <kbps>
  control-point <string>
  signals session-id query-string-param <string> state query-string-param
    <string> profile query-string-param <string>
  hash-verify digest <digest_type> shared-secret <string>{append | prefix}
    match query-string-param <string>
  seek query-string-param <string> [enable-tunnel] [seek-length query-
    string-param <string>]

```

Notes:

- **connection max-bandwidth**—Set the maximum bandwidth for a session. The actual session bandwidth is between the AFR (Assured Flow Rate) and this value. Even if there is available bandwidth in the link, Media Flow Controller does not allocate more than this value for a session. When it is a full download, Media Flow Controller tries to allocate the max-bandwidth to the session. Default is **0** (unbounded) with the Media Flow Controller license, **200** kbps without it; you must have the license to change the unlicensed default. In prefix mode, use **no connection** to reset the default.
- **control-point**—Specify either **server** or **player** (default) for SmoothFlow signaling. If **player**, the player at the client side explicitly signals the bandwidth changes and Media Flow Controller adjusts the bit-rate of the video accordingly. If **server**, Media Flow Controller detects the bandwidth variations at the client side and adjusts the bit-rate of the video accordingly.
- **signals**—Set triggers for delivery functions (required configuration). Your client player must understand the query params you configure here; defaults are **sid** (for session-id), **sf** (for state), and **pf** (for profile).
 - **session-id query-string-param <string>**—Specify a query param name to signal the session ID; default is **sid**.
 - **state query-string-param <string>**—Specify a query param to signal the SF state; default is **sf**. The allowed values (sent by the client player) are **0**, **1**, **2**, **3**, and **4** and these denote: **0**=disable SmoothFlow; **1**=start a SmoothFlow session; **2**=client player request for client Asset Index file; **3**=client player request for profile adaptation; **4**=initiate SmoothFlow processing for content (chunks and names video files, creates the Asset Index file if needed, and queues content).
 - **profile query-string-param <string>**—Specify a query param name to set the media bit-rate profile; default is **pf**.
- **hash-verify**—Verify the hash value specified in the URL (see [“Using hash-verify” on page 127](#) for details); in prefix mode, use **no hash-verify** to disable.
 - **digest**—Only **md-5** is supported in Release 2.0.7.
 - **match query-string-param**—Specify a string indicating the provided hash value; default for this virtual-player type is **h**.
 - **shared-secret {append | prefix}**—Enter a secret key that is then appended or prefixed (as specified) to the URI, to calculate the hash which is then "matched" with the **match query-string-param** hash value.
- **seek**—Specify a string to implement FLV seek (allows the client player to seek to a specific location of the URL). In prefix mode, use **no seek** to disable. Optionally:
 - **seek-length query-string-param**—Specify a name to signal the number of bytes of data to send from the **seek** start position; referenced value must be in bytes.

- **enable-tunnel**—If set, all **seek** requests to the origin server are tunneled; typically this option needs to be selected only when the origin site changes their seek mechanism. Default is disabled.

virtual-player type youtube

Use **type youtube** (formerly type 5) virtual players to allow caching, and support seek, for YouTube FLV and MP4 media using time offsets. Use **no virtual-player <name>** to delete. See [Chapter 5, “Configuring Virtual Players, Media Fetch and Pre-Staging \(CLI\)”](#) and [“Using Virtual Player Type YouTube” on page 131](#) for implementation details. For transparent proxy, see [“Transparent Proxy Deployments” on page 66](#).

```
virtual-player <name> type youtube
  assured-flow {auto | query-string-param <string> | rate <kbps>}
  cache-name video-id query-string-param <string> format-tag query-string-
    param <string>
  connection max-bandwidth <kbps>
  fast-start {query-string-param <string> | size <KB>}
  seek-config query-string-param <string>
    seek-flv-type {byte-offset | time-secs | time-msec}
    seek-length query-string-param <string>
    seek-mp4-type query-string-param {time-secs | times-msec}
  tunnel-mode {enable | disable}
```

Notes:

- **assured-flow**—Set AFR for this virtual player, this assures that content is delivered at least at the specified rate (but no more than the configured **connection max-bandwidth**) for the session. In prefix mode, use **no assured-flow** to disable again, if needed. By default, AFR is disabled (no delivery rate is assured). See [“Using Network Connection Assured Flow” on page 100](#) for an example.
 - **auto**—Not supported in Release 2.0.7.
 - **query-string-param**—Specify a string; referenced value must be in kilobytes.
 - **rate**—Define a static value, in kbps. A value of **0** (zero) means no throughput at all.
- **cache-name**—Set query params for:
 - **video-id**—Specify a query param string whose value provides the requested video ID (for example, **id**). YouTube video URI requests do not specifically associate a name to a video asset in the URI, instead a unique query param is used.
 - **format-tag**—Specify a query param string whose value provides the requested format (for example, **fmt** or **itag**). YouTube video URI requests do not specifically associate a format to a video asset in the URI, instead a unique query param is used. Acceptable format values are listed in [“Using Virtual Player Type YouTube” on page 131](#).
- **connection max-bandwidth**—Set the maximum bandwidth for a session. The actual session bandwidth is between the AFR and this value. Even if there is available bandwidth in the link, Media Flow Controller does not allocate more than this value for a session. When it is a full download, Media Flow Controller tries to allocate the max-bandwidth to the session. The default is **0** (unbounded) with the Media Flow Controller license, **200**

kbps without it; you must have the license to change the unlicensed default. In prefix mode, use **no connection** to reset the default.

- **fast-start**—Deliver the first set of kilobytes at either the maximum session speed or the available bandwidth; in prefix mode, use **no fast-start** to disable (default is disabled).
 - **query-string-param**—Specify a string (associated value must be in kilobytes).
 - **size**—Define how many kilobytes should be expedited.
- **seek-config query-string-param <string>**—Specify a string to implement MP4 or FLV seek (allows the client player to seek to a specific location of the URL). In prefix mode, use **no seek** to disable. Optionally:
 - **seek-flv-type**—Configure the FLV seek type:
 - **byte-offset**—The value of the seek **query-string-param** sent by the client will be, in bytes. This option must be set for **seek-length** for FLV.
 - **time-secs**—The value of the seek **query-string-param** sent by the client will be, in seconds.
 - **time-msec**—The value of the seek **query-string-param** sent by the client will be, in milliseconds.
 - **seek-length query-string-param**—Specify a string to signal the number of bytes of data to send from the **seek** start position; referenced value must be in bytes. For FLV support of this option, **seek-flv-type** must be set to **byte-offset**.
 - **seek-mp4-type**—Configure the MP4 seek type:
 - **time-secs**—The value of the seek **query-string-param** sent by the client will be, in seconds.
 - **time-msec**—The value of the seek **query-string-param** sent by the client will be, in milliseconds.
 - **tunnel-mode {disable | enable}**—If enabled, all **seek** requests to the origin server are tunneled; typically this option needs to be enabled only when the origin site changes their **seek** mechanism. Default is disabled.

virtual-player type smoothstream-pub

This virtual-player type is required for Smoothstream videos. See [Chapter 5, “Configuring Virtual Players, Media Fetch and Pre-Staging \(CLI\)”](#) and [“Configuring SmoothStream Video Caching \(CLI\)” on page 134](#) for implementation details. This virtual player supports on-demand, file-based media only; it does not support live streaming.

```
virtual-player <name> type smoothstream-pub
  fragment-tag <string>
  quality-tag <string>
```

Notes:

- **fragment-tag**—Set an identifier whose value describes to Media Flow Controller the timestamp of the requested media segment. Default is **Fragments** (case sensitive).
- **quality-tag**—Set an identifier to describe to Media Flow Controller the bit rate of the requested media segment. Default is **QualityLevels** (case sensitive).

Example format of a Smooth Streaming URL created by a Silverlight player:

```
http://test.media.com/bunny.ism/QualityLevels(400000)/
Fragments(video=134345672)
```

In the above example, the player is requesting of Media Flow Controller a media segment whose bit rate is 400 kbps at a timestamp of 134345672 in the video stream.

web

Use these commands to enable and configure settings for the Web interface (also referred to as the Management Console), and proxy.

Configure the Web interface. Web proxy commands follow. See [“Configuring the Web Interface \(CLI\)” on page 120](#) for task details.

```
web
  auto-logout <number_of_minutes>
  enable
  http enable [port <TCP_port>] [redirect]
  httpd listen enable [interface <interface_name>]
  https enable [port <TCP_port>] [certificate regenerate]
  proxy
    auth
      authtype {none | basic}
      basic {password <plaintext_password> | username <username>}
      host <IP_address> port <TCP_port>
  session
    renewal <number_of_minutes>
    timeout <number_of_minutes>
```

Notes:

- **auto-logout**—Control the length of user inactivity required before the Web interface automatically logs out a user. The **no** variant disables the automatic logout feature. Default is **15 minutes**.
- **enable**—Enable or disable (with **no**) the Web interface. Default is **enabled**.
- **http**—Configure HTTP access to the Web interface.
 - **enable**—Enable or disable (with **no**) HTTP access to the Web interface. Default is **enabled**. This setting is only meaningful if the Web interface as a whole is enabled via the **web enable** command.
 - **port**—Set the port number for HTTP access to the Web interface. Default is **8080**. The **no** variant resets to default, but does not disable HTTP.
 - **redirect**—Enables re-direction to HTTPS to mandate secure access.
- **httpd listen**—Configure Web server interface access restrictions.
 - **enable**—Enable or disable (with **no**) the listen interface restricted list for HTTPD. If enabled and at least one non-DHCP interface is specified in the list, HTTP connections are only accepted on those specified interfaces. When disabled, HTTP connections are accepted on any interface. Default is **enabled**.
 - **interface <interface_name>**—Add or delete (with **no**) interfaces to the **listen** list. If the interface is also running as a DHCP client, it is as if the interface was not added.

If DHCP is later turned off on this interface, it is as if the interface was then added to the listen list. Default is **eth0**.

- **https**—Configure HTTPS access to the Web interface.
 - **certificate regenerate**—Generate a new certificate to use for HTTPS.
 - **enable**—Enable or disable (with **no**) HTTPS (HTTP over SSL) access to the Web interface. Default is **enabled**. This setting is only meaningful if the Web interface as a whole is enabled.
 - **port**—Set the port number for HTTPS. Default is **443**. The **no** variant resets it to the default, but does not disable HTTPS.
- **proxy**—Configure Web proxy settings. See [“web proxy.”](#) for details.
- **session**—Configure session settings.
 - **renewal**—Control the length of time before Web session cookies are automatically regenerated. Default is **30** minutes. Use **no** to reset default.
 - **timeout**—Configure time after which a session expires. Default is **150** minutes.

show web

Web interface settings.

web proxy

Manage Web proxy settings. Web proxy is disabled until you configure a host. See [“Configuring the Web Interface Proxy \(CLI\)” on page 120](#) for task details.

web proxy

auth

```

auth
  authtype {<auth_type> | none | basic}
  basic password <plaintext_password> username <username>
  host <IP_address> [port <TCP_port>]

```

Configure a Web proxy; use **no web proxy** to delete.

Notes:

- **auth**—Configure authentication settings for the Web proxy.
 - **authtype**—Configure the type of authentication to be used with a Web proxy. Only matters if a proxy is configured with **web proxy host**. The **no** variant resets the **authtype** to its default, which is **none**.
 - **none**—No authentication required.
 - **basic**—HTTP basic authentication.
 - **basic**—Configure HTTP basic authentication settings for the Web proxy.

- **password** <plaintext_password>—Specify a plaintext password for HTTP basic authentication with an authenticating proxy. Only used if the **web proxy auth authtype** is set to **basic**. The password is accepted and stored in plaintext.
- **username** <username>—Specify a username for HTTP basic authentication with an authenticating proxy. Only used if the **web proxy auth authtype** is set to **basic**. The user name is accepted and stored in plaintext.
- **host**—Specify a proxy to be used for any HTTP or FTP downloads. If no port is specified, the default is **1080**.

write

```
write
  memory
  terminal
```

Notes:

- **memory**—Save running configuration to the active configuration file.
- **terminal**—Show the currently running configuration, rather than the active saved configuration. Lists commands to recreate the current running configuration.

INDEX

A

- aaa (authentication) command [365](#)
 - aaa (authorization) command [365](#)
 - accesslog
 - configure copy for SFTP [227](#)
 - disabling [366](#)
 - options [366](#)
 - troubleshooting rotation [230](#)
 - accesslog command [366](#)
 - activating
 - media-cache [405](#)
 - namespaces [412](#)
 - new disks [108](#)
 - adding
 - ARP entries [368](#)
 - bonding interfaces [369](#)
 - caches to namespaces [410](#)
 - DNS servers [396](#)
 - domain list entry [395](#)
 - entries to ARP cache [368](#)
 - event notify recipients [384](#)
 - hostname/IP mappings [396](#)
 - hosts for SNMP traps [431](#)
 - HTTPD listen interface [459](#)
 - interface comments [393](#), [393](#)
 - NTP servers [422](#)
 - per-class logging overrides [402](#)
 - second address to interface [393](#)
 - SNMP listen interfaces [431](#)
 - SSH client known host [433](#)
 - SSH server listen interface [434](#)
 - TACACS server [444](#)
 - users [96](#)
 - Admin up, interface state [394](#)
 - admin user default password [97](#)
 - admission control, overview [48](#)
 - AES-128, about [95](#)
 - alarms
 - configure [435](#)
 - stats [437](#)
 - watermark [435](#)
 - analytics cache-ingest size-threshold command [367](#)
 - analytics cache-promotion command [367](#)
 - analytics command [367](#)
 - analytics memory-limit command [368](#)
 - application fms command [368](#)
 - applying
 - configurations [118](#)
 - Media Flow Controller license [97](#)
 - policies via namespace [50](#)
 - ARP (address resolution protocol) [368](#)
 - arp command [368](#)
 - assured flow
 - network connection [420](#)
 - overview [47](#)
 - type qss-streamlet virtual players [453](#)
 - virtual player type 1 [452](#)
 - assured-flow rate, setting [100](#)
 - authentication
 - access log push [366](#)
 - options [365](#)
 - RADIUS (CLI) [423](#)
 - RADIUS (Web Interface) [259](#)
 - SNMP, community-based [430](#)
 - TACACS (CLI) [444](#)
 - TACACS (Web Interface) [259](#)
 - Web proxy [460](#)
 - authorization
 - options [365](#)
 - user capabilities [447](#)
 - auto-logout
 - CLI [373](#)
 - Management Console [459](#)
- ## B
- banner command [368](#)

- banners
 - commands [368](#)
 - configuring [92](#)
- bond command [369](#)
- bonding
 - commands [369](#)
 - configuring [93](#)
- boot
 - commands [370](#)
 - image file, location [391](#)
 - system (CLI) [119](#)
- boot command [370](#)
- boot process, about [228](#)
- build ID and date, checking [117](#)
- C**
- cache
 - ARP, add/delete entries [368](#)
 - configuring analytics [100](#)
 - hierarchy [45](#)
 - media, configure [404](#)
 - namespace options [307](#)
 - performance tuning, reverse proxy [56](#)
 - performance tuning, transparent proxy [74](#)
 - promotion troubleshooting [226](#)
 - RAM, display current [215](#)
 - show RAM [429](#)
- cachelog command [371](#)
- caching
 - cookies [104](#)
 - validating objects' timestamps [412](#)
- caching all contents for a Website, How To [121](#)
- capability, user accounts [447](#)
- changing
 - accesslog defaults warning [183](#)
 - configuration (CLI) [376](#)
 - configuration (Web Interface) [276](#)
 - configurations [377](#)
 - duplex/speed warning [393](#)
 - interface IP addresses [393](#)
 - SSH server keys [434](#)
 - terminal modes [380](#)
 - user capabilities [447](#)
- CHDs, stats [439](#)
- checking version and status [117](#)
- clear arp-cache command [372](#)
- CLI
 - configuration modes [83](#)
 - logging in [83](#)
 - options [84](#)
 - prefix mode [83](#)
- cli command [372](#)
- client
 - DHCP restart (CLI) [89](#)
 - RADIUS retransmissions [424](#)
 - SSH [430](#)
 - TACACS retransmissions [444](#)
 - TELNET [445](#)
- clock
 - configuring [92](#)
 - set for system [375](#)
 - set with NTP [423](#)
- clock command [375](#)
- collect counters command [375](#)
- command
 - modes [83](#)
 - options [84](#)
- comment, add/remove for interface [393](#), [393](#)
- community, SNMP [430](#)
- concurrent session
 - command [420](#)
 - setting [100](#)
- configuration changes, deferred update [153](#)
- configuration command [376](#)
- configuration modes [83](#)
- configuration text command [378](#)
- configurations
 - manage [377](#)
 - running, save [461](#)
 - saving [88](#)
 - saving to another system [118](#)
- configure terminal command [380](#)
- configuring
 - banners, system clock [92](#)
 - caching [404](#)
 - caching analytics [100](#)
 - cut and paste for interfaces [90](#)
 - DSR load balancing [160](#)
 - email event notifications [208](#)
 - interfaces, hostname, domain list, DNS, and default gateway (CLI) [88](#)
 - logging [198](#)

- namespaces [141](#)
 - static routes, link bonding [93](#)
 - user accounts [447](#)
 - users and passwords [96](#)
 - Web interface [120](#)
 - connecting [83](#), [232](#)
 - Management Console [459](#)
 - RADIUS options (CLI) [423](#)
 - RADIUS options (Web Interface) [259](#)
 - SSH options [434](#)
 - connection max-bandwidth
 - network [420](#)
 - setting [100](#)
 - connection pooling [47](#)
 - cookies
 - and state management [104](#)
 - controlling caching [104](#)
 - controlling in requests [104](#)
 - copying
 - access log [366](#)
 - configuration files [377](#)
 - cut and paste interface configuration [90](#)
 - counters
 - per-namespace [214](#)
 - reset [376](#)
 - reset for rate-limit stats [436](#)
 - set thresholds [202](#)
 - creating
 - bonding interface [369](#)
 - configuration file [377](#)
 - namespaces [141](#)
 - virtual players [449](#)
 - cryptographic hash algorithms [95](#)
- ## D
- deactivating
 - HDDs [309](#)
 - media-cache [405](#)
 - namespaces [412](#)
 - debug command [380](#)
 - debugging
 - dumps [224](#)
 - excessive output [223](#)
 - files [387](#)
 - default gateway, configuring (CLI) [88](#)
 - defaults
 - admin user password [97](#)
 - AFR (AssuredFlow rate) [48](#)
 - CLI [84](#)
 - connection limit, unlicensed [97](#)
 - delivery protocol interface [102](#)
 - domain [208](#)
 - email events detail option [208](#)
 - logging level [198](#)
 - login credentials (CLI) [85](#)
 - login credentials (Web Interface) [233](#)
 - management port [120](#)
 - media cache tiers [103](#)
 - namespace, origin fetch options [341](#)
 - SNMP "public" community enabled [329](#)
 - thresholds [207](#)
 - default-user, local authentication (CLI) [365](#)
 - default-user, local authentication (Web Interface) [269](#)
 - defining
 - hostname (CLI) [88](#)
 - namespaces [141](#)
 - network connection parameters [420](#)
 - server map [170](#)
 - virtual players [123](#)
 - deleting
 - ARP cache entries [368](#)
 - bonding interfaces [369](#)
 - configuration files [377](#)
 - DNS servers [396](#)
 - domain list entry [395](#)
 - event notify recipients [384](#)
 - hostname / IP mappings [396](#)
 - hosts for SNMP traps [431](#)
 - HTTPD listen interface [459](#)
 - image file [391](#)
 - licenses [399](#)
 - namespace objects [146](#)
 - NTP servers [422](#)
 - per-class logging overrides [402](#)
 - second address from interface [393](#)
 - SNMP listen interfaces [431](#)
 - SSH client known host [433](#)
 - SSH server listen interface [434](#)
 - TACACS server [444](#)
 - users [97](#)
 - delivery
 - admission control [48](#)
 - connection pooling [47](#)
 - options, for virtual players [449](#)

- policies via namespaces [50](#)
 - protocol and interfaces (CLI) [380](#)
 - protocol and interfaces (Web Interface) [287](#)
 - restarting [89](#)
 - SmoothFlow, about [49](#)
 - supported HTTP requests [43](#)
 - supported protocols [42](#)
 - supported RTSP requests [44](#)
 - delivery policies
 - type break virtual players [235](#)
 - type generic virtual player [450](#)
 - delivery protocol command [380](#)
 - delivery protocols, supported [42](#)
 - deployment
 - process, reverse proxy [55](#)
 - process, transparent proxy [68](#)
 - requirements, reverse proxy [54](#)
 - requirements, transparent proxy [67](#)
 - DES, about [95](#)
 - DHCP, interfaces [393](#), [393](#)
 - Digital Millennium Copyright Act (DMCA), compliance [73](#)
 - disabling
 - accesslog [366](#)
 - disk caching [404](#)
 - interfaces [393](#)
 - logging to an account [448](#)
 - Management Console [459](#)
 - mfc_probe (internal watchdog) [217](#)
 - NTP server [422](#)
 - SNMP (CLI) [430](#)
 - SNMP (Web Interface) [329](#)
 - SSH server [434](#)
 - stats alarms [435](#)
 - TACACS retransmissions [444](#)
 - traps [430](#)
 - disks
 - CHD stats [439](#)
 - deactivating [309](#)
 - enabling caching [404](#)
 - replacing [108](#)
 - sample stats [440](#)
 - SNMP events [333](#), [432](#)
 - stats alarms [207](#)
 - using namespace cache-inherit [142](#)
 - display
 - accesslog settings [367](#)
 - ARP contents [368](#)
 - banners [369](#)
 - bonded interfaces [370](#)
 - bridged interfaces [371](#)
 - cachelog settings [372](#)
 - configured system commands [378](#)
 - current configuration [461](#)
 - current RAM cache [215](#)
 - errorlog settings [387](#)
 - files [388](#)
 - fuselog settings [391](#)
 - installed licenses [400](#)
 - interface information [394](#)
 - Management Console settings [460](#)
 - namespaces [413](#)
 - notification settings [385](#)
 - routing information [397](#)
 - SNMP settings [431](#)
 - SSH client information [433](#)
 - stats [204](#)
 - streamlog settings [444](#)
 - system time, date, and timezone [375](#)
 - system version [429](#)
 - threshold settings [450](#)
 - tracelog settings [446](#)
 - DMCA compliance [73](#)
 - DNS
 - add/delete servers [396](#)
 - configuring (CLI) [88](#)
 - show configuration [429](#)
 - domain list, configuring (CLI) [88](#)
 - domains
 - add IP entry [395](#)
 - email option [383](#)
 - show list [391](#)
 - DOS attacks, preventing [421](#)
 - downloading
 - cache log [191](#)
 - configuration files [377](#)
 - tech-support log [202](#)
 - tracelog [198](#)
 - DSR, direct server return, requirements [160](#)
 - duplex, interfaces [393](#), [393](#)
- ## E
- email
 - class options [385](#)
 - configuration options [383](#)

- configuring notifications [208](#)
 - event name options [384](#)
- email command [383](#)
- enable command [386](#)
- enabling
 - access log [366](#)
 - autosupport emails [208](#)
 - disk caching [404](#)
 - interfaces [393](#)
 - log seconds field [401](#)
 - Management Console [459](#)
 - NTP [422](#)
 - SNMP (CLI) [430](#)
 - SNMP (Web Interface) [329](#)
 - SSH server [434](#)
 - stats alarms [435](#)
 - traps [430](#)
- errorlog
 - command [386](#)
 - module names and codes [187](#)
- eth0, set [403](#)
- event notifications, configuring [208](#)
- exit command [387](#)
- export stats file [203](#)
- export, statistics [436](#)

F

- factory configuration, revert to [377](#)
- fast-start
 - overview [124](#)
 - virtual player type 0 [451](#)
- fetch
 - image file [392](#)
 - origin, namespace [414](#)
- file command [387](#)
- filename
 - configuration, new [376](#)
 - for access log [366](#)
 - stats export [436](#)
- files
 - configuration, create [377](#)
 - export statistics [436](#)
 - logging, configure [400](#)
 - upload stats [203](#)
- filesize, accesslog [367](#)
- first time login (CLI) [85](#)

- first time login (Web Interface) [233](#)
- fmsaccesslog command [388](#)
- fmsedgelog command [389](#)
- format
 - accesslog [366](#)
 - for access log [366](#)
 - logs, set [402](#)
- FTP
 - pre-staging content, How To [139](#)
 - requirement [33](#)
 - user, for namespace (CLI) [153](#)
- fuselog command [390](#)

G

- global
 - delivery options (CLI) [380](#)
 - delivery options (Web Interface) [287](#)
 - network connection options [419](#)
 - RADIUS [423](#)
 - SSH client options [432](#)
 - TACACS [259](#), [444](#)
- global resources, list [215](#)

H

- hardware specifications, Media Flow Controller [38](#)
- hash digests, type yahoo virtual players [454](#)
- hash verify
 - example [127](#)
 - virtual player type 0 [451](#)
- healthcheck, type yahoo virtual players [454](#)
- hierarchical caching, about [45](#)
- host
 - RADIUS, set [423](#)
 - TACACS [444](#)
- host ID, finding [117](#)
- host-key, SSH [434](#)
- hostname command [391](#)
- hostname, configuring (CLI) [88](#)
- HTTP
 - access to Management Console [459](#)
 - cache directive [307](#)
 - codes that trigger origin escalation [166](#)
 - fetch for videos, How To [138](#)
 - supported access methods [43](#)
 - transaction rate stat [208](#)
- http

managing cookies 104

I

identifying interfaces 393

idle timeout, network connection 420

image command 391

image file, management 391

installing

image file 391

licenses 399

upgrades 119

interface command 392

interfaces

access restrictions, SSH 434

add/delete second address 393

bond 93

bonding 369

changing IP addresses 393

configuring (CLI) 88

cut and paste configuring 90

delivery protocol 382

enable/disable 393

identifying 393

management 90

management, configure 403

origin server fetching 90

setting for media traffic (CLI) 380

setting for media traffic (Web Interface) 287

SNMP, access 431

traffic 90

internal watchdog

about 216

disable 217

IP address

for syslog 402

interfaces 393, 393

origin server 417

ip command 394

ip filter chain rule command 397

K

KB and KiB, definitions 32

key

license activation 399

RADIUS, global 424

RADIUS, host 423

SSH 434

TACACS, global 444

TACACS, host 444

L

ldap command 398

level

setting stats thresholds 435

severity, logs 402

license command 399

licenses

installing or deleting 399

Media Flow Controller, applying 97

link bonding, configuring 93

Link up, interface state 394

listen interfaces

HTTPD 459

SNMP 431

SSH, server 434

listen port, delivery protocol 382

listing

namespace objects 146

namespaces 413

virtual players 450

load balancing, DSR requirements 160

logging

configuration 400

configuring 198

local, configure 402

rotation caveat 183

set rotation criteria 401

setting thresholds 202

status sub-codes 175

logging command 400

logging in, first time (CLI) 85

logging in, first time (Web Interface) 233

login banner, set 368

logs, local, viewing 403

M

mailhub, email option 384

management command 403

Management Console, see Web interface 120

management interface, configure 403

max-bandwidth, network connection 420

maximum, allowed namespaces 301

- MB and MiB, definitions [32](#)
- MD5, about [95](#)
- media
 - protocols supported [54](#)
- media cache
 - AssuredFlow, about [47](#)
 - commands [404](#)
 - hierarchical caching [45](#)
 - managing (CLI) [103](#)
 - managing (Web Interface) [308](#)
 - namespace age threshold [415](#)
- Media Flow Controller license, applying [97](#)
- media, pre-staging with FTP [139](#)
- media-cache command [404](#)
- memory status, checking [117](#)
- merging, configurations [377](#)
- mfc_probe
 - about [216](#)
 - disable [217](#)
- mfdlog command [407](#)
- motd (message of the day) banner, set [368](#)
- moving
 - configuration files [377](#)
 - image files [391](#)
 - stats files [387](#)
- MTU, interfaces [393](#), [393](#)

N

- namespace
 - command [409](#)
 - get information [214](#)
 - object command [418](#)
 - origin server command [416](#)
- namespace delivery protocol
 - http client-request command [413](#)
 - origin-request command [415](#)
- namespace delivery protocol origin-fetch command [414](#)
- namespaces
 - about [50](#)
 - configuring [141](#)
 - FTP user (CLI) [153](#)
 - how many allowed [301](#)
 - set UUID [142](#)
 - using domain regex [144](#)
 - using for proxy configurations [148](#)
 - using FQDN:port [144](#)

- using match uri regex [146](#)
 - using object delete | list [146](#)
 - using origin-fetch cache-age [143](#)
 - using precedence [145](#)
 - validate object timestamp [412](#)
- network
 - connection options [419](#)
 - prevent DOS attacks [421](#)
 - setting connection options [100](#)
- network command [419](#)
- new session, admission checks [48](#)
- NFS
 - fetch for images, How To [137](#)
- notifications, events
 - commands [384](#)
 - setting [208](#)
 - setting thresholds [202](#)
- ntp command [422](#)
- ntpdate command [423](#)
- ntpdate, set system clock [423](#)

O

- origin clustering, overview [162](#)
- origin escalation
 - creating the XML file [166](#)
 - overview [164](#)
- origin fetch
 - namespace options [414](#)
 - ports vs. traffic ports [89](#)
- origin server
 - definition [32](#)
 - interface [90](#)
 - setting AssuredFlow rate [48](#)
 - settings, transparent proxy [69](#)

P

- passwords
 - access log [366](#)
 - pre-stage [410](#)
 - troubleshooting, lost admin [227](#)
 - users, configuring [96](#)
 - web proxy [121](#)
- peer, NTP, add/remove [422](#)
- performance tuning
 - reverse proxy [56](#)
 - transparent proxy [74](#)

- ping command [423](#)
- ports
 - auth, Radius [423](#)
 - auth, TACACS [444](#)
 - listen, delivery protocol [382](#)
 - SSH server [434](#)
 - traffic vs. origin fetch [89](#)
- prefix mode [83](#)
- pre-staging content [139](#)
- prevent simple DOS attacks [421](#)
- privileges, user accounts [447](#)
- product model, finding [117](#)
- product release, checking [117](#)
- Progressive Download (PDL), about [42](#)
- prompt-key, TACACS [444](#)
- protocols
 - supported for fetch and delivery [54](#)
- protocols, supported for delivery [42](#)
- proxy configurations [148](#)
- proxy, Management Console, setting [120](#)
- public community, SNMP [329](#)
- push, access log [366](#)

R

- radius-server command [423](#)
- RAID arrays, not supported [91](#)
- RAM cache, display current [215](#)
- ram-cache command [424](#)
- rate maps, type qss-streamlet virtual players [453](#)
- rebooting (CLI) [119](#)
- regex, using in namespaces [144](#), [146](#)
- release version, checking [117](#)
- reload command [425](#)
- removing, disk caches [105](#)
- renaming namespaces [142](#)
- replacing a bad disk [108](#)
- reset counters [376](#)
- reset factory command [425](#)
- resolving, incoming client requests [312](#), [426](#)
- resource-pool
 - find global resources [215](#)
- restarting
 - delivery service [89](#)

- reverse proxy
 - deployment process [55](#)
 - deployment requirements [54](#)
 - performance tuning [56](#)
 - protocol support [54](#)
- reverting
 - to factory configuration [377](#)
 - user account capability [447](#)
- rotating logs, caveat [183](#)
- rotating, logs [401](#)
- RTSP
 - fetch for videos, How To [138](#)
 - supported access methods [44](#)

S

- samples, stats [441](#)
- saving
 - config file to another system [118](#)
 - config to storage [378](#)
 - running config to active [461](#)
 - settings [88](#)
- SCP format and requirement [33](#)
- server
 - DNS, add/remove [396](#)
 - NTP, add/remove [422](#)
 - origin, namespace option [341](#)
 - RADIUS [423](#)
 - RADIUS (Web Interface) [259](#)
 - SNMP [430](#)
 - TACACS (CLI) [444](#)
 - TACACS (Web Interface) [259](#)
 - TELNET [445](#)
- server platforms, supported [38](#)
- server-map command [426](#)
- server-map for incoming requests [426](#)
- service command [428](#)
- session admission control, overview [48](#)
- setting
 - management interface [403](#)
 - media traffic interfaces (CLI) [380](#)
 - media traffic interfaces (Web Interface) [287](#)
 - network connection options [100](#)
 - SSH keys [227](#)
 - thresholds [202](#)
 - user capabilities [96](#)
 - UUID for namespace [142](#)

- severity level, logs [402](#)
 - SFTP format and requirement [33](#)
 - SHA1, about [95](#)
 - show command [429](#)
 - shutdown
 - interface [393](#), [393](#)
 - signature, for image installs [392](#)
 - slogin command [430](#)
 - SmoothFlow
 - about [49](#)
 - type smoothflow virtual players [455](#)
 - SNMP
 - "public" community [329](#)
 - events and recommendations [333](#)
 - snmp-server command [430](#)
 - speed, interfaces [393](#), [393](#)
 - ssh
 - client command [432](#)
 - command [432](#)
 - keys, setting [227](#)
 - server command [434](#)
 - states of user accounts [95](#)
 - static routes, configuring [93](#)
 - statistics
 - configure [435](#)
 - report files [387](#)
 - stats
 - alarm command [437](#)
 - chd command [439](#)
 - command [435](#)
 - sample command [441](#)
 - upload file [203](#)
 - status
 - media-cache [405](#)
 - namespace [412](#)
 - sub-codes [175](#)
 - streamlog command [443](#)
 - supported protocols [54](#)
 - system
 - boot [370](#)
 - clock, commands [375](#)
 - clock, configuring [92](#)
 - collect information [445](#)
 - display version [429](#)
 - hostname [391](#)
 - memory, checking status [117](#)
 - reboot (CLI) [119](#)
 - reset to factory [425](#)
 - set clock [423](#)
 - system log (syslog)
 - access log replication [366](#)
 - interpreting [201](#)
 - system-wide resources, list [215](#)
- ## T
- tacacs-server command [444](#)
 - tcpdump command [445](#)
 - tcpdump reports [387](#)
 - tech-support command [445](#)
 - tech-support log, viewing/uploading [202](#)
 - telnet command [445](#)
 - telnet-server command [445](#)
 - terminal command [446](#)
 - thresholds, setting [202](#)
 - timeout
 - CLI [373](#)
 - Management Console [459](#)
 - network connection [420](#)
 - RADIUS [424](#)
 - TACACS [444](#)
 - Web session cookie [460](#)
 - timezone, system [375](#)
 - tracelog command [446](#)
 - tracelog, uploading [198](#)
 - traceroute command [447](#)
 - traffic
 - interfaces (CLI) [380](#)
 - interfaces (Web Interface) [287](#)
 - ports vs. origin fetch ports [89](#)
 - transparent proxy
 - deployment process [68](#)
 - deployment requirements [67](#)
 - example configuration [70](#)
 - example YouTube configuration [71](#)
 - origin-server settings [69](#)
 - performance tuning [74](#)
 - upgrading for new functions [69](#)
 - traps, SNMP [430](#)
 - troubleshooting
 - accesslog rotation [230](#)

- accesslog via SFTP 226
- cache promotion 226
- changing duplex/speed 393
- changing IP addresses 393
- concurrent session limit default 101
- deferred configuration updates 153
- disk state messages 107
- excessive debugging output 223
- file not cached 225
- file not getting cached 225
- HTTP access methods 43
- licenses 224
- log rotation 183
- lost admin password 227
- Media Flow Controller license restrictions 97
- namespace configurations 225
- namespace domain configuration 225
- namespaces, activating 412
- no Web interface connection 230
- notifications 208
- RAID arrays 91
- restarting the delivery service 89
- RTSP access methods 44
- SCP and FTP requirement 33
- SNMP events and recommendations 333
- traffic ports vs. origin fetch ports 89
- URL length 226
- using namespace cache-inherit 142
- using namespace precedence 145

U

- upgrading configurations 119
- upload command 447
- uploading
 - accesslogs 189
 - cache log 191
 - configuration files 377
 - log files 401
 - namespace object list 147
 - namespace objects 146
 - service logs 189
 - stats file 203
 - stats files 203
 - system log file 200
 - tracelog 198
- uri-prefix
 - for namespaces 341
- uri-prefix, example and usage 34

URL

- SCP format 33
- SFTP format 33

- URL length, troubleshooting 226

- user accounts, states 95

- username command 447

users

- admin default password 97
- commands 447
- configuring 96

V

- validating objects in cache 412

- version, system 429

- video players, supported 38

viewing

- accesslogs 189
- local log file 403
- namespace objects 146
- service logs 189
- tech-support logs 202

virtual players

- listing all 450
- overview 51
- type break, delivery policies 235
- type generic, delivery policies 450
- type qss-streamlet, AFR rate maps 453
- type smoothflow 455
- type yahoo, hash digests and healthchecks 454
- type youtube 457

- virtual-player command 449

W

warnings

- accesslog defaults 183
- changing duplex/speed 393
- deactivating HDDs 309
- excessive debugging output 223
- reverting to factory defaults 377
- saving settings 88

- watermark, definition 435

- web command 459

Web interface

- configuring (CLI) 120
- troubleshooting no connection 230

- web proxy command 460

write command [461](#)

Y

YouTube, type youtube virtual players [457](#)

