



Junos[®] Space

Junos Space Media Flow Activate Management Guide

Release

3.3



Published: 2013-06-21

Revision 1

Juniper Networks, Inc.
1194 North Mathilda Avenue
Sunnyvale, California 94089
USA
408-745-2000
www.juniper.net

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

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

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

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

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

Junos Space Media Flow Activate Management Guide
Release 3.3, Revision 1
Copyright © 2013, Juniper Networks, Inc.
All rights reserved.

Revision History
04 January 2013—Junos Space Media Flow Activate Management Guide

The information in this document is current as of the date on the title page.

END USER LICENSE AGREEMENT

The Juniper Networks product that is the subject of this technical documentation consists of (or is intended for use with) Juniper Networks software. Use of such software is subject to the terms and conditions of the End User License Agreement ("EULA") posted at <http://www.juniper.net/support/eula.html>. By downloading, installing or using such software, you agree to the terms and conditions of that EULA.

Table of Contents

	About This Guide	xi
	Junos OS Documentation and Release Notes	xi
	Objectives	xi
	Audience	xi
	Documentation Conventions	xii
	Documentation Feedback	xiii
	Requesting Technical Support	xiv
	Self-Help Online Tools and Resources	xiv
	Opening a Case with JTAC	xiv
Part 1	Media Flow Activate	
Chapter 1	Overview	3
	Media Flow Activate Overview	3
	Understanding Media Flow Controller Management with Media Flow Activate	5
Part 2	Media Flow Devices	
Chapter 2	Configuration	11
	Monitoring a Media Flow Controller Dashboard from MFA	11
	Launching the Media Flow Controller Secure Console from MFA	12
	Configuring MFC Device Interfaces	14
	Configuring Bonded Interfaces	16
	Configuring Devices	18
	Upgrading or Rolling Back the Media Flow Controller Software Image	21
	Restoring Media Flow Controller Device Configuration	22
	Replicating Media Flow Controller Device Configuration	24
	BGP Traffic Steering Overview	26
	Configuring BGP-Based Traffic Steering	27
	Restarting Media Flow Controller Devices or Services	29
	Tagging Media Flow Controller Objects	30
Chapter 3	Resource Pools	33
	Resource Pools Overview	33
	Creating Resource Pools	34
	Actions on Resource Pools	35
	Provisioning Resource Pools to MFC Devices	36
	Managing Resource Pool Associations	37

Part 3	Design Elements	
Chapter 4	Access Log Profiles	41
	Understanding Access Log Profiles	41
	Creating Access Log Profiles	45
	Actions on Access Log Profiles	49
Chapter 5	Cache-Tuning Policies	51
	Understanding Cache-Tuning Policies	51
	Creating Cache-Tuning Policies	52
	Actions on Cache-Tuning Policies	65
Chapter 6	Origin Maps	67
	Understanding Origin Maps	68
	Creating Consistent Hash Maps	69
	Creating Escalation Maps	72
	Actions on Origin Maps	75
Chapter 7	Policy Scripts	77
	Understanding Policy Scripts	77
	Adding Policy Scripts	78
	Actions on Policy Scripts	79
Chapter 8	Virtual Players	81
	Understanding Virtual Players	81
	Creating Virtual Players	83
	Actions on Virtual Players	88
Part 4	Service Design	
Chapter 9	Overview	91
	Service Design Overview	92
Chapter 10	Network Optimization Service	97
	Network Optimization Services Overview	97
	Creating Network Optimization Services	98
	Creating Network Optimization Service XML Files for Import	105
Chapter 11	HTTP Reverse Proxy Service	111
	HTTP Reverse Proxy Services Overview	111
	Creating HTTP Reverse Proxy Services	112
	Creating Reverse Proxy Service XML Files for Import	123
	Creating Reverse Proxy Service XML Files for Export	132
	Purging Content from Media Flow Controller Devices	133
Chapter 12	Content Ingest Service	135
	Content Ingest Services Overview	135
	Creating Content Ingest Services	135
	Actions on Services	137

Part 5	Service Provisioning	
Chapter 13	Overview	141
	Provisioning Services Overview	141
	Provisioning Services	142
	Managing Provisioned Services	144
Part 6	Configuration Templates	
Chapter 14	Overview	149
	Configuration Templates Overview	149
Part 7	Network Monitoring	
Chapter 15	Fault Monitoring	157
	Fault Monitoring with SNMP	157
Part 8	Audit Logs	
Chapter 16	Overview	167
	Audit Logs Workspace Overview	167
Part 9	Job Management	
Chapter 17	Overview	175
	Job Management Workspace Overview	175
Part 10	Reference	
Chapter 18	Sample XML Schema	179
	Sample XML Schema for Network Optimization and Reverse Proxy Services	179
Chapter 19	Quick Reference	191
	Quick Reference to Tasks in Media Flow Activate	191
Part 11	Index	
	Index	201

List of Figures

Part 3	Design Elements	
Chapter 5	Cache-Tuning Policies	51
	Figure 1: Add Policy—Cache tier Tab	55
	Figure 2: Add Policy—Expiry & revalidation Tab	59
	Figure 3: Add Policy—Tunnel override decision Tab	62
Chapter 8	Virtual Players	81
	Figure 4: Add Virtual Player Window—Connection Properties Tab	85
Part 4	Service Design	
Chapter 10	Network Optimization Service	97
	Figure 5: Create Network Optimization Service—General Tab	99
Chapter 11	HTTP Reverse Proxy Service	111
	Figure 6: Create HTTP Reverse Proxy Service—General Tab	113
Part 5	Service Provisioning	
Chapter 13	Overview	141
	Figure 7: HTTP Reverse Proxy Provisioning - Select Devices	142

List of Tables

	About This Guide	xi
	Table 1: Notice Icons	xii
	Table 2: Text and Syntax Conventions	xii
Part 3	Design Elements	
Chapter 4	Access Log Profiles	41
	Table 3: Reason Codes Recorded on System Log for Access Log Files	44
Part 4	Service Design	
Chapter 10	Network Optimization Service	97
	Table 4: Example: Domain Regex Values	104
	Table 5: contentDirectDefinition Options and Elements	106
Chapter 11	HTTP Reverse Proxy Service	111
	Table 6: rproxyDefinition Options and Elements	125
Part 6	Configuration Templates	
Chapter 14	Overview	149
	Table 7: MFC CLI Commands Supported Through Junos Space Configuration Templates	152
Part 7	Network Monitoring	
Chapter 15	Fault Monitoring	157
	Table 8: SNMP Alarm Events	159
	Table 9: SNMP Clear Alarm Events	161

About This Guide

This preface provides the following guidelines for using the Media Flow Activate documentation:

- [Junos OS Documentation and Release Notes on page xi](#)
- [Objectives on page xi](#)
- [Audience on page xi](#)
- [Documentation Conventions on page xii](#)
- [Documentation Feedback on page xiii](#)
- [Requesting Technical Support on page xiv](#)

[Junos OS Documentation and Release Notes](#)

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 the Junos operating system (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 Activate graphical user 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.

Documentation Conventions

Table 1 on page xii defines notice icons used in this guide.

Table 1: Notice Icons



Icon	Meaning	Description
	Informational note	Indicates important features or instructions.
	Caution	Indicates a situation that might result in loss of data or hardware damage.
	Warning	Alerts you to the risk of personal injury or death.
	Laser warning	Alerts you to the risk of personal injury from a laser.

Table 2 on page xii defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
Bold text like this	Represents text that you type.	To enter configuration mode, type the configure command: user@host> configure
Fixed-width text like this	Represents output that appears on the terminal screen.	user@host> show chassis alarms No alarms currently active
<i>Italic text like this</i>	<ul style="list-style-type: none"> Introduces or emphasizes important new terms. Identifies book names. Identifies RFC and Internet draft titles. 	<ul style="list-style-type: none"> A policy <i>term</i> is a named structure that defines match conditions and actions. <i>Junos OS System Basics Configuration Guide</i> RFC 1997, <i>BGP Communities Attribute</i>
<i>Italic text like this</i>	Represents variables (options for which you substitute a value) in commands or configuration statements.	Configure the machine's domain name: [edit] root@# set system domain-name <i>domain-name</i>

Table 2: Text and Syntax Conventions (*continued*)

Convention	Description	Examples
Text like this	Represents names of configuration statements, commands, files, and directories; configuration hierarchy levels; or labels on routing platform components.	<ul style="list-style-type: none">To configure a stub area, include the stub statement at the[edit protocols ospf area area-id] hierarchy level.The console port is labeled CONSOLE.
< > (angle brackets)	Enclose optional keywords or variables.	stub <default-metric <i>metric</i>>;
(pipe symbol)	Indicates a choice between the mutually exclusive keywords or variables on either side of the symbol. The set of choices is often enclosed in parentheses for clarity.	broadcast multicast (<i>string1</i> <i>string2</i> <i>string3</i>)
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	rsvp { # Required for dynamic MPLS only
[] (square brackets)	Enclose a variable for which you can substitute one or more values.	community name members [<i>community-ids</i>]
Indentation and braces ({ })	Identify a level in the configuration hierarchy.	<pre>[edit] routing-options { static { route default { nexthop <i>address</i>; retain; } } }</pre>
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
GUI Conventions		
Bold text like this	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none">In the Logical Interfaces box, select All Interfaces.To cancel the configuration, click Cancel.
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select Protocols>Ospf .

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/us/en/local/pdf/resource-guides/7100059-en.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 Management tool: <http://www.juniper.net/cm/>

To verify service entitlement by product serial number, use our Serial Number Entitlement (SNE) Tool: <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 Management 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, see <http://www.juniper.net/support/requesting-support.html>.

PART 1

Media Flow Activate

- [Overview on page 3](#)

CHAPTER 1

Overview

- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)

Media Flow Activate Overview

Media Flow Activate (MFA) is a Junos Space–based management application that helps you to simplify the configuration and deployment of Juniper Networks Media Flow Controller devices. With Media Flow Activate, network administrators can centrally configure their content delivery nodes and rapidly provision new content delivery services.



NOTE:

- Use Media Flow Activate to manage Media Flow Controllers that you have discovered with the Junos Space Network Platform application. For information about discovering Media Flow Controllers, see the “Discovering Media Flow Controllers” section in the *Media Flow Activate Installation Guide*. At this time, you cannot use the SNMP method to discover Media Flow Controllers via the Junos Space Network Platform application; you must use the Ping method.
- Media Flow Activate version 3.3 can manage Media Flow Controller devices installed with a Media Flow Controller version 12.2.4 image.

Refer to the “Version Compatibility Matrix” table in the *Media Flow Activate Installation Guide*, for more information.

If you are an Internet service provider, you can create a **Network Optimization Service** to transparently cache a popular website, thereby saving bandwidth and optimizing Internet content delivery.

If you are a content provider or a Content Delivery Network (CDN) service provider who owns or manages any content, you can create an **HTTP Reverse Proxy Service** to efficiently deliver that content.

To use Media Flow Activate to manage Media Flow Controllers, see the following topics:

- [“Understanding Media Flow Controller Management with Media Flow Activate” on page 5](#) for information about managing discovered devices, including software upgrades, device restarts, and service restarts, on selected Media Flow Controllers.
- [“Configuring Devices” on page 18](#) for information about configuring the interfaces to deliver media for transparent and reverse proxy services.
- [“Configuring BGP-Based Traffic Steering” on page 27](#) for information about redirecting traffic from a peering router to Media Flow Controller by advertising certain destination IP network addresses.
- [“Resource Pools Overview” on page 33](#) for information about configuring resource utilization limits for concurrent connections and bandwidth usage for multiple tenants that are hosted on Media Flow Controller.
- [“Understanding Virtual Players” on page 81](#) for information about setting media delivery options for video trick play (such as seek, fast start, and full download) and authentication. You configure a virtual player for each type of media that you deliver. A single virtual player can be used in multiple configured services. See [“Creating Virtual Players” on page 83](#) for configuration details and for information about how to import a virtual player.
- [“Understanding Cache-Tuning Policies” on page 51](#) for information about setting cache-handling options, including how long objects can stay in the cache. A single cache-tuning policy can be used in multiple configured services. See [“Creating Cache-Tuning Policies” on page 52](#) for configuration details and for information about how to import a cache-tuning policy.
- [“Understanding Origin Maps” on page 68](#) for information about origin maps (consistent hash map and escalation map). You configure a consistent hash map when you want to create a cluster of nodes and distribute the incoming requests across these nodes, thereby increasing the cache storage capacity (see [“Creating Consistent Hash Maps” on page 69](#) for information about creating a consistent hash map). You configure an escalation map when you want to configure multiple redundant HTTP origin servers for failover protection (see [“Creating Escalation Maps” on page 72](#) for information about creating an escalation map).
- [“Understanding Access Log Profiles” on page 41](#) for information about configuring access log profiles. You configure an access log profile to tune the access log format and storage. Access logs are used to analyze the HTTP traffic handled by Media Flow Controller.
- [“Understanding Policy Scripts” on page 77](#) for information about configuring policy scripts. Using Media Flow Activate, you can bind a policy script to a service, which enables you to have a greater control over how Media Flow Controller caches and delivers objects when the service receives requests from clients.
- [“Service Design Overview” on page 92](#), [“Network Optimization Services Overview” on page 97](#), and [“HTTP Reverse Proxy Services Overview” on page 111](#) for information about configuring websites for media delivery, and [“Content Ingest Services Overview” on page 135](#) for information about preloading content from origin servers at predefined time intervals. See [“Creating Network Optimization Services” on page 98](#), [“Creating HTTP Reverse Proxy Services” on page 112](#), and [“Creating Content Ingest Services” on](#)

[page 135](#) for configuring Network Optimization service, HTTP Reverse Proxy service, and Content Ingest service, respectively.

- [“Provisioning Services Overview” on page 141](#), [“Provisioning Services” on page 142](#), and [“Managing Provisioned Services” on page 144](#) for information about provisioning configured services to selected Media Flow Controllers.
- [“Configuration Templates Overview” on page 149](#) for information about managing Media Flow Controller specific device templates. Device templates provides a way to configure the CLI commands that are not supported through the Media Flow Activate GUI.

Though you can configure a Media Flow Controller device by using any of the following options—device templates, Media Flow Activate GUI, or console—the typical flow would be to use the device templates to perform the infrastructure provisioning before performing the service provisioning by using the Media Flow Activate GUI. Use the console to configure a feature that is not supported by Media Flow Activate either through the purpose built GUI or device configuration templates.

- [“Fault Monitoring with SNMP” on page 157](#) for information about monitoring the performance statistics of Media Flow Controller devices.
- [“Audit Logs Workspace Overview” on page 167](#) for information about monitoring the tasks initiated from the Media Flow Activate GUI.
- [“Job Management Workspace Overview” on page 175](#) for information about monitoring jobs, which represent user-initiated actions on selected Junos Space objects.



NOTE: See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about configuring Media Flow Controllers. See [“Quick Reference to Tasks in Media Flow Activate” on page 191](#) for a list of tasks that you can perform in Media Flow Activate.

Related Documentation

- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Understanding Media Flow Controller Management with Media Flow Activate

This topic describes the **Manage MFCs** page and the actions you can take on this page.

After you access the Media Flow Activate **Manage MFCs** page, you can select any discovered devices and take actions on them by using the **Actions** list.

Click the **Actions** list, then click one of the following action links:

- **Launch DashBoard**—Click to display the Media Flow Controller Console login page. Log in to display the dashboard page for that Media Flow Controller. The statistical information and the graphs show the usage for the selected Media Flow Controller. See [“Monitoring a Media Flow Controller Dashboard from MFA” on page 11](#) for details about launching a dashboard from Media Flow Activate.
- **Launch Secure Console**—Click to display the Secure Console page. Enter your login credentials to open an SSH connection to connect to Media Flow Controller directly from Media Flow Activate. See [“Launching the Media Flow Controller Secure Console from MFA” on page 12](#) for details about launching a Media Flow Controller secure console from Media Flow Activate.
- **Software Image Management**—Click to display the **Software Image Management** page. See [“Upgrading or Rolling Back the Media Flow Controller Software Image” on page 21](#) for details about upgrading or rolling back a Media Flow Controller software image.
- **Restore Device(s)**—You can choose to restore or remove the service configurations of the selected Media Flow Controllers. See [“Restoring Media Flow Controller Device Configuration” on page 22](#) for details about restoring a Media Flow Controller device configuration.
- **Replicate Device**—You can replicate or clone the device configuration of one Media Flow Controller to one or more Media Flow Controllers. See [“Replicating Media Flow Controller Device Configuration” on page 24](#) for details about replicating a Media Flow Controller device configuration.
- **Device Configuration**—Click to display the **Device Configuration** dialog box. See [“Configuring Devices” on page 18](#) for details about configuring the Media Flow Controller delivery interfaces.
- **Configure BGP**—Click to display the **Configure BGP** dialog box. See [“Configuring BGP-Based Traffic Steering” on page 27](#) for details about configuring BGP-based traffic steering in Media Flow Controllers.
- **Restart Service**—Click to display the **Restart Service** confirmation page. Click **Yes** to restart the delivery service on the selected device or devices. See [“Restarting Media Flow Controller Devices or Services” on page 29](#) for details about restarting services on Media Flow Controllers.
- **Restart Device(s)**—Click to display the **Restart Device(s)** confirmation page. Click **Yes** to restart the selected device or devices. See [“Restarting Media Flow Controller Devices or Services” on page 29](#) for details about restarting Media Flow Controller devices.
- **Tagging an Object**—Click **Tag Device(s)**, **Untag Device(s)**, or **View Tags** to tag, untag, or view tags, respectively. See [“Tagging Media Flow Controller Objects” on page 30](#) for details about tagging Media Flow Controller objects.

From the **Manage MFCs** page, you can also:

- Manage or create design elements—Click **Design Elements** from the left navigation panel to navigate to the Design Elements workspace. See “[Understanding Access Log Profiles](#)” on page 41, “[Understanding Cache-Tuning Policies](#)” on page 51, “[Understanding Origin Maps](#)” on page 68, “[Understanding Policy Scripts](#)” on page 77, and “[Understanding Virtual Players](#)” on page 81 for information about design elements.
- Manage or create delivery services for websites—Click **Service Design** from the left navigation panel to navigate to the Service Design workspace. See “[Service Design Overview](#)” on page 92 for information about the services that are supported from Media Flow Activate.
- Provision existing services—Click **Service Provisioning** from the left navigation panel to navigate to the Service Provisioning workspace. See “[Provisioning Services Overview](#)” on page 141 for information about provisioning services to Media Flow Controllers.
- Monitor the faults in Media Flow Controllers—Click **Network Monitoring** from the left navigation panel to navigate to the Network Monitoring workspace. See “[Fault Monitoring with SNMP](#)” on page 157 for monitoring faults in Media Flow Controllers.
- Monitor the audit logs that are generated—Click **Audit Logs** from the left navigation panel to navigate to the Audit Logs workspace. See “[Audit Logs Workspace Overview](#)” on page 167 for information about monitoring the audit logs.
- Check the status of the jobs that you are running—Click **Job Management** from the left navigation panel to open the Job Management workspace > Jobs status page. You can check the status of the following job types: Provisioning, Restart Service, Software Upgrade, or Restart Devices. See “[Job Management Workspace Overview](#)” on page 175 for information about monitoring the jobs that are initiated from Media Flow Activate



NOTE: See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about the features described in this topic.

Related Documentation

- [Media Flow Activate Overview on page 3](#)
- [Configuration Templates Overview on page 149](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

PART 2

Media Flow Devices

- [Configuration on page 11](#)
- [Resource Pools on page 33](#)

CHAPTER 2

Configuration

- [Monitoring a Media Flow Controller Dashboard from MFA on page 11](#)
- [Launching the Media Flow Controller Secure Console from MFA on page 12](#)
- [Configuring MFC Device Interfaces on page 14](#)
- [Configuring Bonded Interfaces on page 16](#)
- [Configuring Devices on page 18](#)
- [Upgrading or Rolling Back the Media Flow Controller Software Image on page 21](#)
- [Restoring Media Flow Controller Device Configuration on page 22](#)
- [Replicating Media Flow Controller Device Configuration on page 24](#)
- [BGP Traffic Steering Overview on page 26](#)
- [Configuring BGP-Based Traffic Steering on page 27](#)
- [Restarting Media Flow Controller Devices or Services on page 29](#)
- [Tagging Media Flow Controller Objects on page 30](#)

Monitoring a Media Flow Controller Dashboard from MFA

Purpose	Media Flow Activate allows you to view the Dashboard for any selected Media Flow Controller.
Action	From the Manage MFCs page, select a device. On the Actions list, select Launch Dashboard . The Login page for the selected Media Flow Controller is displayed. Log in to see the Dashboard page of the Management Console for that Media Flow Controller.

Meaning The Dashboard provides usage information for the system.

Statistics

- **Cumulative since**—Time this Media Flow Controller has been running without reboot or shutdown
- **GB delivered**—Total byte count of all objects that Media Flow Controller has delivered since running
- **Cache hit ratio**—Number of objects that Media Flow Controller has served from the RAM or disk divided by the total number of objects served; this includes the following parameters:

- **Bandwidth**—Total number of bytes delivered from the RAM or disk divided by the total number of bytes delivered
- **Number of Requests**—Total number of objects delivered from the RAM or disk divided by the total number of objects delivered (irrespective of their size)
- **Objects Delivered**—Total number of objects served by this Media Flow Controller since running

Graphs

- **Open Connections**—Media Flow Controller connections to the client, both HTTP and RTSP, and origin manager connections (om-session)
- **Weekly Bandwidth Savings**—Saved bandwidth—that is, bandwidth used by traffic that did not come from the origin server
- **Cache Throughput**—Bandwidth and place from which data was served
- **Cache Tier Throughput**—Green indicates that data is served from cache; yellow indicates that data is promoted from cache; red indicates that data is evicted from cache



NOTE: See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about the Media Flow Controller dashboard.

Related Documentation

- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Launching the Media Flow Controller Secure Console from MFA

You can use the **Launch Secure Console** feature in Media Flow Activate to open an SSH session to connect to a previously discovered Media Flow Controller. After you connect to a device, a terminal window is opened for that SSH connection in which you can enter CLI commands to monitor or troubleshoot the device.

Because this feature initiates the SSH session from the Junos Space server (rather than from your browser), it provides a secure and reliable connection to Media Flow Controller. You can establish separate SSH connections to one or more Media Flow Controllers simultaneously. A separate window is spawned for each SSH connection. However, only one SSH session per Media Flow Controller can be established at a time.

To open an SSH connection to Media Flow Controller, the following conditions must be met:

- Media Flow Controller should have been previously discovered in Media Flow Activate. That is, you can establish SSH connections only to those Media Flow Controllers that are displayed in Media Flow Activate.
- SSH v2 is enabled on Media Flow Controller.
- The status of Media Flow Controller is “UP.”
- A valid username and password have been configured on the Media Flow Controller.
- The **terminal type** parameter is set to **console**, **dumb**, or **ansi**. If **terminal type** is set to **ansi**, make sure that the **Terminal length** parameter is set to a value of **999**.

You can set the **terminal type** by running the following commands (make sure that you perform this task before running any other CLI commands on the SSH console):

- a. After you log in to Media Flow Controller, type **enable**.

The following is displayed: *Media Flow Controller host name #*

- b. Type **configure terminal**.

The following is displayed: *Media Flow Controller host name (config) #*

- c. Type **terminal type console** to set up the **terminal type**.



TIP: By default, if you are inactive, you are automatically logged out of the SSH console after five minutes. To prevent this automatic log out due to inactivity, select **Platform > Administration > Manage Applications > (Choose) Network Application Platform > (action) Modify Application Settings > (link) User > Automatic logout after inactivity** and set the value to 30 minutes.

To launch the secure console of Media Flow Controller:

1. From the left navigation panel, click the plus sign (+) adjacent to **Media Flow Devices**. The **Manage MFCs** page is displayed.
2. Select Media Flow Controller.
3. On the **Actions** list, select **Launch Secure Console**. The **SSH to Device** page is displayed.
The **IP** field displays the IP address of the selected Media Flow Controller and is typically unavailable.
4. In the **Username** and **Password** fields, enter the administrator login credentials of the selected Media Flow Controller. The name and password must match the name and password configured on Media Flow Controller.
5. Click **Connect** to establish an SSH session with the selected Media Flow Controller. A terminal window opens in a non-modal pop-up with an SSH connection opened on the selected Media Flow Controller.



NOTE: You might encounter the error messages “Unable to Connect,” “Authentication Error,” or “Connection Lost or Terminated,” which are displayed as standard text in the terminal window. When an error occurs, all other functionality in the terminal window is stopped. If you encounter such an error, close the terminal window and open a new SSH session.

6. From the terminal window prompt, enter CLI commands to monitor or troubleshoot the device.



NOTE: See the “Secure Console” section of the *Junos Space Network Application Platform User Guide* for detailed information about this feature. For more information about the Media Flow Controller CLI commands, see the *Media Flow Controller CLI Command Reference Guide*.

Related Documentation

- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Configuring MFC Device Interfaces

Typically, eth0 is used for management, eth1 for origin fetch, and other interfaces for traffic.

To configure an MFC device interface:

1. From the left navigation panel, click **Media Flow Devices**. The **Manage MFCs** page is displayed.
2. Select the discovered device for which you want to configure the interfaces.
3. On the **Actions** list, select **Configure Interfaces**. The **Interface Configuration** page is displayed. All available interfaces are also displayed.
4. In the **Interfaces** section, select the interface that you want to configure and click **Edit**. The **Edit Physical Interface** page is displayed.



NOTE: When you edit a loopback interface, you can change only the secondary address parameters. Other parameters are not configurable.

5. On the **Edit Physical Interface** page, perform the following tasks:
 - In the **IPv4 Address** field, enter an IP address.
 - In the **Subnet Mask** field, enter an appropriate subnet mask.

- Next to the **Secondary Address List** section, click **Add** to add a list of secondary addresses. The **Add Secondary Address List** page is displayed.
6. On the **Add Secondary Address List** page that is displayed, perform the following tasks:
 - In the **Index** field, enter an appropriate value. You can enter only integers in this field.
 - In the **IPv4 Address** field, enter the secondary address.
 - In the **Subnet Mask** field, enter a subnet mask.
 - Select the **ARP** check box.
 - Click **Add**. You are taken to the **Edit Physical Interface** page.
 7. On the **Edit Physical Interface** page, in the **Link Options** section, perform the following tasks:
 - In the **MTU** field, add an appropriate value. You can enter a value from 0 through 65,535 bytes. The default value is **1500** bytes.
This sets the largest number of bytes that a frame can carry.
 - From the **Speed** list, select the appropriate speed. The default is **AUTO NEGOTIATION**.
 - From the **Duplex** list, select the appropriate mode. The default is **Auto**.
 - Select the **ARP** and **DHCP** check boxes.
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 does not contain a primary DHCP interface by default. Setting a primary interface ensures that DHCP messages arrive only on that interface. If you select the **DHCP** check box, the values you entered in the **IP Address** and **Subnet Mask** fields are not used.
 - Click **OK**.

You can enable an interface by selecting the interface and clicking the **Enable** button. Click **OK** on the confirmation page. A new job is created. At any point, you can select the interface that you have enabled and click **Disable** to disable the interface.

Related Documentation

- [Configuring Bonded Interfaces on page 16](#)
- [Configuring Devices on page 18](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Configuring Bonded Interfaces

Configure bonded interfaces to create a port channel or aggregated link for load distribution across links and increased link availability.



NOTE: Only one bonded interface is allowed. Up to four interfaces can be bonded in a single bonded interface configuration.

To configure a bonded interface:

1. From the left navigation panel, click **Media Flow Devices**. The **Manage MFCs** page is displayed.
2. Select the discovered device for which you want to configure a bonded interface.
3. On the **Actions** list, select **Configure Interfaces**. The **Interface Configuration** page is displayed. All available interfaces are also displayed.
4. In the **Bonded Interface** section, click **Add**. The **Add Bonded Interface** page displays.
5. On the **Bond Config** tab, specify the following information:
 - In the **Name** field, enter a name for the bonded interface.
 - In the **Description** field, enter a description for the bonded interface.
 - From the **Available** section, select the individual interfaces and use the right arrow to move them to the **Selected** section.
 - In the **Bonding Attributes** section, from the Mode list, select the bonding mode. The different modes supported include:
 - **BALANCE_RR**—This is a “Round-robin” mode, which sends TCP/IP packets belonging to the same session across multiple links. Out-of-order TCP packets coming through different links are retransmitted. This mode supports load balancing and failover. This is the default.
 - **BALANCE_XOR_LAYER3_PLUS_4**—In this mode, traffic to a particular network peer goes across multiple links, although packets belonging to a single connection or session do not span multiple links. This mode supports load balancing and failover. In this mode, a link is selected on the basis of the TCP port and IP address.
 - **LINK_AGG_LAYER3_PLUS_4**—This mode allows the automatic negotiation of port bundling to form a single logical channel between Link Aggregation Control Protocol (LACP) links. This mode also supports load balancing and failover.
 - In the **Up Delay Time** field, enter a value in milliseconds. You can enter a value from 0 through 92,23,37,20,36,85,47,75,807 milliseconds. The default value is 0 milliseconds.

This is the wait period before enabling a slave after detecting a link recovery.

- In the **Down Delay Time** field, enter a value in milliseconds. You can enter a value from 0 through 92,23,37,20,36,85,47,75,807 milliseconds. The default value is **0** milliseconds.

This is the wait period before disabling a slave after a link failure is detected.

- In the **Link Monitoring Time** field, enter a value in milliseconds. You can enter a value from 0 through 92,23,37,20,36,85,47,75,807 milliseconds. The default value is **100** milliseconds.

This is the frequency at which the links are monitored.

6. On the **Interface Configuration** tab, specify the following information:

- In the **IP Address** field, enter an IP address
- In the **Subnet Mask** field, enter an appropriate subnet mask.
- Next to the **Secondary Address List** section, click **Add** to add a list of secondary addresses. The **Add Secondary Address List** page is displayed.

On the **Add Secondary Address List** page, specify the following information:

- In the **Index** field, enter an appropriate value. You can enter only integers in this field.
 - In the **IP Address** field, enter the secondary address.
 - In the **Subnet Mask** field, enter a subnet mask.
 - Select the **ARP** check box.
 - Click **Add**.
- In the **Link Options** section, in the **MTU** field, enter an appropriate value.

This sets the largest number of bytes a frame can carry. You can enter a value from 0 through 65,535 bytes. The default value is **1500** bytes.

- Select the **ARP** and **DHCP** check boxes.

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 does not contain a primary DHCP interface by default. Setting a primary interface ensures that DHCP messages arrive only on that interface. If you select the **DHCP** check box, the values you entered in the **IP Address** and **Subnet Mask** fields are not used.

7. Click **Add**.

You can modify the bonded interfaces by selecting the bonded interface and clicking the **Edit** button. You can also enable the bonded interface by selecting the interface and clicking the **Enable** button. Click **OK** on the Confirmation page. A new job is created. At any point in time, if you want to disable the bonded interface, select the bonded interface and click the **Disable** button.

You can delete a bonded interface by selecting the bonded interface and clicking the **Delete** button.

Related Documentation

- [Configuring MFC Device Interfaces on page 14](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Configuring Devices

You configure delivery interfaces for Media Flow Controller devices to specify which interfaces to use for media traffic. In this configuration, you also select the Proxy mode for selected devices. The Proxy mode can be **T-Proxy** for transparent proxy or **R-Proxy** for reverse proxy.

Transparent proxies cache popular content and optimize backhaul network utilization. You make the cache 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 server. A transparent proxy requires no browser configuration and is not visible to end users.

Reverse proxies cache and deliver content for a set of domains; client requests are routed to a configured IP address. This setup reduces network and CPU load on an origin server by serving previously retrieved content and enhances user experience by decreasing latency.

Before you begin, you need to know which interfaces you want to use for media delivery. Typically, eth0 is reserved for management and eth1 is reserved for origin server traffic.

To configure delivery interfaces on selected Media Flow Controllers:

1. From the left navigation panel, click **Media Flow Devices**. The **Manage MFCs** page is displayed.
2. From the list of Media Flow Controllers that are listed on the Manage MFCs page, select the Media Flow Controller that you want to configure.
3. On the **Actions** list, select **Device Configuration**. The **Device Configuration** page is displayed.
4. For **Proxy mode**:
 - Select **T-Proxy** to configure the interface for transparent proxy services. This sets the selected Media Flow Controller interfaces for delivery (**delivery protocol http interface <client_traffic_NIC>**) and enables them for transparent proxy (**delivery protocol http transparent <client_traffic_NIC> enable**).
 - Select **R-Proxy** to configure the interface for reverse proxy services. This sets the selected Media Flow Controller interfaces for delivery (**delivery protocol http interface <client_traffic_NIC>**).
5. In the **Select Delivery Interfaces** area, from the **Available** pane, double-click the interfaces you want to receive and respond to delivery requests. Each double-clicked interface moves to the **Selected** pane.

6. In the **Origin Lookup** section, in the **Connection Timeout** field, enter an appropriate value in seconds. You can enter a value from 6 through 120 seconds. The default value is **10** seconds.
7. Select the **Use multiple IP address in DNS response** check box.
8. If you want Media Flow Controller to authenticate the origin server before downloading content, select the **Enable SSL Authentication** check box. If this feature is enabled, the origin server is authenticated every time a transaction is initiated with the origin server by Media Flow Controller. Media Flow Controller uses the trusted Certification Authority (CA) certificates that are configured in the device for this purpose. In addition, there are a few more configuration steps that you have to perform to enable this feature for reverse proxy and content ingest services. These configuration steps are listed in the [“Creating HTTP Reverse Proxy Services” on page 112](#) and [“Creating Content Ingest Services” on page 135](#) sections.
9. In the **Log File Handling** area, in the **Purge Frequency** field, enter the number of hours for which you want to store the log files in Media Flow Controller. The log files are purged when this value is reached. The default value is six hours. That is, Media Flow Controller deletes files that are older than six hours and retains files that are less than six hours. By default, the log files are stored in the LogExport folder from which they are purged. If you have previously configured any value for this parameter in Media Flow Controller (either through the MFA GUI or CLI), the UI displays this value in this field, instead of the default value.

If you have configured both **Purge Frequency** and **Log Partition Size Limit**, Media Flow Controller purges the log files when one of the criteria is met (whichever comes first).

To disable time-based purging, set the purge frequency to zero. You can enter a value from 1 through 24 hours. The default value is **6** hours.
10. In the **Log Partition Size Limit** field, specify the percentage of disk size that the LogExport folder can reach. The log files within this folder are purged when this value is reached. The default value is **85%**. That is, when the LogExport folder size is 85% of the disk size, Media Flow Controller purges all the log files within this folder. After the purge, the log files are lost. We recommend that you store the log files in an external server at regular intervals. If you have previously configured any value for this parameter in Media Flow Controller (either through the MFA GUI or CLI), the UI displays this value in this field, instead of the default value. You can configure a value from 0.001 through 100%.

If you have configured both **Purge Frequency** and **Log Partition Size Limit**, Media Flow Controller purges the log files when one of the criteria is met (whichever comes first).
11. Select the **Enable Log Pull** check box.
12. In the **User Authentication Key** field, provide the authentication key to enable an external client to automatically pull access log files from Media Flow Controller's LogExport folder without password authentication.

Only the LogTransfer user is allowed to log in to Media Flow Controller by using the SFTP protocol. Follow the instructions listed in the [“Using SSH in Automated Scripts \(CLI\)”](#) section of the *Juniper Networks Media Flow Controller Administrators Guide* to generate the SSH key for this user from an external client.



CAUTION: To generate the access log files, it is necessary that you configure the access log profile first and then associate the log profile with a reverse proxy service. For more information about creating an access log profile and associating it with a reverse proxy service, see [“Creating Access Log Profiles” on page 45](#) and [“Creating HTTP Reverse Proxy Services” on page 112](#), respectively.

13. Click **Configure**.
14. Click **Ok** on the confirmation page. The selected interfaces are configured for the selected mode of delivery.



NOTE: See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about access log profiles.

Related Documentation

- [Creating Access Log Profiles on page 45](#)
- [Creating HTTP Reverse Proxy Services on page 112](#)
- [Creating Content Ingest Services on page 135](#)
- [Restarting Media Flow Controller Devices or Services on page 29](#)
- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Upgrading or Rolling Back the Media Flow Controller Software Image

Using Media Flow Activate, you can upgrade or roll back the software running on Media Flow Controllers. Both these operations interrupt the content delivery services and therefore require the network operations center (NOC) operators to gracefully take the Media Flow Controllers “out of service” for maintenance before performing an upgrade or a rollback.

Typically, a fresh Media Flow Controller containing two partitions has the same version of the software image installed on both partitions (that is, in active and standby partitions). To perform an upgrade, you can download the latest software image in the standby partition and start Media Flow Controller with this software image. However, if you find that the upgraded image is unstable or for some other reason you want to revert to the previous running image, Media Flow Activate provides the capability to roll back to the previous software image and configuration that was in use before it was upgraded. This image and the corresponding configuration are available in the standby partition.



NOTE:

Before you begin an upgrade or a rollback:

- Ensure that you have the URL of the software upgrade image when you want to perform an upgrade.
- The **Connection Status of the Media Flow Controllers** is up.

To perform a software upgrade on selected Media Flow Controllers by using Media Flow Activate:

1. From the left navigation panel, click **Media Flow Devices**.
2. Select the Media Flow Controllers and click **Software Image Management**.
3. In the **Download & Install** field, enter the URL of the image to be downloaded (a newer version of the image from what is currently running in Media Flow Controllers). You can select one of the following protocols for the download: **HTTP**, **HTTPS**, **SCP**, **FTP**, **SFTP**, and **TFTP**.
4. Click **Ok**. A pop-up is displayed with a status message, “**Please click on Job Id to view details**,” and the job ID. Click the job ID to check whether the Media Flow Controllers have successfully downloaded and copied the image to their standby partitions.
5. Select the Media Flow Controllers to which you have downloaded the new image and click **Software Image Management**.
6. Click **Boot standby image** to reboot Media Flow Controllers with the new image.
7. Click **Ok**. Media Flow Controllers reboot with the image in the standby partition. A pop-up is displayed with a status message, “**Please click on Job Id to view details**,” and the job ID. Click the job ID to check whether the upgrade was successful. If the upgrade was a failure, then Media Flow Controller automatically reboots with the image that

was running previously. In this case, you may have to retry upgrading the Media Flow Controllers.

If the upgrade is successful, the partition with the upgraded image becomes the active partition, whereas the other partition with the previous image becomes the standby partition. After you verify that the upgrade has been successful, make sure that you put the Media Flow Controllers back into service. The upgrade preserves the current saved configurations.

To roll back to the software image version from which it was upgraded, you can select the Media Flow Controllers and click **Boot standby image**. Media Flow Controller reboots with the image in the standby partition. A pop-up is displayed with a status message, “Please click on Job Id to view details,” and the job ID. Click the job ID to check whether the rollback was successful.



NOTE: When you initiate an upgrade or a rollback on multiple Media Flow Controllers, the job that is created comprises several subjobs; each subjob represents the user-initiated action for a single Media Flow Controller. From the Job Management workspace, select the main job to view details about the subjobs, such as whether they were successful or not.

**Related
Documentation**

- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Restoring Media Flow Controller Device Configuration

You can restore configurations within Media Flow Controller when it has experienced a hardware failure (such as a root disk crash) resulting in the loss of configuration data stored on the device. After the hardware is restored, either by replacing the FRU or the entire server, you can use Media Flow Activate to rapidly restore the services delivered by that device.

This section describes the process of restoring the device configuration of Media Flow Controller after a complete (non-recoverable) failure or corruption of the Media Flow Controller configuration file.



NOTE: Restore the device configuration only after you have restored the Media Flow Controller platform configuration (infrastructure-level configuration) by using the configuration templates stored in Media Flow Activate or manually.

The Media Flow Activate GUI provides you with the following options:

- **Restore service(s)**—This action restores the service configurations on the selected devices.

Media Flow Activate maintains a configuration database, which serves as the source of reference for configuration-related information. So, when you choose to restore the configuration of a device, Media Flow Activate restores the configuration from this database rather than from the device itself.

- **Remove all services**—This action removes the service configurations from the selected devices. Note that this action does not set the devices to the factory default configuration. It only wipes out or resets the existing service-specific configuration.

To restore or remove the device configuration:

1. Click **Media Flow Devices**.
2. Select the Media Flow Controllers whose configuration must be restored or removed.
3. From the **Actions** list, select either **Restore service(s)** to restore the services or **Remove all services** to remove the services from the Media Flow Controllers. A new job is created and a message with the job ID is displayed. Click the job ID to know whether the action is a success or a failure.

If the job is successful, then the following configurations are restored or removed:

- Content Ingest services
- HTTP Reverse Proxy services
- Network Optimization services
- Access log profiles
- Cache-tuning policies
- Policy scripts
- Origin maps
- Resource pools
- Virtual players

Usually, the following configurations are not restored or removed:

- Interface configuration
- Device configuration
- BGP configuration
- Space configuration templates

Example Workflow for Restoring Device Configuration

Consider that you want to restore the device configuration of Media Flow Controller, MFC-A. Perform the following steps to restore the device configuration:

1. Delete all the existing service-specific configurations by clicking **Remove all services** so that there are no conflicts while restoring the configuration of MFC-A.
2. Use Junos Space configuration templates to restore the platform-level configuration of MFC-A. In addition, manually restore the BGP, device, and interface configuration of MFC-A by using the UI.
3. Click **Restore service(s)** to restore the service configuration of MFC-A with the configuration stored in the Media Flow Activate database.

Related Documentation

- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Replicating Media Flow Controller Device Configuration

From Media Flow Activate, you can replicate or clone the device configuration from one Media Flow Controller to several Media Flow Controllers. This operation is useful when new Media Flow Controllers are added to the network to expand capacity or replace a failed device. You can quickly replicate standard configuration templates on the new devices.



NOTE: Replicate the device configuration only after you have replicated the Media Flow Controller platform configuration (infrastructure-level configuration) by using the configuration templates stored in Media Flow Activate or manually.

To replicate a device configuration:

1. Click **Media Flow Devices**.
2. Select the Media Flow Controller whose configuration must be replicated.
3. On the **Actions** list, select **Replicate Device**.
4. Select the Media Flow Controllers on which the device configuration must be replicated.
5. Click **Replicate**. A new job is created and a message with the job ID is displayed. You can click the job ID to know whether the replication job is a success or a failure. If the new device contains any previous resource pool configuration, then this action fails. This prevents any accidental overwrites.

Media Flow Activate maintains a configuration database, which serves as the source of reference for configuration-related information. So, when you choose to replicate the configuration of a device, Media Flow Activate replicates the configuration from this database rather than from the device itself.

If the job is successful, then the following configurations are replicated in the selected Media Flow Controllers:

- Content Ingest services
- HTTP Reverse Proxy services
- Network Optimization services
- Access log profiles
- Cache-tuning policies
- Policy scripts
- Origin maps
- Resource pools (both global and user-defined resource pool configurations)
- Virtual players

Usually, the following configurations are not replicated because they are unique to a device:

- Interface configuration
- Device configuration
- BGP configuration
- Space configuration templates—In this case, the Template feature provides the facility to set device-specific values and hence there is no need of replication.

If the purpose of replication is to replace the servers with a higher-performance server, then after the new server is put in service, one or more of the existing servers that were cloned or replicated can be removed from service.

**Related
Documentation**

- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

BGP Traffic Steering Overview

Media Flow Controller listens for HTTP requests and either serves the content from its local cache or fetches the requested content from origin servers. In this deployment, Media Flow Controller is exposed to much non-cacheable traffic that wastes valuable caching resources. The solution is to separate cacheable traffic from non-cacheable traffic to provide better bandwidth savings and throughput by efficiently using Media Flow Controller resources. The Border Gateway Protocol (BGP) traffic steering feature enables you to direct traffic from a peering router to Media Flow Controller by advertising certain destination IP addresses. You can use this feature in transparent and reverse proxy deployments.

- In transparent proxy deployment, you can use this feature as a replacement for the Policy Based Routing (PBR) feature, which is currently used to direct HTTP traffic from an access router to Media Flow Controller.
- In reverse proxy deployment, you can advertise a set of IP addresses mapped to different services or resource pools to an access router. This configuration can be used for load sharing within a POP.

To redirect cacheable traffic to Media Flow Controller:

1. Create a list of IP network addresses, so that any HTTP traffic intended for these IP addresses is redirected to Media Flow Controller. This list can be static or dynamic.

Media Flow Controller serves the requested object from its cache or fetches it from the origin servers.



NOTE: In networking terminology, this list of IP addresses is often called “IP whitelist.”

2. Configure the neighbors.

For example:

- Configure edge router A to which Media Flow Controller advertises the IP whitelist. Media Flow Controller acts as the next hop to this router for the IP addresses in the whitelist. This router must support basic BGP functionalities.
- Configure edge router B for Media Flow Controllers to forward any non-HTTP, non-RTSP, and non-RTMP traffic.

The flow of traffic is as follows:

- For request traffic with a destination IP address that is not listed in the whitelist, the access router A forwards the traffic to the server through the next-hop router B, bypassing Media Flow Controller. The return traffic takes the reverse path.
- For unwanted traffic (that is, non-HTTP, non-RSTP, and non-RTMP traffic) that is destined to the IP addresses that are listed in the whitelist, access router A forwards the traffic to Media Flow Controller, which then forwards it to the default next-hop

router B. The return traffic uses the client address as the destination IP address and bypasses Media Flow Controller.

- For intended traffic that is destined to IP addresses that are listed in the whitelist (that is, HTTP, RSTP, or RTMP requests), if there is a cache hit, the object is served from the cache. If there is no cache hit, Media Flow Controller fetches the object from the origin server and serves the request.

When intended traffic is forwarded to the origin server, one of Media Flow Controller's own IP addresses is used to replace the client address as the source IP address. This ensures that the response traffic is routed to Media Flow Controller for caching.



TIP: Because there is a likelihood that unintended traffic can be sent to Media Flow Controllers and this traffic needs to be forwarded to the next-hop router, IP forwarding must be enabled on Media Flow Controllers. Use the `network connection ip-forward` command to enable IP forwarding.

Related Documentation

- [Configuring BGP-Based Traffic Steering on page 27](#)
- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Configuring BGP-Based Traffic Steering

Border Gateway Protocol (BGP)–based traffic steering allows Media Flow Controller to receive only cacheable traffic by advertising certain IP addresses through BGP to the peering router.

To configure BGP routing on Media Flow Controller:

1. From the left navigation panel, click **Media Flow Devices**. The **Manage MFCs** page is displayed.
2. Select the discovered device in which you want to configure BGP.
3. On the **Actions** list, select **Configure BGP**. The **Configure BGP** page is displayed.
4. On the **Configure BGP** page, enter information in the following fields:

- **Local AS**—Enter the number of the autonomous system to which Media Flow Controller belongs. Specify a value between 1 and 4,294,967,295. This number identifies Media Flow Controller to other BGP routers.

An autonomous system is a group of routers and their associated networks operating under a single technical administration. This system appears as a single entity to external systems. Each autonomous system is assigned an identifying number by an Internet registry or a network provider. This number identifies the router (in this case, Media Flow Controller) to other BGP routers.

This field is mandatory. The AS number cannot be changed after it is configured because it is a unique identifier. If you need to change the AS number, disable BGP and then configure the AS number again.

- **Local Router ID**—Enter the IP address that identifies Media Flow Controller.
- **Keepalive interval**—Enter the BGP keepalive interval in seconds. The default value is **60** seconds. It is recommended that you enter a value from 7 through 21,845 seconds.

The keepalive time indicates how often Media Flow Controller sends a keepalive message to the neighboring router to indicate that it is still alive. BGP systems exchange keepalive messages to determine whether a link or host has failed or is no longer available.

- **Hold Time**—Enter the hold time in seconds. If a keepalive message is not received by Media Flow Controller from a BGP peer within the hold time, then the peer is marked down. The hold time must be at least three times the keepalive interval. The default value is **180** seconds. It is recommended that you enter a value from 21 through 65,535 seconds.
- **Redistribute connected networks**—Select to dynamically advertise directly connected networks.

For example, consider a number of logical interfaces (such as loopback interfaces) added to a physical interface. One way to advertise all these interfaces is to add them one by one from the **List of static network addresses to advertise** area. The other simpler option is to add the physical interface to the list of advertised static IP addresses and enable this feature, whereby any logical interface connected with this physical interface is automatically advertised to the BGP peers by Media Flow Controller.

- **List of peer routers**—Add a list of peering routers (BGP neighbors or BGP peers) by clicking **Add** and providing the IP addresses and AS numbers of the routers. Media Flow Controller establishes TCP connections with its peers by using the IP addresses provided in the configuration list. It is mandatory that you configure at least one peer router.

Media Flow Controller can have both external and internal peers. A peer is considered external if its AS number differs from Media Flow Controller's own AS number.

To remove a peer, select a peer and then click **Remove**.

- **List of static network addresses to advertise**—Add a list of destination IP addresses to advertise, such that any traffic intended for these IP addresses is redirected to

Media Flow Controller. Media Flow Controller serves the requested objects from its cache or fetches them from the origin servers. Click **Add** or **Remove** to add or remove network IP addresses.

- Click **Configure** to enable BGP traffic steering. If BGP is previously configured on Media Flow Controller, then click the **Reconfigure** or **Disable** button to reconfigure the existing BGP configuration or disable BGP traffic steering, respectively. Alternatively, click **Cancel** to close the configuration page.

Related Documentation

- [BGP Traffic Steering Overview on page 26](#)
- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Restarting Media Flow Controller Devices or Services

You can restart selected Media Flow Controller devices or the delivery services on them by using Media Flow Activate.

Restarting a Media Flow Controller brings down all services; restarting services on a Media Flow Controller stops media delivery services.

To restart selected Media Flow Controllers by using Media Flow Activate:

1. From the left navigation panel, click the plus sign (+) adjacent to **Media Flow Devices**. The **Manage MFCs** page is displayed.
2. Select the discovered devices that you want to restart.
3. On the **Actions** list, select **Restart Device(s)**. The **Restart Device(s)** configuration page is displayed.
4. Click **Yes** to proceed with the restart and close the window. Click **No** to cancel the restart and close the window. The selected Media Flow Controllers are restarted; there is no progress bar or completion message, but the status on the Manage MFCs page changes.

To restart services on selected Media Flow Controllers by using Media Flow Activate:

1. From the left navigation panel, click the plus sign (+) adjacent to **Media Flow Devices**. The **Manage MFCs** page is displayed.
2. Select the discovered devices on which you want to restart a service.
3. On the **Actions** list, select **Restart Service** from the list of available actions. The **Restart Service** configuration page is displayed.

4. Select the **Delivery Service** and **Content Ingest Service** options as needed and click **Restart**.
5. Click **Yes** to proceed with the restart and close the window. Click **No** to cancel the restart and close the window. A job is created. You can view the status of the job in the **Job Management** workspace.



NOTE: See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about restarting services and devices.

**Related
Documentation**

- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Tagging Media Flow Controller Objects

Tagging allows you to label and categorize objects from an application workspace manage inventory landing page. Subsequently, you can view and use these tags to easily search for multiple objects to view their status or perform a bulk action on them without having to select each object individually.

To tag objects, you navigate to the application workspace manage inventory landing page (such as Media Flow Devices, Service Design), select the objects that are to be tagged, and select **Tag** on the **Actions** list.



TIP: When you specify the tag name, make sure that it does not start with a space; contain a comma, double quotation marks, or parentheses; or exceed 255 characters.

The tags that you create are private and hence are visible only to you. However, to share the tags with other users, you need the “Tag Administrator” privileges.

For detailed instructions about tagging an object, refer to the “Tagging an Object” section in the *Junos Space Network Application Platform User Guide*.

After you tag the object, you can perform the following actions:

- **Filtering inventory using tags**—To perform an operation on multiple objects, tag the objects with an identical tag name and later filter them using the specific tag. After all the objects are listed, you can perform the permitted actions from the **Actions** list after selecting them.

To filter Media Flow Controller objects by using a tag:

1. On the workspace inventory page, from the **Search** list (usually located at the upper-right corner of the page), select **Tags**.

2. From the adjacent list, select the tag name. Alternatively, in the list, you can type the first letter of the tag to narrow down the list.

The inventory page displays only the objects that are tagged with the selected tag.

- Viewing tags—The **View Tags** action from application workspace inventory pages allows you to see all the tags that you have assigned to a managed object. You must first tag a managed object to see its tags. The tags that are displayed are those that are created by you (private tags) and those that are shared (public tags).

To view the tags assigned to an inventory object, navigate to the application workspace manage inventory landing page, select the object, and select **View Tags** on the **Actions** list.

For detailed instructions about viewing the tags associated with a managed object, refer to the “Viewing Tags” section in the *Junos Space Network Application Platform User Guide*.

- Untagging an object—You can untag or remove a tag from an object on a workspace inventory page. You can select only one object at a time to untag.

To untag an object, navigate to the application workspace manage inventory landing page, select the object, and select **Untag** on the **Actions** list. From the dialog box, select the tags to untag.

Related Documentation

- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

CHAPTER 3

Resource Pools

- [Resource Pools Overview on page 33](#)
- [Creating Resource Pools on page 34](#)
- [Actions on Resource Pools on page 35](#)
- [Provisioning Resource Pools to MFC Devices on page 36](#)
- [Managing Resource Pool Associations on page 37](#)

Resource Pools Overview

When you want to use Media Flow Controller to host multiple tenants (customers or Web portals), it becomes necessary to allocate the available Media Flow Controller resources among these tenants depending on various criteria, such as whether they are silver, gold, or platinum customers. Using Media Flow Activate, you can create a resource pool to configure resources that are relevant for a tenant, such as the maximum bandwidth and the maximum allowed sessions.

Before you start configuring a resource pool, consider the following parameters:

- The maximum bandwidth that you want to allocate to a resource pool
- The maximum concurrent sessions that you want to permit for a resource pool

Because the resource pool is provisioned on Media Flow Controller, the preceding parameters cannot exceed the maximum resources available on the device and are identified by the “global resource pool.”

After you create the resource pools:

- Provision the resource pools to Media Flow Controller. This action partitions the available resources among these pools.
- Bind a reverse proxy service to the resource pool.

The preceding actions ensure that the incoming requests to the website or domain configured in the service do not exceed the limits set by the resource pool, thereby ensuring that the remaining resources are available for other services configured in the same Media Flow Controller. If no pool is configured, the service receives all the available resources.

For more information about resource pools, see the “Media Flow Controller Multi-Tenancy Management” section in the *Media Flow Controller Administrator's Guide*.

**Related
Documentation**

- [Creating Resource Pools on page 34](#)
- [Actions on Resource Pools on page 35](#)
- [Provisioning Resource Pools to MFC Devices on page 36](#)
- [Managing Resource Pool Associations on page 37](#)
- [Provisioning Services on page 142](#)
- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Creating Resource Pools

You create a resource pool when you want to host multiple tenants on Media Flow Controller and want to partition the available system resources among these tenants. With resource pools, you can configure the maximum bandwidth and the maximum number of concurrent sessions that must be permitted for a tenant.

To configure a resource pool:

1. From the left navigation panel, click the plus sign (+) adjacent to **Media Flow Devices**. The **Manage MFCs** page is displayed.
2. Click the plus sign (+) adjacent to **Manage Resource Pools**.
3. Click **Add Resource Pool**. The **Add Resource Pool** page is displayed. Enter information in the following fields:
 - **Resource Pool Name**—Name of the resource pool, which must be unique
 - **Description**—(Optional) Description of the resource pool
 - **Max Bandwidth(Mbps)**—Maximum bandwidth that you want to reserve for this resource pool, in Mbps
 - **Max Concurrent Sessions**—Maximum number of incoming client sessions supported by this resource pool
4. Click **Ok**, **Cancel**, or **Reset**. **Ok** instantiates the values you set, **Cancel** closes the page, and **Reset** returns all values to their defaults.

You are returned to the Manage Resource Pools page. If you created a resource pool, you see the newly created resource pool on this page.

Against each resource pool, you can view the configured maximum bandwidth and maximum allowed sessions, whether the resource pool is provisioned to the Media Flow Controller server, the users who created and modified the resource pool, and when it was last modified.

When you associate a reverse proxy service with this resource pool, the incoming traffic to the website or domain is bound by the parameters defined by the resource pool.

**Related
Documentation**

- [Actions on Resource Pools on page 35](#)
- [Provisioning Resource Pools to MFC Devices on page 36](#)
- [Managing Resource Pool Associations on page 37](#)
- [Provisioning Services on page 142](#)
- [Resource Pools Overview on page 33](#)
- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Actions on Resource Pools

From the **Manage Resource Pools** page, you can perform the following actions on resource pools by clicking the links on the **Actions** list. You have to select the resource pools before performing any actions on them:

- **Modify Resource Pool**—Other than the resource pool name, you can modify all the configuration settings, such as the maximum bandwidth that you want to allocate to the resource pool, the maximum concurrent sessions that it can support, and the description.

You can modify only one resource pool at a time.

- **Delete Resource Pool(s)**—Delete one or more resource pools.

If the resource pool is provisioned to a device, deprovision the resource pool before deleting it.

- **Provision Resource Pool**—Provision the resource pools on Media Flow Controller servers. See [“Provisioning Resource Pools to MFC Devices” on page 36](#) for more information about provisioning the resource pools.
- **Manage Resource Pool Associations**—View the association of a resource pool to the Media Flow Controller servers to which it has been provisioned. You can reprovision the resource pool or deprovision it from Media Flow Controller servers depending on the actions that you choose to perform. See [“Managing Resource Pool Associations” on page 37](#) for more information about managing the resource pools associated with Media Flow Controllers.

You can manage only one resource pool at a time.

- **Tag It**—Tag the resource pools.
- **View Tags**—View the tags associated with a specific resource pool.
- **Untag It**—Untag the selected tags from the specific resource pool.

For more information about tagging, see [“Tagging Media Flow Controller Objects” on page 30](#).

- Related Documentation**
- [Provisioning Resource Pools to MFC Devices on page 36](#)
 - [Managing Resource Pool Associations on page 37](#)
 - [Provisioning Services on page 142](#)
 - [Creating Resource Pools on page 34](#)
 - [Resource Pools Overview on page 33](#)
 - [Media Flow Activate Overview on page 3](#)

Provisioning Resource Pools to MFC Devices

After creating a resource pool, you have to provision it on Media Flow Controller devices to ensure that the configured resources are reserved for that pool from the available system resources.

1. From the left navigation panel, click the plus sign (+) adjacent to **Media Flow Devices**.
2. Click **Manage Resource Pools**. The Manage Resource Pools page is displayed.
3. Select the resource pools.
4. On the **Actions** list, select **Provision Resource Pool**. The **Provision Resource Pool to Device** dialog box appears.

This dialog box displays all the available devices (**Device Name** column) to which you can provision the resource pool, including their IP addresses (**IP Address** column), whether they are up (**Connections Status** column), and the list of pools (**Current Resource Pool List** column) that have been previously provisioned on the device.

It is recommended that you provision the resource pools on those devices that are up. When a device is down, the provisioning may not be a success.

Additionally, click **View** under the **Current Resource Pool List** column to view the list of pools that were previously provisioned on a specific device before provisioning additional pools to the device.



CAUTION: While provisioning, make sure that the resources allocated to a resource pool do not exceed the available system resources.

5. Select the devices to which you want to provision the resource pools.
Use the **Search** option to display specific Media Flow Controllers by filtering using their names or tags.
6. Click **Next**. The selected pools and the selected devices are displayed. If you want to make any modification to this list, click **Previous** and make changes, as needed.
7. Click **Provision**. The **Job Information** dialog box appears with a job ID.
8. Click the job ID to check whether the provisioning job was successful or click **OK** to exit.

If the job is not successful, the possible causes could be one of the following:

- Devices were not up.
- Resource pool allocation exceeded the available system resources (global pool) for a certain device.

**Related
Documentation**

- [Managing Resource Pool Associations on page 37](#)
- [Provisioning Services on page 142](#)
- [Actions on Resource Pools on page 35](#)
- [Creating Resource Pools on page 34](#)
- [Resource Pools Overview on page 33](#)
- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Managing Resource Pool Associations

You can view the association of the resource pools to Media Flow Controller servers to which they have been provisioned and take suitable actions, such as reprovisioning them if the earlier provisioning had failed.

From the **Manage Resource Pool Associations** page, you can:

- Reprovision the resource pools

Typically, you reprovision the resource pools when you have made changes to the existing configuration and want the new configuration to take effect or if the previous provisioning had failed.

- Delete the resource pools

Typically, you delete the resource pools when they are no longer used or needed. This action causes the resources allocated to the resource pools to be automatically returned to the global resource pool.



CAUTION: If a reverse proxy service has been successfully provisioned to a resource pool, deprovision the service before deleting the resource pool.

To perform these actions:

1. From the **Manage Resource Pools** page, select the resource pool.
2. On the **Actions** list, select **Manage Resource Pool Associations**. The **Manage Resource Pool Associations** dialog box appears.

This dialog box lists the devices on which the resource pool was provisioned and indicates whether the provisioning was successful. Use the **Search** option to display specific Media Flow Controllers by filtering using their names or tags.



TIP: Provisioning a resource pool to a device is successful only when the device is “In Sync” status. Verify the device status in the **Managed Status** column before provisioning.

3. Select the device and perform one of the following tasks:

- Click **Provision Again** to reprovision the resource pools.

The **Job Information** dialog box appears. Click the job ID to view the job's status or click **Ok** to exit.

- Click **De-provision** to deprovision the resource pools. You are prompted for a confirmation. Click **Yes** to deprovision the resource pools or **No** to exit.



CAUTION: If a reverse proxy service has been successfully provisioned to a resource pool, deprovision the service before deleting the resource pool.

- Click **Cancel** to exit.

**Related
Documentation**

- [Provisioning Services on page 142](#)
- [Provisioning Resource Pools to MFC Devices on page 36](#)
- [Actions on Resource Pools on page 35](#)
- [Creating Resource Pools on page 34](#)
- [Resource Pools Overview on page 33](#)
- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

PART 3

Design Elements

- [Access Log Profiles on page 41](#)
- [Cache-Tuning Policies on page 51](#)
- [Origin Maps on page 67](#)
- [Policy Scripts on page 77](#)
- [Virtual Players on page 81](#)

CHAPTER 4

Access Log Profiles

- [Understanding Access Log Profiles on page 41](#)
- [Creating Access Log Profiles on page 45](#)
- [Actions on Access Log Profiles on page 49](#)

Understanding Access Log Profiles

This topic describes what access log profiles are used for and the information you need to know before creating an access log profile.

Access log profiles enable you to customize access logging and store log files in an external server. You use access logs to record the HTTP or HTTPS transactions handled by Media Flow Controller and capture information, such as the timestamp, remote user, and so on, about the transaction.

Using access log profiles, you can configure:

- Log filename
- Log file size
- Log format
- File rotation interval
- Maximum number of log files to retain
- External server to which the log files must be uploaded
- Criteria for transactions that should not be logged in the access logs, such as:
 - HTTP response codes
 - HTTP delivered object or content size threshold

Before you configure access log profiles, you must consider the following facts:

- You can associate a service with a single log profile only. However, multiple services can share a single log profile. Log records generated by multiple services, but referring to a common log profile, are written to the log file specified by the profile configuration.
- If you do not associate a log profile with a service, the service is automatically associated with the default log profile. The default log profile is created on either Media Flow

Controller fresh installation or Media Flow Controller system upgrade. Previously created log profiles are maintained as is after an upgrade. Because the default log profile is already present, you cannot create a log profile with the name “default” from Media Flow Activate.

- Though MFA supports creating any number of log profiles, the provisioning of the service fails if the log profile associated with the service exceeds the maximum number of log profiles that Media Flow Controller can support (which is 32).
- Whether you have sized the log partition spaces appropriately to accommodate the log files. It is recommended to have at least 64 GB of disk space to store log files.
- Whether you want Media Flow Controller to export log files to an external server (log push) or an external client to import log files from Media Flow Controller (log pull).

After you create an access log profile, associate the profile with relevant services (that you had created previously). This ensures that the HTTP and HTTPS transactions processed by these services are recorded in the log profile according to the log profile configuration.

Default log profile configuration

```
Access Log Profile: default
  Log Filename : access.log
  Max Filesize : 100 MiB
  Max files to hold : 10
  Max time duration : 0 Minutes
  Format Type: w3c-ext-default
  Format : %h - - %t %r %s %b
  Auto Copy URL : -Not Configured-
  Object Size to Skip : 0
  Response Codes to Skip : None
```

Log Push and Log Pull Overview

You can periodically store log files in an external server through the push or pull method. In the push method, Media Flow Controller exports log files to the configured external server by using the FTP, SFTP, or SCP (the default) protocol. In the pull method, an external client pulls log files from Media Flow Controller. It is recommended that you configure only one of the methods rather than both. However, if you configure both log pull and push, the method that was configured last takes precedence.

The access log profile configuration is common to both push and pull methods. Access log files are generated on the basis of this configuration. Typically, an access log file is closed and a new one is opened for writing new log records when the configured rotation interval or maximum file size is met (whichever comes first) or when there is a change to the log record format. If you have configured log push, as soon as a log file is closed, Media Flow Controller exports the closed log file to an external server. In addition, this closed file is immediately moved from the folder in which it was temporarily stored to a permanent folder, which is the LogExport folder.

The names of the log files that are stored in the LogExport folder use the following format: `<hostname>_<filename><profile-name>_<version>_<start-time>_<end-time>.gz`, where:

- *hostname*—Hostname of Media Flow Controller
- *filename*—Name of the log file provided by the user

Media Flow Controller creates the log file with the user-specified filename and it records the log entries in this file. This file is stored in a temporary folder. When the log rotation criteria is met, the file is closed and moved to the LogExport folder.

- *profile-name*—Name of the access log profile
- *version*—Version of the file. When more than one file is generated with the same start and end times, possibly due to a change in log format, the version information helps to differentiate among these files.
- *start-time*—Time when the file is supposed to have been opened for Media Flow Controller to write new records based on the log rotation interval. The start time has the following format: YYYYMMDDhhmm.

If time-based rotation is disabled, the start time represents the time when the log file was created.

- *end-time*—Time when the file should have been closed. A premature file closure may occur when there is a configuration change or a service failure. In both cases of premature closure, the version number is incremented; however, the end time remains the same if the file is valid within the rotation interval. So, the start and end times are computed based on the rotation interval and not on the actual file closure times if time-based rotation is configured. The end time has the following format: YYYYMMDDhhmm.

If time-based rotation is disabled, the end time represents the time when the log file was closed.

Filename example:

`cmbu-vxa20-test_access.log_default_3_201206140030_201206140035.gz`, where “cmbu-vxa20-test” is the hostname, “access.log” is the filename, “default” is the profile-name, “3” is the version number of the log file, “201206140030” represents the start time, and “201206140035” represents the end time.

In the log push method, Media Flow Controller exports the log files to the configured external server when the log rotation criteria are met.

The log pull method allows an external client to connect to Media Flow Controller by using SFTP to import log files from Media Flow Controller. Only LogTransferUser can access and download log files from the LogExport folder. This user is created automatically during Media Flow Controller manufacture or upgrade and has only read access to the LogExport folder. You can configure a password-less access for this user (from the client to Media Flow Controller) by configuring SSH utilities. Perform the steps listed in the “Using SSH in Automated Scripts (CLI)” section of the *Media Flow Controller Administrator's Guide* to generate the SSH public key. After you generate the SSH public key, paste the key in the **Media Flow Devices > Actions** drawer > **Device Configuration >**

User Authentication Key field to configure Media Flow Controller to use this key for LogExportUser from the external client.

Monitoring Access Log Files

Log messages are written to the system log whenever an access log file is:

- Closed and moved to the LogExport folder.
- Deleted due to:
 - Storage space being full
 - File expiry
 - Forced purge

The log messages use the following format: (**<Reason-Code>**)**<free-form message>**:**<filename>**. System log parsers can use **<Reason-Code>** to identify the operation that was performed on the log file.

Table 3: Reason Codes Recorded on System Log for Access Log Files

Reason Code	Description
1000	Log file was closed and moved to the LogExport folder.
1001	Log file was deleted because it expired (the time threshold was exceeded).
1002	Log file was deleted due to storage space constraints.
1003	Log file was deleted because the user initiated a forced purge of the LogExport folder.



CAUTION:

- Files may be missing due to service downtime.
- If the time is changed on the system, files may be replaced or lost.

Related Documentation

- [Creating Access Log Profiles on page 45](#)
- [Actions on Access Log Profiles on page 49](#)
- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Creating Access Log Profiles

You create an access log profile when you want to customize access logs. An access log profile is used to track details of the HTTP and HTTPS transactions handled by Media Flow Controller and store log files in an external server.



CAUTION: Though MFA supports creating any number of log profiles, the provisioning of the service fails if the log profile associated with the service exceeds the maximum number of log profiles that a Media Flow Controller can support (which is 32).

To configure an access log profile on the **Design Elements** workspace:

1. From the left navigation panel, click the plus sign (+) adjacent to **Design Elements**.
2. Click the plus sign (+) adjacent to **Manage Log Profile**.
3. Click **Add Access Log Profile**. The **Add Access Log Profile** page is displayed.
4. On the **Basic Properties** tab, specify the following information:
 - **Log Profile Name**—Enter the name of the access log profile, which must be unique. Log profile names must be defined in 7-bit ASCII, alphanumeric format only.
 - **Description**—(Optional) Enter a description of the access log profile.
For example, "To capture information about the HTTP requests to the YouTube website."
 - In the **Log File Parameters** area, specify the file in which the access logs must be logged and how to handle the log files after a certain time period (log rotation). From this area, configure an external server to which the log files can be exported when the log rotation criterion is met.



NOTE: Typically, log files have a tendency to become voluminous over time. This can pose a problem when you are trying to locate specific information. Log rotation addresses this issue. Log rotation happens when one of the following criteria is met (whichever comes first):

- Rotation interval
- File size

However, rotation parameters are verified only when an activity is written to the log.

You can configure the following fields in this area:

- **File Name**—Enter the name of the file where the access log is stored. The default filename is **access.log**. This file is initially stored in a temporary folder from which it is moved to a permanent folder (which is the LogExport folder) when the file

is closed. Typically, an access log file is closed and a new one is opened for Media Flow Controller to write new log records when the configured rotation interval or maximum file size is met (whichever comes first) or when there is a change to the log record format.

- **Max File Size (MiB)**—Define a size threshold, in MiB, for uploads or log rotation. When the log file reaches the configured size, Media Flow Controller treats this file as closed and opens a new file for Media Flow Controller to write log entries. The closed file is moved to the LogExport folder. If you have configured a server to export this log file (in the **Export Path** field), Media Flow Controller auto-uploads the closed log file to the specified destination.

To disable size-based file rotation, clear the **Max File Size** check box. However, either size-based or time-based rotation must be enabled. You can enter a value from 10 through 100 MiB. The default value is **100** MiB.



NOTE: 1 MiB (mebibyte) is equivalent to 1024x1024 bytes.

- **File Closure Frequency (minute(s))**—Set a file rotation time interval in minutes. The default is 15—that is, after 15 minutes, Media Flow Controller closes the current log file and opens a new file for Media Flow Controller to write log entries. The closed file is moved to the LogExport folder. If you have configured a server to export this log file (in the **Export Path** field), Media Flow Controller auto-uploads the closed log file to the specified destination.

To disable time-based file rotation, clear the **File Closure Frequency** check box. However, either size-based or time-based rotation must be enabled. You can enter a value from 5 through 60 minutes. The default value is **15** minutes.

- **Export Path**—Specify the server to which the access log files must be auto-uploaded after the log rotation criterion is met by using the SCP (the default), FTP, or SFTP protocol.

Use the following format in the **Export Path** field to configure the server:

[<username>]:[<password>]@<hostname>[:<port>]/<path>/.

For SCP and STP, it is mandatory to provide the hostname and path. The path must end with a forward slash. If no folders are provided, the “/” denotes the root folder. For example: **usera:pwdb@hostnameec:8022/youtubefolder/.**

When there is an export failure, a system log message with the profile name and log file name is logged in the system log file, **server.log**, for the respective Media Flow Controller.

If you do not want to export the log files, leave this field empty.

5. In the **Do not Log** area, configure what log records can be discarded. There are no corresponding log entries for these records in the log file.
 - **Object size lesser than (bytes)**—Enter a minimum size for the objects that are retrieved from the Media Flow Controller cache or origin servers, in bytes, for which

a log record is written to the log file. Log entries for objects smaller than or equal to the size specified is not written to the log file.

You can enter a value from 0 through 4,294,967,295 bytes. The default value is 0 byte. A value of zero (the default) means that no filter should be applied and all logs can be written to the log file.

- **Http Response Code**—Add a list of HTTP response codes so that when Media Flow Controller receives these codes as responses from the origin servers, those transactions are not recorded in the log file. To add an HTTP response code, click **Add**. To remove any or all of the response codes from the list, select the codes and click **Remove**.

One of the response codes that you might want to consider filtering out is “206 (Partial Content).” When Media Flow Controller makes a request for an object (such as a large file) from the origin server, the server might serve the object in parts. Each of these responses has a response code of 206 to indicate that the server has fulfilled the partial GET request for Media Flow Controller. When the object has been fully delivered, the origin server sends a “200 OK” response. Here, you might want to filter out all the partial responses by adding **206** in the “Http Response Code” list.

6. On the **Log Format** tab, select a log format for an access log profile from the **Log Record Format Type** list. This format essentially determines what kind of information is logged in a log file for an HTTP or HTTPS transaction. The supported format types are: **W3C Ext Default** (which is the default format), **CLF**, **NCSA-COMBINED**, and **Custom Record Format**.

When you select a format from the **Log Record Format Type** list, the supported format parameters and the associated format strings are displayed in a table format just below the list. For example, when you select the CLF log format, you can view the information that is captured in the log file, such as the remote host, remote user, and so on.

Only Custom Record Format is configurable. All other log formats are not configurable—that is, you cannot choose the format strings to associate with the specific log format.

To configure **Custom Record Format**:

- a. From the **Log Record Format Type** list, select **Custom Record Format**.
- b. Click **Add/Edit**. The **Select Log Format Strings** dialog box appears.
- c. In the **Dynamic Format String – Header** area, configure any valid request-header, response-header, cookie, and user comment to the format string. These are custom fields that enable you to log any of the request or response headers present in the HTTP message.

For example, if you want to log the cache control header present in both request and response headers, enter **Cache-Control** in the **Request Header Match** and **Response Header Match** fields and then click **Add**.

- d. In the **Available** pane, double-click the format strings for which information must be logged in the log file. Each double-clicked format string moves to the **Selected** pane.
 - e. Click **OK**. You are returned to the previous page.
7. Click **OK**, **Cancel**, or **Reset**. **OK** instantiates the values you set, **Cancel** closes the configuration page, and **Reset** returns all values to their defaults.

You are returned to the **Manage Log Profile** page. If you have successfully created a log profile, you can view the newly added log profile on this page. This page displays the following information:

- Log profile name
- Log file in which the logs are recorded
- Log format type, which determines the information captured in the log file
- User who created the log profile
- Last user who modified the log profile
- Timestamp when the last modification was made

You need to associate this log profile with a service so that details regarding the HTTP or HTTPS request to that service (namespace) are logged in the corresponding log file, which can then be analyzed, if needed. For more information about associating a log profile with a service, see [“Creating HTTP Reverse Proxy Services” on page 112](#).



CAUTION: No disk space checks are performed when a profile is created. On a VXA2010 device, a 330-GB partition is set aside for logging. (For pacifica, no store exists.) For instance, if two profiles are created with each profile limiting a file size to 10 GB and with 30 files to retain, this results in a total expected size of 600 GB. Such checks are not performed. This exceeds the log partition size and might cause log files to be overwritten or lost. It is recommended that administrators export log files to an archiving device at regular intervals.

**Related
Documentation**

- [Actions on Access Log Profiles on page 49](#)
- [Understanding Access Log Profiles on page 41](#)
- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Actions on Access Log Profiles

From the **Manage Log Profile** page, you can perform the following actions on access log profiles by clicking the links on the **Actions** list. You have to select the access log profiles before performing any actions on them:

- **Show Log Profile Details**—Click this link to view the configuration of the selected access log profile. In the pop-up, you can sort the data in the **Log Record Format** table and select what columns you want to display by:
 - Mousing over a column and clicking the list
 - Selecting **Sort Ascending** or **Sort Descending** to sort the data in ascending or descending order
 - Selecting **Columns** and choosing the columns to display
- **Modify Log Profile**—Other than the profile name, you can modify all other configuration settings. After you save the changes, the revised configuration is reflected in all the services that use this profile. You can modify only one profile at a time.

Changing log format causes, irrespective of any configured thresholds (rotation interval or maximum file size), the closure of the current log file. A new file is opened and all the log records are logged in this file.

- **Delete Log Profile(s)**—Delete one or more access log profiles.

If the access log profile is associated with a service, you must make sure that the service is no longer provisioned to a device before deleting the access log profile.



CAUTION: Each profile, upon creation, uses some storage to save the log files. When a profile is deleted, the log files are deleted and the storage space is reclaimed. The administrator is responsible for ensuring that all log files under a profile are backed up or exported to external storage before deleting that profile. A deleted profile cannot be recovered, and all log files stored in it are lost.

Related Documentation

- [Understanding Access Log Profiles on page 41](#)
- [Creating Access Log Profiles on page 45](#)
- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

CHAPTER 5

Cache-Tuning Policies

- [Understanding Cache-Tuning Policies on page 51](#)
- [Creating Cache-Tuning Policies on page 52](#)
- [Actions on Cache-Tuning Policies on page 65](#)

Understanding Cache-Tuning Policies

This topic describes what cache-tuning policies are used for and the information you need to know before creating a cache-tuning policy.

The **Cache Tuning Policy** function allows you to set cache-handling parameters. Before configuring cache-tuning policies, you must know details about your delivery environment and desired caching behavior, including:

- How long you want the content for a specific service to be held in cache, and the threshold parameters for determining this.
- What objects you want to be cached. The Media Flow Controller does not cache certain objects by default, such as objects with query strings or cookies.



NOTE: Media Flow Controller uses the concept of “hotness” (popularity) to determine when to promote an object to various cache tiers. Tiers comprise RAM, Solid-state drive (SSD), Serial Attached SCSI (SAS), and Serial ATA (SATA). The hotness computation is based on the hit frequency (that is, the number of hits as a function of time). For example, an object with 400 hits an hour is considered hotter than an object with 500 hits in one day.

See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about cache-tuning policies.

Related Documentation

- [Creating Cache-Tuning Policies on page 52](#)
- [Actions on Cache-Tuning Policies on page 65](#)
- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)

- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Creating Cache-Tuning Policies

The **Cache Tuning Policy** workspace enables you to set cache-handling parameters.

Before configuring cache-tuning policies, you must know details about your delivery environment and desired caching behavior, including:

- How long you want the content for a specific service to be held in cache, and the threshold parameters for determining this.
- What objects you do not want excluded from caching. The Media Flow Controller does not cache certain objects by default, such as objects with query strings or cookies.

You apply your cache-tuning policy to websites you create for delivery of different media. After the policy is created, you can apply it to any number of websites.

To configure cache-tuning policies on the **Design Elements** workspace:

1. Click the plus sign (+) adjacent to **Design Elements**.
2. Click the plus sign (+) adjacent to **Manage Cache Tuning Policy**.
3. Click **Add Policy**. The **Add Policy** page is displayed.
4. On the **General** tab, specify the following information:
 - **Policy Name**—Enter the name of the cache-tuning policy, which must be unique.
 - **Description**—(Optional) Enter the description of the cache-tuning policy.
 - In the **Common Settings** area, for **Cache fill**, select one of the following options:
 - **Client Driven**—Allow Media Flow Controllers to fetch only as much data as the client requested. Media Flow Controllers stop downloading an object from the origin server after fetching the amount of data requested by the client, or the client stops receiving or viewing it.



CAUTION: It is not recommended to configure this option if the origin server is known to not support byte-range requests. Instead, select the **Aggressive** option. This is because when Media Flow Controller receives a second request for the same object, it delivers the partial file available from the cache to the client and makes a byte-range request to the origin server for the remaining bytes. However, if the origin server does not support byte-range requests, it responds with “200 OK” for the byte-range request and the object delivery is stopped. In addition, the partial file is not deleted from the cache.

- **Aggressive threshold**—Configure this option only if you have configured Media Flow Controller to cache objects in client-driven mode. Using this option, you can specify the hotness threshold of an object above which Media Flow Controller switches from client-driven mode to controlled aggressive mode. That is, in case of byte-range requests, when the hotness of the object becomes greater than or equal to the configured **Aggressive threshold** value, Media Flow Controller automatically fetches additional data from the origin server and caches it, even without a client request. The additional data that is fetched is 2 MB, if data is cached in Serial ATA (SATA) or Serial Attached SCSI (SAS) disk; otherwise, the data fetched is 256 KB if data is cached in SSD disk. Note that even if the byte range is set to something like 512 KB in the client request, Media Flow Controller still fetches the next 2 MB from the origin server (for SAS and SATA disks) if **Aggressive threshold** is configured and the hotness of the object is greater than or equal to this configured value.

You can set the value from 0 through 100 (when Media Flow Controller is configured in client-driven mode, a value of 0 [zero] for **Aggressive threshold** means that this feature is disabled and Media Flow Controller is purely client-driven). The default value is **9**.

Recommendations: Set this value to 0 (zero) for Network Optimization (transparent proxy) service deployments because you may not want Media Flow Controller to fetch any additional content other than what the client requested.

Example: Consider a client requesting a 10-MB object in byte-range requests of 1 MB. If there is a cache miss, Media Flow Controller forwards the same byte-range request to the origin server, and the response is cached and served by Media Flow Controller. Consider that the client had sent three byte-range requests. In client-driven mode, Media Flow Controller would have fetched only as much data as requested by the client and hence would have cached only 3 MB till now. However, if **Aggressive threshold** is configured, when the hotness of the object becomes greater than or equal to this configured value, Media Flow Controller automatically fetches the next 2 MB of the object and caches it without any client request. This is because Media Flow Controller anticipates that the client would request the next chunk of the object when the hotness of the object reaches the configured **Aggressive threshold** value. In this example, even though the client had requested 3 MB of the object, Media Flow Controller caches 5 MB of the object.

- **Aggressive**—Allow Media Flow Controllers to fetch the full object irrespective of the amount of data that the client requested. This configuration proves useful for services that serve popular objects that are large (such as videos, installation packages, and PDF files).

Example: Consider a client requesting a 10-MB object in byte-range requests of 1 MB. After Media Flow Controller has served the first byte-range request of 1 MB to the client, it automatically sends another byte-range request to the origin server to fetch the entire object (that is, 1 MB plus 1 byte to the end of the file). Media Flow Controller caches the remaining 9 MB of the object to serve future client requests.

- **Exclude Domain Name from Cache Index**—Select this check box if you want Media Flow Controller to exclude the domain name from the cache index when it creates the cache index for an object.

Media Flow Controller associates a cache index with each object that is cached, so that when a request for an object is received, Media Flow Controller uses the cache index to determine whether the object is available in the cache or not before fetching the object from the origin server. Therefore, in scenarios where multiple domains deliver the same object, including the domain name in the cache index can result in unnecessary cache miss and would require fetching the same object from the origin server again. Therefore, excluding the domain name from the cache index improves media delivery throughput.

- **Set X-Forwarded-For header**—Select this check box if you want Media Flow Controller to include the X-Forwarded-For header in the client request while it forwards this request to the origin server (due to a cache miss).

When the client requests an object, the request may or may not contain the X-Forwarded-For header:

- If the client request does not include an X-Forwarded-For header, Media Flow Controller adds this header with the IP address of the client (which sent the request to Media Flow Controller) while forwarding this request to the origin server.
- If the client request includes an X-Forwarded-For header with some value, then Media Flow Controller appends the client's IP address at the end of the existing value and forwards the request to the origin server.

Typically, the information in this header enables the origin server to track the systems through which the request has traversed by analyzing the IP addresses captured in this header (starting from the client, proxy1 [which could be a Media Flow Controller server], and so on).

In transparent proxy deployments, we recommend that you exclude this header to achieve transparency.

5. On the **Cache tier** tab, specify the following information:

Figure 1: Add Policy—Cache tier Tab

Add Policy

General **Cache tier** Expiry & revalidation Tunnel override decision

Cache-age Threshold: 60 secs

Object size based threshold settings

Min Object Size Threshold: 0 KB Max Object Size Threshold: 0 KB

Disk Ingest Threshold: 4096 bytes Fast Ingest Threshold: 0 bytes

☒ **Disk cache settings**

Cache Tier	Free block threshold(%)	Group read	Read size(in Kbytes)
SAS	50	Enabled	2048
SATA	50	Enabled	2048
SSD	50	Disabled	256

Cache ingest hotness threshold: 3 requests

URI depth threshold: 10

OK Cancel Reset

- **Cache-age Threshold**—Enter the time in seconds so that any object whose expiry time is less than this value is cached only in the Media Flow Controller RAM cache. Media Flow Controller, by default, caches objects with an expiry time of less than 60 seconds in the RAM cache. The expectation is that these objects may be modified very often in the origin server and hence are not worth storing in disk caches (because this would otherwise waste disk I/O operations).
- In the **Object size based threshold settings** area, **Min Object Size Threshold** and **Max Object Size Threshold**—Objects of sizes greater than or equal to the minimum size threshold and lesser than or equal to the maximum size threshold that you set are cacheable; objects smaller or larger than those sizes are not. Instead, Media Flow Controller tunnels those objects.

You can enter a value from 0 through 4,29,49,67,295 KB. The default minimum and maximum object size threshold is 0 KB.

- **Disk Ingest Threshold**—Enter a size limit, in bytes, for storing objects in the disk cache. The default is 4096. For example, a value of 4 means you can store all fetched objects larger than or equal to four bytes in the disk cache. 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 smaller than this threshold, it is cached and served from the Media Flow Controller RAM cache.

Setting a threshold can improve disk-cache performance because small objects need not be written to disk and can be cached and served directly from the RAM cache.

- **Fast Ingest Threshold**—Enter the maximum size of an object that has to be ingested or cached into the fastest cache tier in the disk cache. Objects smaller than or equal

to the configured size and greater than or equal to the value configured in "Disk Ingest Threshold" are automatically written to the fastest cache tier. The default is 0 (zero), which means that no objects are directly promoted to the fastest tier. The maximum allowed value is **4,294,967,295** (4 GB).

- In the **Disk cache settings** area, enable disk read options per cache tier. Media Flow Controller supports SAS, SATA, and SSD types of disk cache tiers and is capable of detecting the disk types. Media Flow Controller organizes the disk into 2-MB blocks if the disk type is SAS or SATA. If the disk type is SSD, then Media Flow Controller organizes the disk into 256-KB blocks. Therefore, one disk block is of size 2 MB or 256 KB depending on the disk type.

You can configure the following for each type of cache tier:

- **Free block threshold(%)**—Enter a value so that Media Flow Controller deletes all the objects in a block when the usage of the block falls below the configured percentage value when an object is deleted from the block.

When you delete an object from Media Flow Controller, if the usage of the block from which the object has been deleted is less than or equal to the configured **Free block threshold(%)** value, then Media Flow Controller deletes the remaining objects from that block to reclaim the complete block and adds this block to the list of free blocks so that the block can be used for the next caching operation. However, if the usage of the block from which the object is deleted is greater than the configured **Free block threshold(%)** value, Media Flow Controller does not delete the remaining objects from that block.

The default value is **50%**. That is, when the occupancy of a block falls below 50%, Media Flow Controller deletes the remaining objects within that block to free the entire block so that it can be used for the next caching operation.

You can enter a value from 0 through 100%. A value of zero means that even if objects are deleted from a disk block, Media Flow Controller on its own does not delete the remaining objects from that block. However, a value of 100% means that even if one of the objects is deleted from a disk block, Media Flow Controller deletes the rest of the objects from that block and reclaims the entire block for the next caching operation.

Example (for SAS or SATA disk): Consider a block containing two objects of sizes 1.5 MB and 0.5 MB. If the **Free block threshold(%)** value is set at 50% and you delete the 1.5-MB object from the disk cache, then the disk block usage falls to 25%, which is less than the configured value. In this case, Media Flow Controller automatically deletes the remaining 0.5-MB object from its disk cache and reclaims the entire 2-MB block for the next caching operation. Now, if the client requests for the deleted 0.5-MB object, Media Flow Controller fetches this object from the origin server and caches and serves this object. However, instead of deleting the 1.5-MB object, if you delete the 0.5-MB object, the disk block usage is 75%, which is more than the configured value of 50%. In this case, Media Flow Controller does not automatically delete the 1.5-MB object.

- **Group read**—Enable or disable reading of all the objects from a disk block to the RAM cache.

When you enable **Group read**, if the client requests an object that is not available in the RAM cache but is available in the disk cache, Media Flow Controller reads the entire 2-MB block (in case of SAS or SATA) into the RAM cache instead of reading the specific object from the disk cache. This helps in reducing the number of disk reads. For example, if **Group read** is enabled in the SAS or SATA disk and the client requests a 32-KB file, then Media Flow Controller reads the entire 2-MB block into the RAM cache but delivers only the requested 32-KB file to the client. If **Group read** is disabled, then Media Flow Controller reads only the specific file (in this case, the 32-KB file) from the disk cache into the RAM cache and serves that file to the client.

The default value is **Enabled** for SAS and SATA disks; **Disabled** for an SSD disk.

Recommendations: Enable for HTTP reverse proxy service deployments or adaptive bit-rate streaming caching. Disable for Network Optimization (transparent proxy) service deployments.

Example: When you use the adaptive bit-rate streaming for videos, you may cache a video in chunks within a block in Media Flow Controller. When the user requests the first chunk of the video, if you have enabled **Group read**, Media Flow Controller automatically reads the remaining chunks of the video within the block (assuming that the remaining chunks are cached within the same block) and caches them into the RAM cache. So, when the client requests the next chunk of the video, the chunk is served immediately from the RAM cache.

- **Read size(in Kbytes)**—Enter the size of data that Media Flow Controller must read from the disk cache into the RAM cache, in a single disk read.

The default value is **2048 KB** (2 MB) for SAS and SATA disks; **256 KB** for an SSD disk.

The range of values that you can enter are:

- SATA disk—2048 KB (default) or 256 KB. No intermediate values are permitted—that is, the read size is either 2048 KB or 256 KB.
 - SAS disk—2048 KB (default) or 256 KB. No intermediate values are permitted—that is, the read size is either 2048 KB or 256 KB.
 - SSD disk—256 KB (default) or 32 KB. No intermediate values are permitted—that is, the read size is either 256 KB or 32 KB.
- **Cache ingest hotness threshold**—Enter a value, which defines the minimum object hotness required to ingest or promote an object to a disk cache tier.

You can enter a value from 3 through 65,535. The default value is **3**.

Example: When a client requests an object for the first time, Media Flow Controller sets the hotness of the object to 3. If the **Cache ingest hotness threshold** value is left at its default value of 3, then after only the first request, the hotness of the object matches the hotness threshold, which makes the object eligible for ingestion into the lowest disk cache tier. If Media Flow Controller contains SATA, SAS, and SSD disk cache tiers, then this object is ingested into the SATA disk cache tier, which is the lowest disk cache tier. If the **Cache ingest hotness threshold** value is set to 6,

then the object is served from the RAM cache until the hotness of the object reaches 6. When the hotness of the object reaches 6, the object is ingested into the lowest disk cache tier.

Media Flow Controller increments or decrements the hotness of an object on the basis of how frequently or infrequently the object is requested. For example, if the object is requested a second time within the system average interval, then Media Flow Controller increments the hotness of the object to 6 (that is, $2 * \text{the Cache ingest hotness threshold value}$). When the hotness of the object reaches 6, the object is automatically promoted to the next fastest disk cache tier, which is the SAS disk cache tier. When the hotness of the object reaches 18 (that is, $6 * \text{the Cache ingest hotness threshold value}$), the object is ingested into the fastest disk cache tier, which is SSD. The next request for the object is served from the SSD disk cache tier.

To summarize, Media Flow Controller automatically promotes popular or “hot” objects (that is, the most requested objects) to a faster cache tier. “Cold” objects (that is, the least requested objects) remain in slower cache tiers.

- **URI depth threshold**—Enter the depth of the directories that can be cached in the Media Flow Controller disk cache. Using this configuration, you limit the number of directory levels that are created in the Media Flow Controller’s disk cache while it caches an object, thereby preventing Media Flow Controller from caching objects with long URLs. Objects with long URLs are cached only in the RAM cache resulting in decreased latency as compared to serving them from the disk cache.

If the client request contains M directories in the URL (including the first slash) and **URI depth threshold** is set to a value N, where M is less than or equal to N, then Media Flow Controller caches the origin server response for this request in the disk cache. If M is greater than N, then Media Flow Controller caches the response only in its RAM cache and not in its disk cache.

Similarly, if you have configured a crawler with the base URL containing M directories and the link depth is set to N, then for Media Flow Controller to cache the crawled objects in its disk cache, you have to make sure that the **URI depth threshold** value is greater than or equal to $M+N+1$.

You can enter a value from 0 through 20. The default value is 10.

Example 1: Consider that a client is requesting the `/videos/flv/sample.flv` object and that Media Flow Controller **URI depth threshold** is set to 10. Media Flow Controller caches this object in its disk cache because the directory depth of the object is 3, which is less than the **URI depth threshold** value of 10. Directory depth is calculated on the basis of the number of slashes in the URL, starting from the first slash.

If you modify the **URI depth threshold** to 3 and if a client requests the `/data/videos/flv/sample.flv` object, then this object is cached only in Media Flow Controller’s RAM cache and not in its disk cache because the directory depth of the object is 4, which is greater than the **URI depth threshold** value of 3.

Example 2: Consider that you have configured a crawler with the base URL as “/a/b” and link depth as 10. In addition, the **URI depth threshold** value is left at its default value of 10. Now, if the crawler tries to cache an object with an HTTP URL of

`/a/b/1/2/3/4/5/6/7/8/9/10/sample.txt`, then this object is cached only in Media Flow Controller's RAM cache and not in its disk cache because the object is considered to be under a directory depth of 13, which is greater than the **URI depth threshold** value of 10.

6. On the **Expiry & revalidation** tab, specify the following information:

Figure 2: Add Policy—Expiry & revalidation Tab

- In the **Cache Age Settings** area, **Default Cache Age**—Enter a cache age value in seconds, which is used as the expiry time if the origin server did not send any expiry time through the Expires header or the Cache-Control header's max-age directive while serving the object.

You can enter a value from 0 through 9,46,72,800 seconds. The default value is 0 seconds, which means that the origin response is tunneled to the client.

- In the **Cache Age Override** area, set expiry policies on the basis of the content type. You can set the expiry time (Max-Age) for the cached content on the basis of its type, beyond which the content type is considered to be expired and undergoes a revalidation. Revalidation involves determining whether the content has been modified in the origin server or not. If the content has been modified, then the existing

cached content in Media Flow Controller is deleted and the modified content is fetched from the origin server and cached in Media Flow Controller.

- To set Max-Age for a specific content type, select **Content Type**, add the content type, and enter the Max-Age value. Repeat the process to add multiple content types.
- To set Max-Age for any content type, select **Any Content** and set the Max-Age value.
- You can also select both **Any Content** and **Content Type**. For example, if you set 2880 seconds for the **application/flv** content type and 900 seconds for any content type, any flv content is revalidated after 2880 seconds, whereas for all other content types, the content is revalidated after 900 seconds.
- In the **Allow revalidation** area, specify whether Media Flow Controller must revalidate the object in the cache when it expires or when the object is close to expiry. Media Flow Controller revalidates the object with the origin when the content is close to its expiry (10 percent of life left) due to a preset cache age. This revalidation action is triggered by a client requesting the object. Set this configuration to minimize transit bandwidth usage and to improve cache-hit ratio.

When a client requests an object, Media Flow Controller performs a cache lookup. If there is a cache hit, Media Flow Controller further checks whether the object is expired or not before serving the object to the client. If the object is expired, Media Flow Controller sends a revalidation request to the origin server. The request contains the if-modified-since header, which contains the value of the last-modified time sent by the origin server at the time of caching. The origin server checks the modified time of the actual file against the last-modified time. If the modified time is earlier than or equal to the last-modified time, the origin server sends a “304 Not Modified” response to Media Flow Controller and Media Flow Controller serves the object from its cache. However, if the content has been modified, the server sends a “200 OK” response along with the modified content. Media Flow Controller then replaces the expired content with the modified content in its cache and serves the updated content to the client.

- **Revalidation method**—Select either the **HEAD** or **GET** method for revalidation.

HEAD revalidation requests are more efficient than GET revalidation requests; however, some content websites do not support HEAD requests.

- **Headers for revalidate request**—Select one of the following headers to revalidate: **Last-Modified**, **Etag**, or **Others**. When this is configured, Media Flow Controller compares this header's value in the “200 OK” response sent by the origin server (in response to the Media Flow Controller's revalidation request) against the value in its expired cached object to determine whether the object has been modified in the origin server or not.

Typically, when Media Flow Controller sends a revalidation request to the origin server, the server responds with a “304 Not Modified” or “200 OK” response. Upon receiving a 304 response, Media Flow Controller assumes that the content has not been modified and serves the object from the cache. For a 200 OK response, Media Flow Controller deletes the content from the cache and replaces it with

the modified content sent by the origin server along with the 200 OK response. However, some origin servers that do not support revalidation requests may send the 200 OK response even if the content has not been modified. In such scenarios, you may want Media Flow Controller to perform additional validation whenever it receives a 200 OK response. This can be done by configuring Media Flow Controller to compare the value of a specific header in the 200 OK response against its cached version. If the values are different, then Media Flow Controller caches the new content by deleting the existing content. The headers that you can select to validate are: **Last-Modified**, **Etag**, or **Others**. When you want to validate with a custom header, select **Others** and enter the header name in the text box.

- **Use date header when last-modified is absent**—When you select this check box, Media Flow Controller uses the date header information for revalidation if the last-modified information is missing from the origin server response.

Typically, when Media Flow Controller sends a revalidation request to the origin server, the request contains the if-modified-since header, which contains the value of the last-modified time sent by the origin server at the time of caching. However, if the last-modified information was not sent by the origin server at the time of caching and this feature is enabled, Media Flow Controller sends the date header information in the if-modified-since header in its revalidation request.

- **Flush caches (triggers revalidation across servers)**—Select this check box when you want Media Flow Controller to revalidate its cache entry with the origin server (and not only with the next cache along the path to the origin server) or to reload its cache entry from the origin server. When you enable this feature, end-to-end revalidation occurs irrespective of how many proxies exist between Media Flow Controller and the origin server.

Media Flow Controller sends the revalidation request with "Cache-Control: max-age=0," which ensures that each cache along the way revalidates its cache entry all the way to the origin server.

- In the **Revalidation override** area, configure Media Flow Controller to override the max-age value in the client request's Cache-Control header, so that the request is served from its cache, effectively ignoring the max-age value in the client request.

Select **Override Max-age header** and enter a value in seconds in the **Max-age override threshold** field so that Media Flow Controller serves the requested object from its cache if the max-age value in the incoming request is less than or equal to the configured value and the object is not expired in the cache.

You can enter a value from 0 through 4,294,967,295 seconds. The default value is 0 seconds.

Example: When Media Flow Controller receives a client request with the "cache-control:max-age=0" header, it sends a revalidation request to the origin server irrespective of whether the cached object is expired or not. If the origin server responds with a "304 Not Modified" response, then Media Flow Controller serves the object from its cache. However, if the origin server responds with a "200 OK" response, then Media Flow Controller deletes the existing cached content irrespective of whether it is expired or not, and caches and serves the newly fetched content.

This is the default behavior when **Max-age override threshold** is left at its default value of 0 (zero).

Download managers (that are available as plug-ins to the browser for downloading objects) typically have a tendency to add the “cache-control:max-age=0” header in all their requests for downloading objects, thereby ensuring that a revalidation of the requested object occurs. If the object has been modified in the origin server, then the modified object is fetched and served to the download manager. Often, this results in the following situations:

- Too many revalidation requests are sent to the origin servers.
- Some origin servers that do not support revalidation requests always send a 200 OK response, thereby forcing Media Flow Controller to delete the existing cached object even if the object is not expired or modified, and cache and serve the newly fetched content.

To overcome such situations, set the **Max-age override threshold** value to a value greater than 0 (zero).

To summarize, when a client request contains the “cache-control:max-age=N” header, Media Flow Controller serves the requested object from its cache if N is less than or equal to the **Max-age override threshold** value. However, if the object is expired, then Media Flow Controller compulsorily performs a revalidation irrespective of the configured value to prevent serving stale content from its cache.

7. On the **Tunnel override decision** tab, specify when the Media Flow Controller should override normal tunneling behavior.

Figure 3: Add Policy—Tunnel override decision Tab

Add Policy

General Cache tier Expiry & revalidation **Tunnel override decision**

☒ **Client Request**

☒ Cache Request with query string Select...

☒ Cache Request with specific headers

☐ Auth-header

☐ Cookie-header

☐ Cache-control-header

☒ **Origin Response**

No cache directive: Follow

Response with HTTP 302 Status Code: Tunnel the response

☐ Cache expired objects

☒ Cache objects with cookies ⓘ Select...

☐ Ignore no-transform header

OK Cancel Reset

- Select the **Client Request** check box and select **Cache Request with query string** to cache objects with a query string (such that objects are typically dynamic and often not appropriate for caching). If you do not select **Cache Request with query string** (the default), objects with query string are not cached. If you select **Cache Request with query string**, also select **Strip Query String** (do not include the query string portion of the URL in the cache index) or **Include Query String** (the default).
- Select **Cache Request with specific headers** to override tunneling of objects when specific headers are present in the client requests.

Typically, Media Flow Controller does not cache (or tunnel) objects when the client request contains an **auth-header**, **cookie-header**, or a **cache-control** header. However, select **Auth-header**, **Cookie-header**, **Cache-control-header**, or a combination of these headers for Media Flow Controller to cache the origin server responses to client requests containing any of these headers.

Example for Auth-header: If you have configured to cache the responses for client requests containing the authorization header (by selecting **Auth-header**), then if this is the first request from the client for a specific object, Media Flow Controller forwards this request to the origin server. Origin server authenticates the client and if the authentication is successful, sends the requested object, which is then cached by Media Flow Controller and served to the client. Consider that another client requests the same object but with a different authorization header value in its request. Media Flow Controller continues to serve the object from its cache (if the object has not expired). After the object is cached, further authentication is not performed and all subsequent requests are served from cache only until the object expires.

Example for Cache-control-header: When Media Flow Controller receives a client request containing a cache-control header with the “no-cache” value, by default, Media Flow Controller tunnels the request directly to the origin server. When the request itself is tunneled, the response from the origin server for that request is also tunneled back to the client without being cached in Media Flow Controller. However, by selecting **Cache-control-header**, you can cache the responses to such requests. (Here, Media Flow Controller places such requests in the cacheable path instead of the tunneling path.) Any subsequent requests for these objects are served from the cache.

- In the **Origin Response** area, you can specify:
 - **No cache directive**—Specify what you want Media Flow Controller to do with the directive specified in the Cache-Control headers (for example, “no-cache”, “max-age=0”, and “private”) of the HTTP responses sent by the origin servers.
 - **Follow**—Do not cache the object when the origin server does not want you to cache the object. Media Flow Controller tunnels the origin server response to the client (without caching the object) when the origin server response contains “Cache-Control: private” or “Cache-Control: no-store.” This is the default.

However, if the origin server responds with a “Cache-Control: no-cache” or “Cache-Control: max-age=0,” Media Flow Controller caches the response to the first request without tunneling the response. Before serving any subsequent request, Media Flow Controller compulsorily performs a revalidation. Depending

on the origin server's response, if the object has not been modified, Media Flow Controller updates only the expiry time of the object; otherwise, Media Flow Controller deletes the existing cached object and caches the latest modified object.

- **Override**—By default, Media Flow Controller does not cache objects that are marked “private” or “no-store.” However, a service provider who wants to selectively override the Cache-Control header directive and force Media Flow Controller to cache objects served with these Cache-Control directives to achieve better bandwidth savings can select this value.
- **Tunnel**—Media Flow Controller tunnels the origin response to the client if the Cache-Control directive in the origin response is set to non-cacheable values (such as no-cache, no-store, private, and so on) or max-age=0, or both.
- **Response with HTTP 302 Status Code**—Specify what Media Flow Controller should do when it receives an HTTP 302 response code from the origin server:
 - **Tunnel the response**—(Default) Tunnel the 302 response from the origin server directly to the client without caching the response.
 - **Handle the response**—Send a request to the new URL specified in the “Location” header in the 302 response.

If Media Flow Controller has to revalidate an object, it first tries to revalidate the object with the original origin server. If the original origin server continues to respond with a 302 response, then Media Flow Controller revalidates the object with the new origin server by using the “Location” header in the original origin server's 302 response.

To avoid endless loops due to misconfigured redirects at origin servers, Media Flow Controller supports a maximum of five redirects per request per client. When the number of redirect messages handled by Media Flow Controller for a request exceeds five, Media Flow Controller responds to the client with the last 302 redirect message.

Recommendations: For HTTP reverse proxy service deployments, you may want Media Flow Controller to handle the 302 response and forward the request to the new origin server by using the “Location” header in the 302 response. However, for Network Optimization service deployments, it is recommended that you tunnel the 302 response.

- **Cache expired objects**—Normally, when the origin server responds with a “Cache Control:max-age=0,” Media Flow Controller treats the object as expired and tunnels the response without caching it. Select the check box to override this behavior. Then Media Flow Controller caches the expired object and sets the expiry time of the object to the default cache-age value, or the max-age value configured for a specific content-type or content-type-any. The precedence is as follows: max-age value of specific content-type, then content-type-any, and finally, cache-age-default. Media Flow Controller treats the object as valid for the default cache-age duration and serves the object from its cache during this time.

- **Cache objects with cookies**—Select the check box to cache objects with cookies returned by the origin server. If you do not select this check box (default), objects with cookies are not cached. These objects are associated with a particular client session and often not appropriate for caching. When you select this check box, you can also specify whether such cached objects with cookies can be served directly from the cache when a subsequent request comes in (**Do not validate**) or a validation is required from the origin server before the objects with cookies are served (**Validate with Origin**, which is the default).
- **Ignore no-transform header**—Select the check box to cache the origin server response with the “Cache Control: No-Transform” header.

An origin server might send a “Cache Control: No-Transform” header in its response when it does not want the intermediate proxies to change any aspect of the object before it is served to the client. Usually, Media Flow Controller tunnels such responses directly to the client without caching them. However, if you want to cache these responses, you have to enable this feature.

8. Click **Ok**, **Cancel**, or **Reset**. **Ok** instantiates the values you set, **Cancel** closes the **Add Policy** configuration page, and **Reset** returns all values to their defaults.



NOTE: See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about cache-tuning policies.

Related Documentation

- [Understanding Cache-Tuning Policies on page 51](#)
- [Actions on Cache-Tuning Policies on page 65](#)
- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Actions on Cache-Tuning Policies

From the **Manage Cache Tuning Policies** page, you can perform the following actions on cache-tuning policies by clicking the links on the **Actions** list. You have to select the cache-tuning policies before performing any actions on them:

- **Modify Policy**—Click this link to modify the configuration settings other than the cache-tuning policy name. After the changes are saved, the revised configuration is reflected in all the services that use this cache-tuning policy.
- **Delete Policy(s)**—Click this link to delete one or more cache-tuning policies.

Related Documentation

- [Understanding Cache-Tuning Policies on page 51](#)
- [Creating Cache-Tuning Policies on page 52](#)

- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

CHAPTER 6

Origin Maps

- [Understanding Origin Maps on page 68](#)
- [Creating Consistent Hash Maps on page 69](#)
- [Creating Escalation Maps on page 72](#)
- [Actions on Origin Maps on page 75](#)

Understanding Origin Maps

This topic describes what origin maps are used for and the information you need to know before creating an origin map. In this topic, the term “origin map” encompasses both “consistent hash map” and “escalation map.”

The **consistent hash map** feature enables you to group a number of nodes (Media Flow Controller and non-Media Flow Controller nodes) for load balancing and increase the cache-storage capacity. The incoming requests are distributed among the configured cluster of nodes. A consistent hashing scheme is used to bind the objects to the nodes and any incoming request is directed to the node that stores the requested content. The cached content is uniformly distributed among the caches. From the deployment perspective, consider consistent hash mapping when you want to deploy a number of Media Flow Controllers as mid-tier proxies—for example, when you want to cache content from a number of edge caches. Here, the cluster of nodes in the consistent hash map provides the required storage capacity and load balancing.



NOTE: A master copy of the content, which is the origin server, is required in case the consistent hash map feature fails.

Using escalation maps, you can configure multiple redundant HTTP origin servers for failover protection. If the target origin server fails or returns a configured HTTP code requiring escalation (for example, HTTP 404), another configured origin server is automatically chosen to handle the incoming request. The requests are sequentially initiated to configured origin servers (on the basis of the order in which they are displayed in the UI, with the topmost origin server having the highest priority) until the request is satisfied or all known available origin servers at request-initiation time have been tried. Typically, from the deployment perspective, the first node is configured as the Content Delivery Network (CDN) provider's node, whereas the node that is marked for escalation is the origin server itself, such as cnn.com's server. Here, you use the content provider's server to mitigate any issues in the caching server.



NOTE: See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about consistent hash maps and escalation maps.

Related Documentation

- [Creating Consistent Hash Maps on page 69](#)
- [Creating Escalation Maps on page 72](#)
- [Actions on Origin Maps on page 75](#)
- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Creating Consistent Hash Maps

The consistent hash map feature enables you to create a cluster of nodes (Media Flow Controller or non-Media Flow Controller nodes) to distribute incoming requests across these nodes and increase the cache-storage capacity. For example, consider a Content Delivery Network (CDN) provider that uses a set of four origin servers grouped as a consistent hash map to cache content from cnn.com. The cnn.com's content is spread across these four nodes and any incoming request is directed to the node containing the requested object by using the consistent hashing scheme.

Before configuring consistent hash maps, you must consider:

- The list of nodes you want to group
- The parameters you want to use for monitoring the nodes to know whether they are available or not
- Whether the nodes are configured as reverse proxy servers

To configure a consistent hash map on the **Design Elements** workspace:

1. From the left navigation panel, click the plus sign (+) adjacent to **Design Elements**.
2. Click the plus sign (+) adjacent to **Manage Origin Map**.
3. Click **Add Consistent Hash Map**. The **Add Consistent Hash Map** page is displayed.
4. On the **Basic Properties** tab, enter a name and description in the **Name** and **Description** fields, respectively.
5. Select either **Complete URL** (default) or **Base URL** from the **Hashing Scheme** list. Depending on the selection, either the complete URL or the base URL of the incoming request is considered for computing the hash value. The generated hash value is then used for redirecting the incoming request to the origin server that stores the requested content.
6. In the **List of Origin Nodes** area:
 - To add one or more nodes (Media Flow Controller and non-Media Flow Controller nodes) to the origin map, click **Add**. The **Add Consistent Hash Node** page is displayed. On this page, you can enter details for a non-Media Flow Controller node:
 - a. **Origin Server IP**—Enter the IP address of the origin server.
 - b. **Port**—Enter the TCP port of the origin server. By default, this is set to **80**. To enable secure communication between Media Flow Controller and the origin server, set the port to **443**.
 - c. **HeartBeat Path**—Enter the relative URI to use to heart-beat the node.
 - d. Click **Add** to add the node to the origin map or click **Cancel** to exit the page without making any changes.

If you want to add more than one node, click the **Add more** button.

To add a Media Flow Controller node:

- a. Click **Select MFCs** from the **Add Consistent Hash Node** page. The **Add Origin Map Node** page is displayed.
- b. Select the check box adjacent to the nodes to add them to the origin map. Select only the nodes that are configured as reverse proxy servers.
- c. Enter the **Port**, **HeartBeat Path**, and **Interface** information.

By default, the port and the heartbeat path are set to **80** and **/root/heartbeat.html**, respectively.

- d. Click **Add** to add the selected nodes to the origin map or click **Cancel** to exit the page without making any changes.

If you want to add more than one node, click the **Add more** button.

- To modify the configuration, select the node and click **Modify**. The **Modify Origin Map Node:mfc_name** dialog box is displayed.



NOTE: You cannot modify the configuration of multiple Media Flow Controllers in a single operation. You have to select Media Flow Controllers one at a time and perform the changes.

- a. Modify the port number, heartbeat path, and interface as required.
 - b. Click **Modify** to save the changes or click **Cancel** to exit the dialog box without making any changes.
- To remove any or all of the nodes from the cache cluster, select the nodes and click **Remove**. The **Delete MFC Node(s)** dialog box is displayed. Click **Yes** to remove the nodes or click **No** to exit without making any changes.
7. On the **Connection Properties** tab, in the **Node Monitoring** area, specify the following information:
 - **Retry Count**—Enter the number of request failures that are allowed before the node is declared down.

You can enter a value from 0 through 4,29,49,67,295. The default value is **3**.
 - **Heartbeat Interval**—Enter the time in milliseconds for nodes to wait before the nodes send a “heartbeat” signal to the other nodes indicating that they are available.

You can enter a value from 0 through 36,00,000 milliseconds. The default value is **100** milliseconds.
 - **Connect Timeout**—Enter the allowable time in milliseconds for the connection to the socket to complete.

You can enter a value from 0 through 36,00,000 milliseconds. The default value is **100** milliseconds.

- **Read Timeout**—Enter the allowable time in milliseconds to complete reading from the socket after the connection is established.

You can enter a value from 0 through 36,00,000 milliseconds. The default value is **100** milliseconds.

8. In the **Origin Connection Setting** area:

- **Connect Timeout**—Enter the time in milliseconds after which you want to time out the connection request sent to the origin server.

Increasing the Connect Timeout value minimizes the load on the origin server.

Lowering the Connect Timeout value improves user experience by reducing the wait time for requests.

You can enter a value from 0 through 36,00,000 milliseconds. The default value is **100** milliseconds.

- **Read Timeout**—Enter the time in milliseconds after which you want to time out the read request if there is no response from the origin server.

Increasing the Read Timeout value minimizes the load on the origin server. Lowering the Read Timeout value improves user experience by reducing the wait time for requests.

You can enter a value from 0 through 36,00,000 milliseconds. The default value is **100** milliseconds.

- **Connect Retry Delay**—Enter the duration in milliseconds after which Media Flow Controller retries to establish a connection with the origin server.

You can enter a value from 0 through 36,00,000 milliseconds. The default value is **100** milliseconds.

Increasing the value of Connect Retry Delay minimizes the load on the origin server. Lowering the value of Connect Retry Delay improves user experience by reducing the wait time for request retries.

- **Read Retry Delay**—Enter the duration in milliseconds after which Media Flow Controller retries to read from the origin server.

Increasing the value of Read Retry Delay minimizes the load on the origin server. Lowering the value of Read Retry Delay improves user experience by reducing the wait time for request retries.

You can enter a value from 0 through 36,00,000 milliseconds. The default value is **100** milliseconds.

9. Click **OK**, **Cancel**, or **Reset**. **OK** instantiates the values you set, **Cancel** closes the configuration page, and **Reset** returns all values to their defaults.



NOTE: See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about consistent hash maps.

Related Documentation

- [Actions on Origin Maps on page 75](#)
- [Creating Escalation Maps on page 72](#)
- [Understanding Origin Maps on page 68](#)
- [Creating HTTP Reverse Proxy Services on page 112](#)
- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Creating Escalation Maps

The escalation map feature enables you to distribute content fetch requests across multiple origin servers for failover. For example, if you have configured a set of four nodes as an escalation map, each of these nodes contain the same content. If a node fails, the next node in the escalation map serves the request. You achieve 100-percent availability through this configuration.

Before configuring escalation maps, you must consider:

- The list of nodes you want to group
- The parameters you want to use for monitoring the nodes to ensure that escalation occurs only to the nodes that are currently online
- Whether the nodes are configured as reverse proxy servers

To configure an escalation map:

1. From the left navigation panel, click the plus sign (+) adjacent to **Design Elements**.
2. Click the plus sign (+) adjacent to **Manage Origin Map**.
3. Click **Add Escalation Map**. The **Add Escalation Map** page is displayed.
4. On the **Basic Properties** tab, enter a name and description in the **Name** and **Description** fields, respectively.
5. In the **List of Origin Map Nodes** area, configure a list of origin servers, which are logically viewed as one, where requests are sequentially initiated to specific origin servers (based on the order in which they are displayed in the UI, with the topmost origin server having the highest priority) until the request is satisfied or all known origin servers at request-initiation time have been tried. Origin servers can be Media Flow Controller or non-Media Flow Controller nodes.

- To add one or more nodes (Media Flow Controller and non-Media Flow Controller nodes) to the origin map, click **Add**. The **Add Escalation Map Node** page is displayed. On this page, you can enter details for a non-Media Flow Controller node:
 - a. **Origin Server Name / IP**—Enter the domain name or IP address of the origin server.
 - b. **Port**—Enter the TCP port of the origin server. By default, this is set to **80**. To enable secure communication between Media Flow Controller and the origin server, set the port to **443**.
 - c. **HeartBeat Path**—Enter the relative URI to use to heart-beat the node.
 - d. **HTTP Failure Response Code**—Select the codes that trigger escalation. Click **Add** or **Remove** buttons to add or remove HTTP response codes. By default, any **404**, **500**, and **505** responses from the origin server trigger escalation. If you want to remove any of these default HTTP response codes, select the HTTP response codes and click **Remove**.
 - e. Click **Add** to add the node to the origin map or click **Cancel** to exit the page without making any changes.

If you want to add more than one node, click the **Add more** button.

To add a Media Flow Controller node:

- a. Click **Select MFCs** on the **Add Escalation Map Node** page. The **Add Origin Map Node** page is displayed.
- b. Select the check box adjacent to the nodes to add them to the origin map. Select only the nodes that are configured as reverse proxy servers.
- c. Enter the **Port**, **HeartBeat Path**, **Interface**, and **HTTP Failure Response Code** information.
- d. Click **Add** to add the selected nodes to the origin map or click **Cancel** to exit the page without making any changes.

If you want to add more than one node, click the **Add more** button.

- To modify the configuration, select the node and click **Modify**. The **Modify Origin Map Node** dialog box is displayed.



NOTE: You cannot modify the configurations of multiple Media Flow Controllers in a single operation. You have to select Media Flow Controllers one at a time and make the necessary modifications.

- Modify the port number, heartbeat path, and interface as required.
- Click **Modify** to save the changes or click **Cancel** to exit the dialog box without making any changes.

- To remove any or all of the nodes from the cache cluster, select the nodes and click **Remove**. The **Delete Esc Map Node(s)** dialog box is displayed. Click **Yes** to remove the nodes or click **No** to exit without making any changes.
 - Click **(Up)** or **(Down)** to move a node up or down the hierarchy. The topmost origin server has the highest priority and the request is routed to this origin server first.
6. On the **Connection Properties** tab, in the **Node Monitoring** area, specify the following information:
- **Retry Count**—Enter the number of request failures that are allowed before the node is declared down.

You can enter a value from 0 through 4,29,49,67,295. The default value is **3**.
 - **Heartbeat Interval**—Enter the time in milliseconds for nodes to wait before the nodes send a “heartbeat” signal to the other nodes indicating that they are available.

You can enter a value from 0 through 36,00,000 milliseconds. The default value is **100** milliseconds.
 - **Connect Timeout**—Enter the allowable time in milliseconds for the connection to the socket to complete.

You can enter a value from 0 through 36,00,000 milliseconds. The default value is **100** milliseconds.
 - **Read Timeout**—Enter the allowable time in milliseconds to complete reading from the socket after the connection is established.

You can enter a value from 0 through 36,00,000 milliseconds. The default value is **100** milliseconds.
7. In the **Origin Connection Setting** area:
- **Connect Timeout**—Enter the time in milliseconds when you want to time out the connection request sent to the origin server.

Increasing the value of Connect Timeout minimizes the load on the origin server. Lowering the value of Connect Timeout improves user experience by reducing the wait time for requests.

You can enter a value from 0 through 36,00,000 milliseconds. The default value is **100** milliseconds.
 - **Read Timeout**—Enter the time in milliseconds when you want to time out the read request if there is no response from the origin server.

Increasing the value of Read Timeout minimizes the load on the origin server. Lowering the value of Read Timeout improves user experience by reducing the wait time for requests.

You can enter a value from 0 through 36,00,000 milliseconds. The default value is **100** milliseconds.
 - **Connect Retry Delay**—Enter the duration in milliseconds after which Media Flow Controller retries to establish a connection with the origin server.

Increasing the value of Connect Retry Delay minimizes the load on the origin server. Lowering the value of Connect Retry Delay improves user experience by reducing the wait time for request retries.

You can enter a value from 0 through 36,00,000 milliseconds. The default value is **100** milliseconds.

- **Read Retry Delay**—Enter the duration in milliseconds after which Media Flow Controller retries to read from the origin server.

Increasing the value of Read Retry Delay minimizes the load on the origin server. Lowering the value of Read Retry Delay improves user experience by reducing the wait time for request retries.

You can enter a value from 0 through 36,00,000 milliseconds. The default value is **100** milliseconds.

8. Click **OK**, **Cancel**, or **Reset**. **OK** instantiates the values you set, **Cancel** closes the configuration page, and **Reset** returns all values to their defaults.

Related Documentation

- [Actions on Origin Maps on page 75](#)
- [Creating Consistent Hash Maps on page 69](#)
- [Understanding Origin Maps on page 68](#)
- [Creating HTTP Reverse Proxy Services on page 112](#)
- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Actions on Origin Maps

From the **Manage Origin Maps** page, you can perform the following actions on origin maps by clicking the links on the **Actions** list. You have to select the origin maps before performing any actions on them:

- **Modify Origin Map**—Click this link to modify all configuration settings except the origin map name. After the changes are saved, the revised configuration is reflected in all the services that use this origin map.

You can modify only one origin map at a time.

- **View Origin Map**—Click this link to view the configuration of the selected origin map.

In the pop-up that appears, you can sort the data under the **Origin Map Node List** table and even choose what columns you want to display by:

- Mousing over a column and clicking the list.
 - Selecting **Sort Ascending** or **Sort Descending** to sort the data in ascending or descending order.
 - Selecting **Columns** and choosing the columns to display.
- **Delete Origin Map(s)**—Click this link to delete one or more origin maps.

If the origin map is associated with a service, you must make sure that the service is no longer provisioned to a device before deleting the origin map.

- Related Documentation**
- [Creating Consistent Hash Maps on page 69](#)
 - [Creating Escalation Maps on page 72](#)
 - [Understanding Origin Maps on page 68](#)
 - [Media Flow Activate Overview on page 3](#)
 - [Quick Reference to Tasks in Media Flow Activate on page 191](#)

CHAPTER 7

Policy Scripts

- [Understanding Policy Scripts on page 77](#)
- [Adding Policy Scripts on page 78](#)
- [Actions on Policy Scripts on page 79](#)

Understanding Policy Scripts

This topic describes what policy scripts are used for and the information you need to know before uploading a policy script into the Media Flow Activate database.

The Policy Engine in Media Flow Controller provides an infrastructure for administrators to define rules that control the caching and delivery functions of Media Flow Controller, at runtime. It enables policy administrators to define policies on the basis of the source or destination IP address, content type, request or response headers, and so on.

Policy administrators create policy scripts consisting of a set of rules by using the Tool Command Language (TCL) scripting language. Using Media Flow Activate, you can import a previously created and validated policy script (*.tcl file) into the Media Flow Activate database and bind it to a service. After the service has been provisioned on to a Media Flow Controller, the Policy Engine invokes specific procedures in the policy script when the service receives an HTTP request. During a transaction, the policies are invoked:

- After the connection between the client and Media Flow Controller is established and the HTTP request is received and parsed by the Media Flow Controller. The policy administrator includes any rules that decide whether this connection should be continued or rejected, here.
- After Media Flow Controller receives the request and if there is a cache miss, just before this request is forwarded to the origin server. Any rules based on the URI, query, referrer, or headers are included here. Cache or no cache decision could also be made here.
- After Media Flow Controller receives the response from the origin server and the response is parsed. Any rules based on content length, content type are included here. Cache or no cache decision could also be made here.
- Just before sending the response from Media Flow Controller to the client.

You use the policy scripts in the following scenarios:

- To prevent unauthorized content downloads

- To redirect users to different websites, for error handling or service differentiation
- To improve bandwidth savings by overriding the cache or no-cache directions from the origin server

Using Media Flow Activate, you cannot create TCL scripts. You can only upload a previously created and validated script into the Media Flow Activate database from the Design Elements workspace. After uploading the policy script, you need to bind it to a service. After the service has been provisioned on to a Media Flow Controller, the policies are invoked when the service receives an HTTP request from a client.

**Related
Documentation**

- [Adding Policy Scripts on page 78](#)
- [Actions on Policy Scripts on page 79](#)
- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Adding Policy Scripts

Using Media Flow Activate, you can add a previously created and validated policy script to the Media Flow Activate database and then bind it to a service. After the service is provisioned on to a Media Flow Controller, the Policy Engine invokes the policy rules that are configured within the policy script when there are client requests to the website or domain configured in the service.

Before you begin adding a policy script, make sure that the policy script (*.tcl file) is available on your local system.

To add a policy script:

1. Click the plus sign (+) next to **Design Elements**.
2. Click the plus sign (+) next to **Manage Policy Script**.
3. Click **Add Policy Script**. The **Add Policy Script** page is displayed.
4. Enter information for the following fields on the **Add Policy Script** page:
 - **Select Policy Script File**— Click **Browse** to locate the file (*.tcl file). This file must be available on your local system.
 - **Policy Name**— Enter a name for the policy. It can be the same as that of the policy script filename. You enter a value in this field so that you can identify this policy and associate it with a Network Optimization service or an HTTP reverse proxy service from the Service Design workspace.
 - **Description**— (Optional) Enter a description of the policy script that further identifies the policy you named. For example: "This script is for blocking users from viewing inappropriate content in a campus edge deployment."
 - Click **Add**, **Cancel**, or **Reset**. **Add** instantiates the values you set, **Cancel** closes the configuration page, and **Reset** returns all values to their defaults.

If the policy has been successfully added, you can view the newly added policy on the **Manage Policy Scripts** inventory landing page. This page displays the following information:

- Policy filename
- User who added the policy script
- User who modified the policy script
- Timestamp when the last modification was made

You need to bind the policy to a service so that the rules defined in the policy script are invoked when a client makes a request to the website or domain configured in that service.

**Related
Documentation**

- [Actions on Policy Scripts on page 79](#)
- [Understanding Policy Scripts on page 77](#)
- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Actions on Policy Scripts

From the **Manage Policy Scripts** page, you can perform the following actions on the policy scripts by clicking the context-sensitive menu that appears when you right-click the policy scripts. You have to select the policy scripts before performing any actions on them:

- **Modify Policy Info**—Click this link to modify the description of the policy script. You cannot modify any other configuration of the policy script from the Media Flow Activate GUI. However, if you want to do so, you have to import a new policy script to the Media Flow Activate database. You can modify only one policy script at a time.
- **Delete Policy(s)**—Click this link to delete one or more policy scripts. If the policy script is associated with a service, unbind the policy script from the service and then delete the policy script. To unbind a policy script from a service, select another policy from the **Policy Script** list or leave the field blank in the “HTTP Reverse Proxy Service Design” workspace.
- **Export Policy Script**—Export the policy script from the Media Flow Activate database to your local system. You may want to do this when you want to modify the Tool Command Language (TCL) script, rename the script file, and later import it back to the Media Flow Activate as a new policy script file and then associate it with a service with the updated configuration.

**Related
Documentation**

- [Understanding Policy Scripts on page 77](#)
- [Adding Policy Scripts on page 78](#)
- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

CHAPTER 8

Virtual Players

- [Understanding Virtual Players on page 81](#)
- [Creating Virtual Players on page 83](#)
- [Actions on Virtual Players on page 88](#)

Understanding Virtual Players

Media Flow Controller uses a **Virtual Player** function that helps optimize media viewing. The **Virtual Player** enables you to configure the parameters in a URL that represent object identity. You can create any number of virtual players; virtual players are used when they are assigned to a configured **Network Optimization** service or **HTTP Reverse Proxy** service.

There are two types of virtual players:

- The **Generic** type has a superset of delivery options appropriate for most media delivery.
- The **You Tube** type provides YouTube-specific options for caching and trick play. The term “trick play” refers to such video viewing functions as seek, fast forward, fast rewind, and so forth.

Before you begin configuring a virtual player, you must have the following information:

- The query parameters used in the URLs to pass information for each type of video you want to deliver with trick play functions, such as seek.
- The MD5 authentication parameters needed for hash verification. See “Hash Verify Overview” in this topic for details about hash verification.
- Bandwidth parameters, including maximum connection limits.
- Parameters for **Fast Start** and **Full Download** functions. The **Fast Start** option provides parameters for delivering files at the fastest possible speed. The **Full Download** option provides parameters for downloading the entire media file at the fastest possible speed. You enter either a query string or a header name to be matched in the request to indicate full download; you can also choose to never or always allow full download.

Hash Verify Overview

Configuring **Authentication Properties** enables Media Flow Controllers to compute an MD5 hash of an incoming URL by combining a part of the URL, specified by the **Hash Computation** option, and including the **Expiry Time Identifier (Q.S.)** query string value, if

used, along with a configured **Shared Secret** that is appended or prefixed (as configured) to the **Location** option. The computed hash digest value is then compared with the hash value provided in the incoming URL via the **Hash Identifier (Query String)**. If a match between the computed and provided hash values is unsuccessful, the request is denied.

The following is an example URL showing **Expiry Time Identifier (Q.S.)** *e* and **Hash Identifier (Query String)** *h*:

`http://www.example.com/media/foo.flv?e=3312665958&h=<128-bit-md-5-hash>.`

If Media Flow Controller encounters this URL, and **Hash Computation** is set to **ABSOLUTE_URL**, Media Flow Controller takes the entire URL up to the configured **Hash Identifier (Query String)** (*&h* in the example).

If **Hash Computation** is set to **RELATIVE_URL**, Media Flow Controller takes the part of the URL after the access method and domain, plus the query string up to the configured match query string (*/media/foo.flv?e=3312665958*, in the example).

If **Hash Computation** is set to **OBJECT_NAME**, Media Flow Controller takes the part of the URL after the last slash, plus the query string up to the configured **Hash Identifier (Query String)** (*foo.flv?e=3312665958*, in the example).

The hash value is then computed by either appending or prefixing to the URL (or part of the URL, if **Hash Computation** is set to **RELATIVE_URL** or **OBJECT_NAME**) the configured **Shared Secret**, and comparing the computed value with the hash value provided via the **Hash Identifier (Query String)** (shown above as the URL section after the last =).

The following is an example when **Shared Secret** is appended and **Hash Computation** is set to **ABSOLUTE_URL**:

Computed hash value =
MD5(`http://video.example.com/public/2010/qwerty.flv?fs=5000&ri=300&rs=1234567`
+ **shared-secret**).

The following is an example when **Shared Secret** is prefixed and **Hash Computation** is set to **ABSOLUTE_URL**:

Computed hash value = **MD5**(**shared-secret** +
`http://video.example.com/public/2010/qwerty.flv?fs=5000&ri=300&rs=1234567`)



NOTE: See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about virtual players.

Related Documentation

- [Creating Virtual Players on page 83](#)
- [Actions on Virtual Players on page 88](#)
- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)

- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Creating Virtual Players

The Virtual Player function enables you to configure the parameters in a URL that represent object identity, providing a way to control the delivery of media.

Before configuring a virtual player, you must have the following information:

- The query parameters used in the URLs to pass information for each type of video you want to deliver with trick play functions, such as seek, fast forward, fast rewind, and so forth.
- The MD5 authentication parameters needed for hash verification. See “Hash Verify Overview” in [“Understanding Virtual Players” on page 81](#) for details about hash verification.
- Bandwidth parameters, including maximum connection limits.
- Parameters for **Fast Start** and **Full Download** functions.

You apply your virtual player to websites that you create for delivery of different media. After you create the virtual player, you can apply it to any number of websites.

To configure virtual players on the **Design Elements** workspace:

1. From the left navigation panel, click the plus sign (+) adjacent to **Design Elements**.
2. Click the plus sign (+) adjacent to **Manage Virtual Player**.
3. Click **Add Virtual Player**. The **Add Virtual Player** page is displayed.
4. On the **Basic Properties** tab, specify the following information:
 - a. Enter a **Player Name**.
 - b. On the **Player Type** list, select either **Generic** or **You Tube**. A **Generic** virtual player provides options allowing you to implement trick play for most video delivery. The **YouTube** virtual player specifically provides trick play for YouTube video delivery.
 - c. **Seek Configuration**—In this area, you can configure the following options to enable viewers to begin video play at different parts of the video:
 - **Start Identifier**—Enter a string whose referenced value (sent by the client player) indicates, in bytes (for FLV) or in seconds (for MP4), when to begin seek.

Value entered is limited to 128 characters; for example, **begin**. For example, a **Start Identifier** query string, with a referenced value of 100, would mean to start seek at the 100th byte (FLV), or 100th second (MP4), of the incoming URL.
 - **End Identifier**—Enter a string whose referenced value is in bytes (sent by the client player) to specify how much data to send after the **Start Identifier**; this is applicable only to FLV media files.

Value entered is limited to 128 characters; for example, **len**. For example, an **End Identifier** query string, with a referenced value of 1000, would mean to stop delivery 1000 bytes after the start of seek.

- **Tunnel Seek Request**—Select this check box to tunnel all seek requests to the origin server. This option needs to be enabled only when the origin site changes its seek mechanism.
- d. **Fast Start**—Select the **Fast Start** check box to enable Media Flow Controller to burst an initial portion of media data to quickly fill the client buffer to enable fast-start of video playback. Configure one of the following options:
- **Default Size (kbps)**—Enter the number of kilobytes that should be expedited.
 - **Use Query String Parameter**—Enter a string; the referenced value must be in kilobytes. This string is in the request header and this value is set as the Fast Start value.
 - **Time (seconds)**—Enter the number of seconds of media data to expedite for delivery. This option is relevant only for video files (such as FLV, MP4, and WMV). The media is delivered at the detected bit rate for the configured duration. If the bit rate cannot be detected or the file is not a video asset, zero bytes are delivered because the fast-start would then revert to a size of 0 (zero) bytes.
5. On the **Connection Properties** tab, specify the following information:

Figure 4: Add Virtual Player Window—Connection Properties Tab

- a. **Maximum Connection Bandwidth**—In this area, you can limit the number of connections any one virtual player can consume by choosing one of the following options:
 - **No limit** (default)—Select this option if there is no limit on the maximum bandwidth for a session.
 - **Value**—Select this option to enter a value in Kbps for the maximum bandwidth for a session. Even if there is available bandwidth in the link, only this value is allocated for a session.
- b. **Video Pacing/ Bit Rate Throttling**—Select this check box to control the bit rate at which Media Flow Controller delivers a video. You can configure Media Flow Controller to deliver the video at the configured rate only when the system resources are available.

Max Bit Rate—This is a best-effort rate control mechanism wherein if the system resources are available, Media Flow Controller delivers the requested video at the specified bit rate. Configure one of the following options:

- **Auto detect**—To auto-detect the bit rate of a video file and enforce the rate of delivery. This feature is also known as video pacing or bit-rate throttling. Media Flow Controller supports bit-rate throttling for MP4, FLV, and WMV/ASF media formats. This is the default.
- **Static**—To statically enforce a fixed delivery rate for all objects. Media Flow Controller enforces this rate of delivery for all the objects across the service to which this virtual player is associated. You specify the value in Kbps.

You can enter a value from 0 through 4,29,49,67,295 Kbps. The default value is 0 Kbps.

- **Query string parameter**—To enforce the delivery rate as requested by the client in its request. In some scenarios, client players can explicitly ask for a particular delivery rate by using a preconfigured query-string parameter. Media Flow Controller looks for preconfigured query-string parameter in the incoming request and extracts its value to enforce the delivery rate. The units that this query string value represents must be explicitly configured. The allowed options are Kbps, KBps, Mbps, and MBps.

You can also configure **Burst factor** to increase the speed of video delivery at a rate greater than the requested or detected rate. By default, Burst factor is set to 1.1 (10 percent faster than the encoded bit rate). The allowable range for the Burst factor is from 1.1 to 3. Any TCP/IP overheads necessary for delivery are automatically accounted for. When Media Flow Controller paces video delivery by using the Burst factor, it enforces a rate equivalent to $\text{Burst factor} \times (\text{Auto detect} \mid \text{Static} \mid \text{Query string parameter})$.

Example: Assume that you have configured: Max Bit Rate control scheme, Auto detect, and Burst factor to 2. If Media Flow Controller detects the encoded bit rate in the metadata of the video as 1000 bits per second, then because of this configuration, Media Flow Controller delivers the video at twice the encoded bit rate (that is, at 2000 bits per second). However, if the system is overprovisioned, then the bandwidth is shared among all active connections, effectively lowering the delivery rate to less than the enforced rate for all connections. To avoid oversubscription, configure the Max Bit Rate option along with proper system provisioning through resource pools.

- c. **Full Download**—Select this check box to allow Media Flow Controller to download the entire media file at the fastest possible speed. To configure **Full Download**, select one of the following options:
 - **Always**—Downloads are always delivered at the fastest possible speed.
 - **Query String Match (Name) and (Value)**—Downloads are delivered at the fastest possible speed when a match is found for the specified query string.
 - **Request Header Match (Name) and (Value)**—Downloads are delivered at the fastest possible speed when a match is found for the specified header name.
6. On the **Authentication Properties** tab, specify the following information:
 - a. **Enable MD5 Authentication**—Select this check box to configure MD5 authentication parameters. Specify the following:

- **Hash Identifier (Query String)**—Enter a string indicating the provided hash value; the default for this virtual-player type is h.
- **Expiry Time Identifier (Query String)**—Enter the query parameter present in the video URL, which acts as the expiry time identifier. At runtime, Media Flow Controller uses this query parameter to extract the value of expiry timestamp specified by the player (that issued this request). Media Flow Controller serves the object only if the request URL is not expired—that is, if the value extracted for the query parameter from the incoming request URL is greater than the current system time.

For example, consider the following request video URL:

"http://www.example.com/media/foo.flv?

e=3312665958&h=ec41f550878f45d9724776761d6ac416." Enter "e" in the Expiry Time Identifier (Query String) field to use the "e" query parameter as the expiry time identifier.

b. For the **Shared Secret**, make the following specifications:

- **Value**—Enter a secret key that is then appended or prefixed (as specified in the Location value) to the URI to calculate the hash, which is then "matched" with the match query-string-param hash value.
- **Location**—Either **APPEND** the shared secret to the front of the URI, or **PREFIX** it to the end of the URI.

c. Choose a **Hash Computation** value:

- **ABSOLUTE_URL**—Use the entire request URL (including the query string up to the configured **Hash Identifier** query string value).
- **RELATIVE_URL**—Use only the URI part of the request URL, excluding the domain, and access method (but including the query string up to the configured **Hash Identifier** query string value).
- **OBJECT_NAME**—Use only the object name part of the request URL (and the query string up to the configured **Hash Identifier** query string value).

7. Click **Ok**, **Cancel**, or **Reset**. **Ok** instantiates the values you set, **Cancel** closes the **Add Virtual Player** configuration page, and **Reset** returns all values to their defaults.



NOTE: You can also add a virtual player by clicking the **Import Virtual Player** link on the **Manage MFCs > Manage Virtual Players** page, **Actions** list, with no virtual player selected. You use a defined XML file to import a virtual player.



NOTE: See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about virtual players.

**Related
Documentation**

- [Actions on Virtual Players on page 88](#)

- [Understanding Virtual Players on page 81](#)
- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Actions on Virtual Players

From the **Manage Virtual Players** page, you can perform the following actions on virtual players by clicking the links on the **Actions** list. You have to select the virtual players before performing any actions on them:

- **Copy Player**—Click this link to create a copy of the selected player. You are prompted for a name for the new virtual player. Other than the name, all other configuration settings remain the same as that of the copied player.
- **Modify Player**—Click this link to modify all other configuration settings other than the name and type of the virtual player. After the changes are saved, the revised configuration is reflected in all the services that use this virtual player.

You can modify only one virtual player at a time.

- **Delete Player(s)**—Click this link to delete one or more virtual players.

Related Documentation

- [Creating Virtual Players on page 83](#)
- [Understanding Virtual Players on page 81](#)
- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

PART 4

Service Design

- [Overview on page 91](#)
- [Network Optimization Service on page 97](#)
- [HTTP Reverse Proxy Service on page 111](#)
- [Content Ingest Service on page 135](#)

CHAPTER 9

Overview

- [Service Design Overview on page 92](#)

Service Design Overview

You use the **Service Design** workspace to create content delivery services. Transparent, reverse proxy, and content ingest content delivery services are supported. You create these services for the types of media you are delivering.

If you are an Internet service provider, create a **Network Optimization Service** to transparently cache a popular website, thereby saving bandwidth and optimizing Internet content delivery.

If you are a content provider or a Content Delivery Network (CDN) service provider who owns or manages any content, create an **HTTP Reverse Proxy Service** to efficiently deliver that content.

If you are a CDN service provider, create a **Content Ingest Service** to ingest popular content before it is needed. Using content ingest service, you can periodically refresh the cached content for sites that are regularly updated.

Before you begin configuring proxy services, it is good to have the following information handy:

- The regular expression (regex) for the **Domain** that will be used in requests for the media for this service. In addition, the file **Path** to those videos in that domain.
- The precedence you want to give this service among all the configured services. The higher the **Precedence** value, the lower the precedence. See “Understanding Precedence” in this topic for details about using the “precedence” parameter.
- For cache-misses (requests for media not in cache), how you want the service to obtain the requested media.
- The **Virtual Player** you want to manage video delivery policies for that site. This is needed when the site being cached serves HTTP video. A single virtual player can be used for multiple sites. See “[Creating Virtual Players](#)” on page 83 to create a virtual player.
- The **Cache Tuning Policy** you want to fine-tune the cache settings on the basis of the network conditions and caching requirements. See “[Creating Cache-Tuning Policies](#)” on page 52 to create a cache-tuning policy.
- If you are creating a reverse proxy service, consider the list of origin servers that Media Flow Controller must access to fetch content upon a cache-miss.

Understanding Precedence

Use **Precedence** to set unambiguous mapping of incoming GET requests in the case of **Domain** overlap; precedence can be set on all domains. The lower the number, the higher the preference for that domain; values 0 (highest precedence) through 10 (lowest precedence) can be used. All services have a default precedence of 0.

Example for network optimization service: Consider three websites and service configurations as follows:

- Website 1

`http://a.com/abc/def/file1.flv`

Name `website1`

Domain `a.com`

Path `/abc/def`

Precedence 1

Origin Server Resolution Follow Host Header

Use Client IP

- Website 2

`http://a.com/abc/file2.flv`

Name `website2`

Domain `a.com`

Path `/abc`

Precedence 2

Origin Server Resolution Follow Host Header

Use Client IP

- Website 3

`http://a.com/pqr/file3.flv`

Name `website3`

Domain `a.com`

Path `/`

Precedence 3

Origin Server Resolution Follow Host Header

Use Client IP

All three websites match **Domain a.com** and **Path /** (slash). In order to ensure that media files at `/abc/def` map to **website 1** and not **website 3**, set the **Precedence** value higher for **website 1**. This is the same as the case with mapping media files at `/abc` to **website 2**. Only media files at the default location, `/`, should be mapped to **website 3** with the **Precedence** value configured the lowest, as shown above. Similar logic applies to the reverse proxy service as well.

Example for reverse proxy service:

- Website 1

`http://a.com/abc/def/file1.flv`

Name `website1`

Domain a.com

Path /abc/def

Precedence 1

HTTP Origin FQDN Server Name 10.102.0.81 Port 80

- Website 2

http://a.com/abc/def/file2.flv

Name website2

Domain a.com

Path /abc

Precedence 2

HTTP Origin Origin map xyz



BEST PRACTICE: Excessive use of precedence has a performance impact because precedence allows for the longest prefix matching. If possible, configure websites so that there are no overlaps in the Domain and Path combination, that is used for mapping the incoming HTTP GET to the requested media files.

Understanding 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 parentheses 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 regular expression 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 regular expression, to a new URI.

The commands for dynamic URI mapping specify that an incoming request having a URI matching the configured regex value (**url to match**) is matched to a cache index string value (**map-to**). The **tunnel unmatched** option specifies that if the match fails, the request is tunneled. With **revalidate cache hit = true** configured, if the origin server returns "Object Is Modified," the transaction is tunneled, and the object is deleted from the cache, provided that its time-to-live (TTL) has expired; and the new object is fetched into the cache.



NOTE: See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about namespaces.

**Related
Documentation**

- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Network Optimization Services Overview on page 97](#)
- [HTTP Reverse Proxy Services Overview on page 111](#)
- [Content Ingest Services Overview on page 135](#)
- [Tagging Media Flow Controller Objects on page 30](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Network Optimization Service

- [Network Optimization Services Overview on page 97](#)
- [Creating Network Optimization Services on page 98](#)
- [Creating Network Optimization Service XML Files for Import on page 105](#)

Network Optimization Services Overview

This topic describes what Network Optimization services are used for and what information you need to know before creating a service.

At a minimum, service instance configuration requires a **Domain**, a **Path** to the media files, and an **Origin Server Resolution** scheme. You can further define control by assigning a configured **Virtual Player** and **Cache Tuning Policy**. The service instance is referenced via the URL in the HTTP request to Media Flow Controller.

For example, if you are serving content through Media Flow Controller for media under the following directories from your origin library:

- `example.com/videos/trg`
- `example.com/videos/UGC`
- `example.com/videos/premiumcontent`

You can create three service websites: **TRG**, **UGC**, and **Premium**, each with a different set of delivery policies.



NOTE: See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about configuring and deploying transparent proxy services.

Related Documentation

- [Creating Network Optimization Services on page 98](#)
- [Service Design Overview on page 92](#)
- [Provisioning Services Overview on page 141](#)
- [Media Flow Activate Overview on page 3](#)

- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Creating Network Optimization Services

Use the **Service Design** workspace to create content delivery services. You create these services for the types of media you deliver.

Before you begin configuring a **Network Optimization** website media delivery service, you need the following information:

- The regular expression (regex) for the **Domain (Regex)** that is used in requests for the media for this service; see [Table 4 on page 104](#) for example domain regex values. You also need the file **Path** to the media in that domain that you expect to deliver.
- The **Precedence** to give this service among all the configured services. The higher the **Precedence** value, the lower the precedence.
- For cache misses (requests for media not in the cache), how you want the service to obtain the requested media.
- The **Virtual Player** to manage video delivery policies for that site. This is needed when the site being cached serves HTTP video. A single virtual player can be used for multiple sites. See [“Creating Virtual Players” on page 83](#) to create virtual players.
- The **Cache Tuning Policy** to fine-tune the cache settings based on the network conditions and caching requirements. See [“Creating Cache-Tuning Policies” on page 52](#) to create cache-tuning policies.

Create a separate **Network Optimization** website delivery service for each media repository that you have.

To configure Network Optimization services on the **Service Design** workspace:

1. From the left navigation panel, click the plus sign (+) adjacent to **Service Design**.
2. Click **Network Opt Services**. The **Network Optimization Services** page is displayed.
3. Click **Add Service Instance**. The **Add Service Instance** page is displayed.

Figure 5: Create Network Optimization Service—General Tab

4. On the **General** tab, specify the following information:

- a. **Name** for the service and **Description** (optional) of the service.
- b. **Domain**—Enter the defined fully qualified domain name (FQDN) that is matched with the incoming HOST header to identify the request. This field is mandatory.

Select **Regex** for Media Flow Controller to treat the entry in the **Domain** field as a regex entry. Enclose all regex entries in double quotation marks—for example, a regex for www.example.com plus example.com could be: “`^.*\example\.com`”. 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. See [Table 4 on page 104](#) for more examples of regex entries.
- c. **Order of serving client request**—Select the policy that determines the order in which the objects are ingested from the RAM cache into the disk cache when handling bulk requests.

To avoid clients waiting for data while Media Flow Controller ingests data into the disk cache, the data serving path and the ingestion path are decoupled in Media Flow Controller. While objects are being served from the RAM cache, the ingestion process in Media Flow Controller picks these objects from the RAM cache and ingests them into the disk cache. The order in which the ingestion process picks these objects for ingestion can be tuned:

- **Last in first out**—(Default) The object served last is ingested into the disk cache first. Because the object is selected for ingestion immediately after it has been served to the client, the probability of finding the object in the RAM cache is high, which prevents refetching of the object from the origin server.
- **First in first out**—The object served first is ingested into the disk cache first. Here, the objects are ingested into the disk cache in the order in which they have been served to the clients from the RAM cache. However, this feature may introduce additional fetches from the origin server if the objects are evicted from the RAM cache before they are ingested into the disk cache.

Recommendations: For caching adaptive bit-rate streaming videos (such as Microsoft Smooth Streaming videos or Apple HTTP Live Streaming videos), it is recommended that you set the configuration to **First in first out**. For Network Optimization service deployments, it is recommended that you set the configuration to **Last in first out**.

Example for “First in first out”: Consider that in a given instant (say, at t_0 seconds), Media Flow Controller receives 100 requests for 100 objects. After Media Flow Controller fetches these objects from the origin server, the order in which these objects are ingested into disk cache is determined by whether Media Flow Controller is configured for **First in first out** or **Last in first out**. If it is configured for **First in first out**, then it ingests these objects into disk cache starting from the first object that was served to the client, followed by the second object, and so on. At t_1 seconds, if Media Flow Controller receives another 100 new requests and if it has cached only 80 objects at the end of t_0 seconds, the cacheable disk queue becomes 120 objects and it starts caching from the eighty-first object. At t_2 seconds, if Media Flow Controller receives another 100 new requests and if it has cached only 160 objects at the end of t_1 seconds, the cacheable disk queue now becomes 140 objects and it starts caching from the hundred and sixty-first object.

- d. **Ignore all object correlation validators**—Select this check box for Media Flow Controller to ignore the object validation fields (such as ETag and Last Modified headers) in the origin server response when it validates an object.

When the ETag or the Last Modified header is different from the previously cached origin server response, by default, Media Flow Controller assumes that the object is of a different version and deletes the cached object and replaces it with the modified object. If you select the **Ignore all object correlation validators** check box, Media Flow Controller ignores any changes to these headers in the origin server response and serves the previously cached object, provided that it has not expired. However, if the object has expired, Media Flow Controller deletes the existing cached object, caches and serves the object fetched from the origin server.

Example: Media Flow Controller uses the ETag header or the Last Modified header in the origin server response to validate whether the object that is currently cached is of the same version as that in the origin server. If the object has been modified in the origin server, then the ETag header value or the Last Modified header value sent by the origin server is different from the value that was previously sent by the origin server. If the values are different, Media Flow Controller deletes the existing

cached content, fetches the modified content from the origin server, and caches it.

At the time of caching the origin server response, Media Flow Controller associates a version number with each of the object that is being cached. This version number is generated from the ETag header value of the origin server response, provided that the ETag header is present; otherwise, it is generated from the Last Modified header value, provided that the Last Modified header is present. If both are absent, then Media Flow Controller generates a random version number. Though this is a good mechanism for validating the freshness of the object, you may find that in CDN deployments, the objects might not be cached at all in Media Flow Controller when the client is making byte-range requests.

Scenario 1: When you have a load balancer set up, where multiple origin servers might respond to byte-range requests, it is possible that the ETag header values are different in the responses sent by each of these origin servers. Consider a client requesting a 10-MB object in byte-range requests of 2 MB. For the first request of 0 to 2 MB, if there is a cache miss, then Media Flow Controller forwards this request to the origin server. If origin server 1 (OS1) responds to this request, Media Flow Controller caches this response and also generates a version number from the ETag header value of the response, and associates it with the object. For the next request of 2 to 4 MB, upon a cache miss, if origin server 2 (OS2) responds, it is likely that the ETag header sent by this origin server might be different. At the time of caching, Media Flow Controller assumes that this is a different version of the object and deletes the previously cached 0 to 2 MB. Media Flow Controller tunnels the 2- to 4-MB response because it cannot cache this content without caching the previous 0 to 2 MB. The remaining byte-range requests are also tunneled to the client. Here, Media Flow Controller does not cache the entire 10-MB object. This is one scenario where even though the response is cacheable, Media Flow Controller does not cache the response.

Scenario 2: Assume the origin server does not send the ETag or the Last Modified header in its response to the byte-range requests. Consider a client is requesting a 10-MB object in byte-range requests of 2 MB. When Media Flow Controller caches the first request of 0 to 2 MB, it generates a random version number and associates it with the cached object. This is because there is no ETag or Last Modified header in the origin server response to generate the version number. When Media Flow Controller tries to cache the next request of 2 to 4 MB, it generates another random version number to associate with the object. At the time of caching, Media Flow Controller compares the version numbers of 0 to 2 MB and 2 to 4 MB of the object. Because the version numbers are different, Media Flow Controller deletes the cached 0 to 2 MB and tunnels the 2 to 4 MB of the object directly to the client. The remaining byte-range requests are also tunneled to the client.

- e. **Custom cache-control header**—Specify a custom cache-control header name in the origin server response to be used for determining cache expiry. Assume that you have configured “xyz” for this field. If the origin server response contains the “xyz” header, then Media Flow Controller sets the expiry time of the object to the value contained in this header. If the xyz header value is 2400, then Media Flow Controller serves the object from its cache for the next 40 minutes from the time of caching. After this duration, the object is considered expired.

Example: Some origin servers do not send the expiry information of an object to intermediate proxies using standard HTTP headers, such as the Expires header or the Cache-control:max-age header. Instead, they send it by using custom headers. Media Flow Controller expects the object expiry information from these standard HTTP headers and when these headers are not present in the origin server's response, it automatically assumes that the origin server has not sent this information. Media Flow Controller then tries to apply the expiry value configured in the **cache-age default** or **cache-age content-type any** CLIs (that is, the values you have configured for **Default Cache Age** or **Cache Age Override** in the Cache tuning policy). However, for Media Flow Controller to use the expiry information present in the custom header of the origin server's response, you must specify the custom header name in the **Custom cache-control header** field. Media Flow Controller now detects that this custom header contains the expiry information for the object and it proceeds to set the object expiry time to this value.

For example, consider that the origin server sends a custom header, "x-cache-control:max-age=3600." For Media Flow Controller to detect that the custom header, "x-cache-control," in the origin server's response contains the expiry information of the object, you need to set the value of the **Custom cache-control header** field to "x-cache-control." Media Flow Controller then serves the object from its cache for the next one hour from the time of caching.

- f. **Path**—Enter the actual file path where the media files are located. The default path is "/" (slash), which refers to all files in the configured **Domain**.

Select **Regex** for Media Flow Controller to treat the entry in the **Path** field as a regex entry.

- g. **Precedence**—Select a value to break ties between services defined with the same **Domain**. The lower the number, the higher the preference for that service; 0 is the default and highest.
- h. **Origin Server Resolution**—Select an option to specify how Media Flow Controller determines and accesses the origin server for cache misses; choose one of the following options:
- **Follow Host Header**—Use the host header of the incoming request as the origin server.
 - **Destination IP**—Use the destination IP address of the incoming request as the origin server.
- i. **Use Client IP**—(Default) Select this check box to use the client IP address in place of the origin server's IP address in the request.
- j. **Tunnel All**—Select this check box to tunnel all incoming client requests directly to the origin server without further processing. By default, this feature is disabled.
- When you enable this feature, all caching-related configuration is ignored and the requests are directly tunneled to the origin servers.
- k. **Virtual Player Name**—Select a defined virtual player; see ["Creating Virtual Players" on page 83](#) to create a virtual player.

- l. **Cache Tuning Policy**—Select a defined policy; see [“Creating Cache-Tuning Policies” on page 52](#) to create a cache-tuning policy.
- m. **Policy Script**—Associate a policy script from a list of previously added policy scripts.
 You typically associate a policy script when you want to have a fine granular control over the various features of Media Flow Controller at runtime.
 Make sure that the policy script file is available on the local system from where Junos Space is accessed.
- n. **Revalidate always**—Select this check box for Media Flow Controller to revalidate its cached object for every client request.
 When Media Flow Controller receives a request from the client and this feature is enabled, Media Flow Controller first serves the object from its cache if the object has not expired. However, in parallel, Media Flow Controller sends a revalidation request to the origin server to determine whether the cached object has been modified in the origin server or not. Depending on the origin server’s response, if the object has not been modified, Media Flow Controller updates only the expiry time of the object; otherwise, Media Flow Controller deletes the existing cached object and caches the latest modified object. This is the default behavior—that is, the objects are revalidated offline. However, to revalidate the objects before serving them to the clients, set **Revalidation Mode** to **Inline**.
- o. From the **Dynamic URIs** tab, configure parameters using which Media Flow Controller can identify dynamic URIs and ensure the delivery of the correct content. An incoming request having a URI that matches the configured regex value (specified in the **URL to Match** field) is matched to a cache index string (specified in the **Map to** field). For more information about URI remapping, see “Understanding Dynamic URI Mapping.”



NOTE: **URL to Match** value must always be configured along with **Map To**. **Tunnel Unmatched** can be selected only if the **URL to Match** and **Map To** values are provided.

- **URL to Match**—Configure a regex expression to match 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. PCRE regex is not allowed; whereas GNU regex is allowed. The default value is NULL.
 Example: `/get/[^/]+/[^/]+/[^/]+/(.*)`
- **Map to**—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.
 Example: `/$1`

- **Tunnel Unmatched**—Select the check box to tunnel the request when no regex match is found.

5. Click **Ok** or **Cancel**. **Ok** instantiates the values you set and closes the page; **Cancel** closes the page without making any changes.



NOTE: To add a service website, go to the **Service Design > Network Opt Services** page. On the **Actions** list, click **Import Service** (this link is active only when you have not selected any services from this page). You use a defined XML file to import a service website.

Table 4: Example: Domain Regex Values

Regex	Matches
"www.example.com example.com"	www.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



NOTE: See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about configuring and deploying transparent proxy services.

- Related Documentation**
- [Media Flow Activate Overview on page 3](#)
 - [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
 - [Service Design Overview on page 92](#)
 - [Network Optimization Services Overview on page 97](#)
 - [Creating Network Optimization Service XML Files for Import on page 105](#)
 - [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Creating Network Optimization Service XML Files for Import

Use the following XML structure to create transparent proxy forms that can be imported to Media Flow Activate. These forms are then available for inclusion in a website service provisioned to selected Media Flow Controllers.

Service Network Optimization “Content Direct” XML Schema

When you customize the following XML structure for your needs, give the **contentDirectDefinition** element a **name** and **description**, then modify the options and elements. These parameters are described in [Table 5 on page 106](#). See the following simple XML schema for transparent proxy service. For information about a schema with a cache-tuning policy and virtual player configuration, see [“Sample XML Schema for Network Optimization and Reverse Proxy Services” on page 179](#).

Simple XML Schema for Transparent Proxy Service

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<contentDirectDefinitionCatalog>
  <contentDirectDefinitions>
    <contentDirectDefinition>
      <name>site1</name>
      <description>test</description>
      <properties>
        <domain>google</domain>
        <isDomainRegex>>false</isDomainRegex>
        <match>
          <path>
            <allFiles>>true</allFiles>
          </path>
        </match>
        <precedence>1</precedence>
        <clientRequest>
          <cacheIndex>
            <urlToMatch>urlToMatch</urlToMatch>
            <mapTo>mapTo</mapTo>
            <tunnelUnMatched>>false</tunnelUnMatched>
          </cacheIndex>
        </clientRequest>
        <originServerResolution>USE_DESTINATION_ADDR</originServerResolution>
        <useClientIP>>true</useClientIP>
      </properties>
    </contentDirectDefinition>
  </contentDirectDefinitions>
</contentDirectDefinitionCatalog>
```

```

    <revalidateCacheHit>false</revalidateCacheHit>
  </contentDirectDefinition>
</contentDirectDefinitions>
</contentDirectDefinitionCatalog>

```

Table 5: contentDirectDefinition Options and Elements

Option	Option or Sub-Element	Option or Sub Sub-Element	Description	CLI Equivalent
name	none	none	Enter a name for the contentDirectDefinition element.	namespace <name>
domain	none	none	Enter the name of the domain against which incoming requests are referenced; media files are located in this domain.	namespace <name> domain regex <regex>
match	path	allFiles	Specify that all files in the given domain may be requested. Alternatively, use the specific files option.	namespace <name> match uri regex <regex>
		specificFiles	Specify a regex path to the files that are requested. Alternatively, use the all files option.	namespace <name> match uri <uri-prefix> /
precedence	none	none	The lower the value, the higher the precedence for this namespace, should there be a conflict. See "Understanding Precedence" in "Service Design Overview" on page 92 for more information about using the "precedence" parameter.	namespace <name> match uri regex <regex> [precedence <number>]

Table 5: contentDirectDefinition Options and Elements (*continued*)

Option	Option or Sub-Element	Option or Sub Sub-Element	Description	CLI Equivalent
clientRequest	cacheIndex	urlToMatch	Configure a regex expression to match the request URL. The regex value for url to match has a maximum character limit of 1024 characters (including NULL); if the URI exceeds this limit, the request is tunneled. Only one url to match expression per service is allowed. No PCRE regex is allowed; only GNU regex is allowed	namespace <name> delivery protocol http client-request cache-index url-match <regex> map-to <map_string> no-match-tunnel
		mapTo	Configure a string to map, or rewrite, the URL portion of the request when a match is found. The mapTo value has a maximum character limit of 2048 characters (including NULL).	
		tunnelUnMatched	Tunnel the request when no regex match for url to match is found.	
originServerResolution	USE_HOST_HEADER	none	Use the host header of the incoming request as the origin server (in case of a cache miss). Optionally, set use client IP to use the client IP address in place of the origin server's IP address in the request.	namespace <name> origin-server http follow header <header>
	USE_DESTINATION_ADDR	none	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.	

Table 5: contentDirectDefinition Options and Elements (*continued*)

Option	Option or Sub-Element	Option or Sub Sub-Element	Description	CLI Equivalent
useClientIP	none	none	Use the client IP address in place of the origin server's IP address in the request.	namespace <name> origin-server http follow header <header> [use-client-ip] or namespace <name> origin-server http follow dest-ip [use-client-ip]
tunnelAll	none	none	Tunnel all incoming client requests directly to the origin server without further processing.	namespace <name> delivery protocol http client-request tunnel-all
revalidateCacheHit	none	none	Set requested objects in cache to always trigger a timestamp revalidation. When this parameter is enabled, performance is impacted because every transaction is revalidated.	namespace <name> delivery protocol http client-request cache-hit action revalidate-always

Some **url to match** regex examples:

- Regex **url to match** 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
```

- Regex **url to match** 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
```

- Regex **url to match** 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** 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:

`$$$1$$ => $/videoplayback?xxx$`

See these topics for additional information:

- [Sample XML Schema for Network Optimization and Reverse Proxy Services on page 179](#)
- [Creating Network Optimization Services on page 98](#)
- [Service Design Overview on page 92](#)
- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

See the *Juniper Networks Media Flow Controller Administrator's Guide* for Media Flow Controller CLI command descriptions.

After you have customized and saved a Transparent Proxy XML configuration file, go to the Service Design > Network Opt Services page. On the **Actions** list, click **Import Service**. From the **Import Service** dialog box, navigate to the XML file you created and click **Ok**. The Transparent Proxy configuration provided in the XML file is made available to Media Flow Activate and can be included in service designs for provisioning to selected Media Flow Controllers through the **Service Provisioning** workspace.

CHAPTER 11

HTTP Reverse Proxy Service

- [HTTP Reverse Proxy Services Overview on page 111](#)
- [Creating HTTP Reverse Proxy Services on page 112](#)
- [Creating Reverse Proxy Service XML Files for Import on page 123](#)
- [Creating Reverse Proxy Service XML Files for Export on page 132](#)
- [Purging Content from Media Flow Controller Devices on page 133](#)

HTTP Reverse Proxy Services Overview

This topic describes what HTTP reverse proxy services are used for and what information you need to know before creating a service.

When you configure Media Flow Controller for HTTP reverse proxy services, it reduces network and CPU load on the origin server by serving previously retrieved content, which enhances user experience by decreasing latency.

The website is referenced via the URL in the HTTP request to Media Flow Controller.



NOTE: See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about reverse proxy deployments.

Related Documentation

- [Creating HTTP Reverse Proxy Services on page 112](#)
- [Provisioning Services Overview on page 141](#)
- [Creating Reverse Proxy Service XML Files for Import on page 123](#)
- [Creating Reverse Proxy Service XML Files for Export on page 132](#)
- [Purging Content from Media Flow Controller Devices on page 133](#)
- [Actions on Services on page 137](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Creating HTTP Reverse Proxy Services

Use the **Service Design** workspace to create content delivery services. You create these services for the types of media you are delivering.

Before you begin configuring an **HTTP Reverse Proxy** website media delivery service, you need the following:

- Regular expression for **Domain (Regex)** that is used in requests for the media for this service (See [Table 4 on page 104](#) for example domain regex values.)
- Criterion to be matched with the incoming URL to identify the request. This criterion can be "URI Path, Header name, or Query String."
- **Precedence** you want to give this service among all the configured services (for the same domain)
- For cache misses (requests for media not in the cache), how you want the service to obtain the requested media (by configuring the origin server settings).
- **Virtual Player** you use to manage video delivery policies for that site (See ["Creating Virtual Players" on page 83](#) to create virtual players.)
- **Cache Tuning Policy** you use to fine-tune the cache settings based on the network conditions and caching requirements (See ["Creating Cache-Tuning Policies" on page 52](#) to create cache-tuning policies.)
- **Policy Script** you use to customize the various features in Media Flow Controller at runtime
- Order in which the objects must be ingested from RAM cache to disk cache
- Dynamic URI parameters, which Media Flow Controller uses to identify dynamic URIs and deliver correct content
- Consider whether to:
 - Authenticate the origin servers before downloading the content
 - Monitor the HTTP and HTTPS transactions handled by Media Flow Controller
 - Classify the traffic by setting the DiffServ code point (DSCP) value
 - Pin the objects to the cache
 - Modify request and response headers
 - Ignore the validation headers when revalidating an object
 - Use a custom header to determine cache expiration
 - Send a "503 Service Unavailable" response when Media Flow Controller is unable to service a request because the resource pool concurrent session limit has been reached or exceeded

To configure HTTP reverse proxy services on the **Service Design** workspace:

1. From the left navigation pane, click the plus sign (+) adjacent to **Service Design**.
2. Click the plus sign (+) adjacent to **HTTP Reverse Proxy Services**.
3. Click **Add Service Instance**. The **Add Service Instance** page is displayed.
4. On the **General** tab, specify the following information:

Figure 6: Create HTTP Reverse Proxy Service—General Tab

Add Service Instance

General | Design Elements | Object Handling | Dynamic URI | Origin Headers | Client Headers

Service Name: !

Description:

Domain: ! ☐ Regex

Order of serving client request:

Match Conditions

☐ Regex

☒ URI Path:

☐ Query String Match Name: Value:

☐ Header Match Name: Value:

Precedence:

Origin Server

☒ FQDN ☐ Origin map

Server Name: !

Port: !

Secure Authentication: ☐

- **Service Name**—Enter the name of the service or website that you want to configure, which must be unique.
- **Description**—(Optional) Enter a description of the service or website.
- **Domain**—Enter the defined fully qualified domain name (FQDN) or regular expression that is matched with the incoming HOST header to identify the request.
Select **Regex** if you want the MFA to treat any entries in the **Domain** field as regex entries. Enclose all regex entries in double quotation marks.
- **Order of serving client request**—Enter the policy that determines the order in which the objects are ingested from RAM cache into disk cache when Media Flow Controller handles bulk requests.

To ensure that clients do not need to wait for data while Media Flow Controller ingests data into disk cache, the data serving path and the ingestion path are decoupled in Media Flow Controller. During the time that objects are being served from RAM cache, the ingestion process in Media Flow Controller picks these objects from RAM cache and ingests them into disk cache. The order in which the ingestion process picks these objects for ingestion can be tuned:

- **Last in first out**—(Default) The object served last is ingested into disk cache first. Because the object is selected for ingestion immediately after it is served to the client, the probability of finding the object in RAM cache is high. This prevents the object from being refetched from the origin server.
- **First in first out**—The object served first is ingested into disk cache first. The objects are ingested into disk cache in the order in which they are served to the clients from RAM cache. However, this feature may introduce additional fetches from the origin server if the objects are evicted from RAM cache before they are ingested into disk cache.



NOTE: Recommendations: For caching adaptive bit rate streaming videos (such as Microsoft Smooth Streaming videos or Apple HTTP Live Streaming videos), it is recommended that you set the configuration to **First in first out**. For Network Optimization service deployments, it is recommended that you set the configuration to **Last in first out**.

Example for “First in first out”: Consider that in a given instant (for example, at t_0 seconds), Media Flow Controller receives 100 requests for 100 objects. After Media Flow Controller fetches these objects from the origin server, the order in which these objects are ingested into disk cache is determined by whether Media Flow Controller is configured for **First in first out** or **Last in first out**. If it is configured for **First in first out**, then it ingests these objects into disk cache starting from the first object that was served to the client, followed by the second object, and so on. At t_1 seconds, if Media Flow Controller receives another 100 new requests and if it has cached only 80 objects at the end of t_0 seconds, the cacheable disk queue becomes 120 objects and it starts caching from the eighty-first object. At t_2 seconds, if Media Flow Controller receives another 100 new requests and if it has cached only 160 objects at the end of t_1 seconds, the cacheable disk queue now becomes 140 objects and it starts caching from the hundred and sixty-first object.

- In the **Match Conditions** area, specify the criteria to be matched with the incoming URL to identify the request. The request is serviced only when there is a match. It provides a finer control over the requests that are serviced by Media Flow Controller.

When there is a match, the request is attached to the specific service and the requested object is delivered as per the configured service parameters. If no matching services are found (which can mean that the requested object may not be in the cache), the requests are terminated.

This feature is useful in scenarios where you have multiple services defined for a website and you want to direct the request to the service that is most relevant for

the request (such as where you might have configured the virtual player and other parameters that are most suitable to cater to that specific content).

For example, you might have created two reverse proxy services for the domain www.cnn.com to cater to videos and images. Assume that you have created the service *cnnv*, which caches the videos under the *cnn/videos* directory structure. You can specify an additional match criterion of */videos* in the *URI Path* field in the Match Conditions area, so that any incoming request such as www.cnn.com/videos/roundtheworld.flv is attached to the “cnnv” service and is processed by this service.

Specify at least one of the following match conditions:

Select **Regex** if you want Media Flow Controller to treat the entries in the following fields as regex entries.

- **URI Path**—Enter a path to be matched in the incoming request URL. You can enter a string or a regular expression. For example, “/videos.” By default, this option is selected and has a value of “/”, which means that any incoming request with the configured domain name followed by a “/” is matched to this namespace. For example, if the domain name configured for a namespace is “www.yahoo.com,” any incoming request URL containing the string “www.yahoo.com/” is matched and served by this namespace.

- **Query String Match Name**—Enter a query string parameter (name-value pair) to be matched in the incoming request URL.

For example, if you enter “*video_id*” and “1”, this service is associated with an incoming URL request such as www.cnn.com/videos/get_video?video_id=1.

- **Header Match Name**—Enter a header to be matched in the incoming request.
- **Precedence**—When there are multiple services defined for the same domain, specify a number to determine which service must be given higher precedence. The lower the number, the higher the preference for that service; 0 is the default and the highest.

For example, you might have configured two services: one that stores any generic video and one that stores videos that are specific to the year 2011 (the directory structure in Media Flow Controller would be something like */videos* and *videos/2011*). If the incoming request is for a video created in 2011, because it matches both these services, the service that has the higher precedence is associated with the request. In this case, the service that stores the 2011 videos must be configured with the higher precedence.

- In the **Origin Server** area, you configure the origin servers that Media Flow Controller must access to fetch content upon a cache miss. You must select either FQDN or Origin Map.
 - Select **FQDN** to configure a single origin server:
 - **Server Name**—Enter the IP address or the server name of the origin server.
 - **Port**—Enter the port number through which the server listens for requests. Usually, because the server listens at port 80, you can enter **80** in this field.

However, if you want to authenticate the origin servers before downloading the content from the origin servers, enter **443**. For more information about authenticating origin servers, see the information provided for the **Secure Authentication** check box.

- Select **Origin map** to configure multiple HTTP origin servers.

Select an origin map from the **Name** list. For more information about creating origin maps, see [“Creating Consistent Hash Maps” on page 69](#) and [“Creating Escalation Maps” on page 72](#).

- Select the **Secure Authentication** check box to authenticate the origin servers before downloading content from the origin servers. If you select this check box, Media Flow Controller encrypts the authentication messages and fetches content from the origin servers by using HTTPS. If you do not select this check box, Media Flow Controller does not authenticate the origin servers and fetches content in plain text by using HTTP.

You use this feature when you want to authenticate and download content from origin servers that reside within nontrusted domains.

For the **Secure Authentication** feature to work in Media Flow Controller:

- Set the port to 443.

If you are using the FQDN option to associate an origin server with the service, enter 443 in the **Port** field. If you are associating an origin map with the service, select the origin map that is configured with port number 443 from the **Name** list.

- Configure the ciphers and CA certificates in Media Flow Controller.

For instructions to configure CA ciphers and CA certificates, see the “Setting Up Content Ingest Manager Security” section in the *Juniper Networks Media Flow Controller Administrator's Guide*.

- Configure **Enable SSL Authentication** from **Media Flow Devices > Device Configuration** for Media Flow Controller.

When you enable the **Secure Authentication** feature:

- Media Flow Controller establishes a TCP connection on port 443 with the origin server.
- The SSL handshake is performed. During the handshake, the origin server sends its digital certificate to Media Flow Controller.
- Media Flow Controller uses the information sent by the server to authenticate the server. If the server is successfully authenticated, Media Flow Controller forwards the client request to the server.

5. On the **Design Elements** tab, specify the following information:

- **Virtual Player Name**—Select a previously defined virtual player with the service; see [“Creating Virtual Players” on page 83](#). The virtual player configuration determines how the videos within the specified domain are delivered.
- **Cache Tuning Policy**—Select a previously defined policy with the service; for more information about creating a cache-tuning policy, see [“Creating Cache-Tuning Policies” on page 52](#). The policy determines when to promote an object to various cache tiers.
- **Log Profile**—Select a log profile with the service from a list of previously created log profiles.

You associate a log profile with a service when you want to monitor the HTTP and HTTP transactions handled by that service. This information is captured on the log file that is configured in the log profile. Typically, the log file is stored in the LogExport directory. For more information about access log profiles, see [“Understanding Access Log Profiles” on page 41](#).



CAUTION: Though MFA supports creating any number of log profiles, the provisioning of the service fails if the log profile associated with the service exceeds the maximum number of log profiles that a Media Flow Controller can support (which is 32).

- **Policy Script**—Select a policy script from a list of previously added policy scripts.

You typically associate a policy script when you want a fine granular control over the various features of Media Flow Controller at runtime.

Make sure that the policy script file is available on the local system from where Junos Space is accessed.

6. On the **Object Handling** tab, specify the following information:

- **DSCP**—Specify a DSCP value so that Media Flow Controller sets the DSCP field of the IPv4 packets to this value for all HTTP responses sent from Media Flow Controller to the client for this service.

You can enter a value from 0 through 63. The default value is 0 (zero).



NOTE: DiffServ provides a mechanism to classify and mark packets belonging to a specific class. This mechanism proves helpful if you have configured a router to differentiate traffic on the basis of class, which can then be used to provide low latency to critical network traffic such as voice or streaming media, while providing a simple best-effort service to noncritical services such as Web traffic or file transfers.

- **Tunnel All**—Select this check box to tunnel all incoming client requests directly to the origin server without further processing. By default, this feature is disabled.

When you enable this feature, all caching-related configuration is ignored and the requests are tunneled directly to the origin servers.

- **Ignore all object correlation validators**—Select for Media Flow Controller to ignore the object validation fields (such as ETag and Last Modified headers) in the origin server response when it validates an object.

When the ETag or the Last Modified header is different from the previously cached origin server response, by default, Media Flow Controller assumes that the object is of a different version and deletes the cached object and replaces it with the modified object. If you select the **Ignore all object correlation validators** check box, Media Flow Controller ignores any changes to these headers in the origin server response and serves the previously cached object, provided that it has not expired. However, if the object has expired, Media Flow Controller deletes the existing cached object, caches the object fetched from the origin server, and serves it.

Example: Media Flow Controller uses the ETag header or the Last Modified header in the origin server response to validate whether the object that is currently cached is of the same version as that in the origin server. If the object has been modified in the origin server, then the ETag header value or the Last Modified header value sent by the origin server is different from the value that was previously sent by the origin server. If the values are different, Media Flow Controller deletes the existing cached content, fetches the modified content from the origin server, and caches it.

At the time of caching the origin server response, Media Flow Controller associates a version number with each object that is cached. This version number is generated from the ETag header value of the origin server response, provided that the ETag header is present; otherwise, from the Last Modified header value, provided that the Last Modified header is present. If both are absent, then Media Flow Controller generates a random version number. Though this is a good mechanism for validating the freshness of the object, you may find that in CDN deployments, the objects might not be cached at all in Media Flow Controller when the client is making byte-range requests.

Scenario 1: If load balancers are set up, where multiple origin servers might respond to byte-range requests, it is possible that the ETag header values are different in the responses sent by each of these origin servers. Consider a client requesting a 10-MB object in byte-range requests of 2 MB. For the first request of 0 to 2MB, if there is a cache miss, Media Flow Controller forwards this request to the origin server. If origin server 1 (OS1) responds to this request, Media Flow Controller caches this response and also generates a version number from the ETag header value of the response and associates it with the object. For the next request of 2 to 4 MB, upon a cache miss, if origin server 2 (OS2) responds, it is likely that the ETag header sent by this origin server might be different. At the time of caching, Media Flow Controller assumes that this is a different version of the object and deletes the previously cached 0 to 2 MB. Media Flow Controller tunnels the 2- to 4-MB response in addition because it cannot cache this content without caching the previous 0 to 2 MB. The remaining byte-range requests are also tunneled to the client. Media Flow Controller does not cache the entire 10-MB object. This is one scenario where even though the response is cacheable, Media Flow Controller does not cache the response.

Scenario 2: Assume that the origin server does not send the ETag or the Last Modified header in its response to the byte-range requests. In this scenario, consider that a

client is requesting a 10-MB object in byte-range requests of 2 MB. When Media Flow Controller caches the first request of 0 to 2 MB, it generates a random version number and associates it with the cached object. This is because there is no ETag or Last Modified header in the origin server response to generate the version number. When Media Flow Controller tries to cache the next request of 2 to 4 MB, it generates another random version number to associate with the object. At the time of caching, Media Flow Controller compares the version numbers of 0 to 2 MB and 2 to 4 MB of the object. Because the version numbers are different, Media Flow Controller deletes the cached 0 to 2 MB and tunnels the 2 to 4 MB of the object directly to the client. The remaining byte-range requests are also tunneled to the client.

- **Custom cache-control header**—Enter a custom cache-control header name in the origin server response to be used for determining cache expiry. Assume that you have configured “xyz” for this field. If the origin server response contains the “xyz” header, then Media Flow Controller sets the expiry time of the object to the value contained in this header. If the xyz header value is 2400, then Media Flow Controller serves the object from its cache for the next 40 minutes from the time of caching. After this duration, the object is considered expired.

Example: Some origin servers do not send the expiry information of an object to intermediate proxies by using the standard HTTP headers, such as the Expires header or the Cache-control:max-age header. Instead, they send it by using custom headers. Media Flow Controller expects the object expiry information from these standard HTTP headers. When these headers are not present in the origin server’s response, Media Flow Controller automatically assumes that the origin server has not sent this information. Media Flow Controller then tries to apply the expiry value configured in the **cache-age default** or **cache-age content-type any** CLIs (that is, the values you configured for **Default Cache Age** or **Cache Age Override** in the cache-tuning policy). However, for Media Flow Controller to use the expiry information present in the custom header of the origin server’s response, you must specify the custom header name in the **Custom cache-control header** field. Media Flow Controller now detects that this custom header contains the expiry information for the object and it proceeds to set the object expiry time to this value.

For example, consider that the origin server sends a custom header, “x-cache-control:max-age=3600.” For Media Flow Controller to detect that the custom header, “x-cache-control,” in the origin server’s response contains the expiry information of the object, you need to set the value of the **Custom cache-control header** field to “x-cache-control.” Media Flow Controller then serves the object from its cache for the next one hour from the time of caching.

- Select the **Cache Pinning** check box to activate the cache pinning options.
 - Select the **Enable Auto Pin** check box. The objects are automatically pinned to the cache.
 - **Pin Header Name**—Enter the header name for cache pinning.

Before you specify a value, it is necessary to know the header name that is sent from the origin server as a response to a request from the caching server. An object is pinned to the cache only when the header name that you configure in this field matches the response header from the origin server.

For example, if you configure the pin header name as *pinnable* in the service when the objects are retrieved from the origin servers, provided that the response headers from the origin servers contain the same term *pinnable*, then the objects are pinned to the cache.

- **Maximum Object Size**—Enter the maximum size of the object, in KB, that can be pinned to the cache. Any object beyond this size is not pinned. The value must be an integer between 0 and 4,294,967,295.
- **Maximum Cache Capacity**—Enter the maximum cache capacity, in GB, to be allocated for caching pinned objects for a specific service. After the capacity is reached, the objects are no longer pinned to the cache. The value must be an integer between 0 and 4,294,967,295.
- **Use Validity Begin Header**—Enter a valid header whose timestamp is used to serve the request.

For example, consider that you set the value to the “VAL-BEGIN”. If the origin response consists of “VAL-BEGIN: Wed, 18 Dec 2010 04:58:08 GMT” header, then the object is pinned but served only after this time.

- **Maximum Cache Capacity (For All Objects)**—Select a cache tier (**SSD**, **SAS**, or **SATA**) and specify the maximum cache capacity, in MB, to be allocated for caching objects in the selected tier. It is recommended that you select the lowest tier for caching the objects. The value must be an integer between 0 and 31,457,280.
7. On the **Dynamic URI** tab, in the **Dynamic URI** area, configure parameters using which Media Flow Controller can identify dynamic URIs and ensure the delivery of the correct content. To accomplish this, an incoming request having a URI that matches the configured regex value (specified in the **URL to Match** field) is matched to a cache index string (specified in the **Map to** field).



NOTE: The URL to Match value must always be configured along with Map To. You can select Tunnel unmatched only if the URL to Match and Map To values are provided.

- **URL to Match**—Configure a regex expression to match 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. PCRE regex is not allowed, whereas GNU regex is allowed. The default value is NULL.

Example: `/get/[^/]+/[^/]+/[^/]+/(.*)`

- **Map to**—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.

Example: `/$1`

- **Use Tunnel unmatched**—Select the check box to tunnel the request when no regex match is found.

8. On the **Origin Headers** tab, in the **Origin request** area, modify the headers in the requests sent from Media Flow Controller to origin servers.
 - **Inherit Host Header**—If you select the check box, Media Flow Controller uses the incoming client request header in its request to the origin server.
 - **Set Host Header**—Upon a cache miss, while forwarding the client request to the origin server, Media Flow Controller by default replaces the host header value sent by the client to the origin server's hostname or the IP address configured in the "Origin Server" area in the GUI. This action ensures that the client request is forwarded to the origin server for fetching the object. However, if you want to override this Media Flow Controller behavior and assign a static domain name or IP address to the host header when the request is forwarded from Media Flow Controller to the origin server, you can set the static IP address or hostname in the **Set Host Header** field. Media Flow Controller uses the value provided in this field for setting the host header instead of using the origin server's IP address or hostname in its request to the origin server.
 - **Add headers to origin request**—From this area, you can add custom headers to the requests forwarded by Media Flow Controller to the origin servers. You can add a maximum of four custom headers.
9. On the **Client Headers** tab, select the **Client response** check box, modify the headers in the responses sent from Media Flow Controller to clients.

An origin server's response to Media Flow Controller's cache-miss request includes headers as well as the requested object. At times, you might not want to forward the cached origin response headers to the client when serving a request. In such scenarios, you can specify what headers must be deleted and, if needed, what headers need to be added in Media Flow Controller responses to the clients. For example, you might want to remove the "Server" header that identifies the origin server from the Media Flow Controller response to the client.

On exceeding resource limits, add retry header to try after—Select this check box and configure the time period (in seconds) in which the clients may resend a request if Media Flow Controller is unable to process the current request because "resource-pool client session limit" is reached. If this parameter is configured, Media Flow Controller adds the "Retry-After:<N>" header in its "503 Service Unavailable" response to the client, so that the client can retry after N seconds.

The resource pool client session limit is derived from the resource pool configuration associated with the service. If the service is not bound to a specific resource pool, then the global resource pool configuration is used and the number of client requests handled by Media Flow Controller for that specific service is determined by this configuration.

Though new connections may be rejected because the resource pool session limits are reached, any existing connections for that service that are handled by Media Flow Controller are gracefully brought to completion.

You can enter a value from 0 through 86,400 seconds. The default value is 0 seconds—that is, the client can request the object immediately after it receives a 503 response from Media Flow Controller.

Example: Assume that a service is bound to a resource pool with a concurrent session limit of 10,000. When Media Flow Controller receives the ten thousand and first request, by default, it rejects this request with a “503 Service Unavailable” response. With this response, the client simply detects that the server is currently unable to handle the request due to a temporary overloading or maintenance of the server. However, when you configure Media Flow Controller to send a “Retry-After:5” header in its “503 Service Unavailable” response, it instructs the client to retry after five seconds.

To delete and add headers, perform the following steps:

1. From the **Delete headers from client response** area, click **Add** and enter the name of the header that you want to delete from the Media Flow Controller response to the client. In this case, add “Server.”
2. From the **Add headers for client response** area, add another header to replace the server that was deleted. In this case, the header name could be “Server” and the value could be “Customized web server.”

You can add and delete a maximum of eight headers. By default, the “via” header is deleted from the Media Flow Controller response to the client. However, if you want to retain it, select the header and click **Remove** from the **Delete headers from client response** area.

10. Click **OK** or **Cancel**. **OK** instantiates the values you set; **Cancel** closes the page without making any changes.



NOTE: See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about configuring a service (namespace).

**Related
Documentation**

- [Provisioning Services Overview on page 141](#)
- [Purging Content from Media Flow Controller Devices on page 133](#)
- [Actions on Services on page 137](#)
- [Creating Reverse Proxy Service XML Files for Export on page 132](#)
- [Creating Reverse Proxy Service XML Files for Import on page 123](#)
- [HTTP Reverse Proxy Services Overview on page 111](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Creating Reverse Proxy Service XML Files for Import

Use the following XML structure to create reverse proxy forms that can be imported to Media Flow Activate. These forms are then available for inclusion in a website service that is provisioned to selected Media Flow Controllers.

After you have customized and saved a Reverse Proxy XML configuration file, go to the Service Design > HTTP Reverse Proxy Service page. On the **Actions** list, select **Import Service**. From the **Import Service** dialog box, navigate to the XML file that you created and click **Ok**. The Reverse Proxy configuration provided in the XML file is made available to Media Flow Activate and can be included in service designs for provisioning to selected Media Flow Controllers through the **Service Provisioning** workspace.

Reverse Proxy Service XML Schema

When you customize the following XML file structure for your needs, give the **rproxyDefinition** element a **name** and **description**, then modify the options and elements. These elements are described in [Table 6 on page 125](#). See the following simple XML schema for an HTTP reverse proxy service. For a complex XML schema, see [“Sample XML Schema for Network Optimization and Reverse Proxy Services” on page 179](#).

Simple XML Schema for Reverse Proxy Service

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rproxyDefinitionCatalog>
  <rproxyDefinitions>
    <rproxyDefinition>
      <name>site1</name>
      <description>Test RProxy</description>
      <properties>
        <domain>hotmail</domain>
        <isDomainRegex>>false</isDomainRegex>
        <match>
          <queryStringMatch>
            <name>name</name>
            <value>value</value>
          </queryStringMatch>
        </match>
        <precedence>1</precedence>
        <clientRequest>
          <cacheIndex>
            <urlToMatch>/google/cbsgaruer374/pics</urlToMatch>
            <mapTo>/google/pics</mapTo>
            <tunnelUnMatched>>false</tunnelUnMatched>
          </cacheIndex>
        </clientRequest>
        <originServer>
          <fqdn>
            <address>
              <IPAddress>10.21.23.45</IPAddress>
              <port>5078</port>
            </address>
          </fqdn>
          <useSecureMode>>true</useSecureMode>
        </originServer>
      </properties>
    </rproxyDefinition>
  </rproxyDefinitions>
</rproxyDefinitionCatalog>
```

```
</originServer>
<cachePinning>
  <enableAutoPin>>false</enableAutoPin>
  <pinHeaderName>TestPinHeaderName</pinHeaderName>
  <maxObjectSizeInKB>500</maxObjectSizeInKB>
  <maxCacheCapacityInGB>1500</maxCacheCapacityInGB>
  <validityBeginHdr>valheader22</validityBeginHdr>
</cachePinning>
<maximumCacheCapacity>
  <cacheTier>SAS</cacheTier>
  <sizeInMB>234</sizeInMB>
</maximumCacheCapacity>
<clientResponse>
  <headersToDelete>
    <value>h3</value>
    <value>h4</value>
  </headersToDelete>
  <headersToAdd>
    <nameValue>
      <name>h1</name>
      <value>v1</value>
    </nameValue>
    <nameValue>
      <name>h2</name>
      <value>v2</value>
    </nameValue>
  </headersToAdd>
  <dscp>44</dscp>
</clientResponse>
<originRequest>
  <inheritHostHeaderValue>true</inheritHostHeaderValue>
  <headersToAdd>
    <nameValue>
      <name>h12</name>
      <value>v12</value>
    </nameValue>
    <nameValue>
      <name>h22</name>
      <value>v22</value>
    </nameValue>
  </headersToAdd>
  <setHostHeader>headerrVal</setHostHeader>
</originRequest>
</properties>
<tunnelAll>true</tunnelAll>
</rproxyDefinition>
</rproxyDefinitions>
</rproxyDefinitionCatalog>
```

Table 6: rproxyDefinition Options and Elements

Option	Option or Sub-Element	Option or Sub-Element	Description	CLI Equivalent
name	none	none	Enter a name for the rproxyDefinition element.	namespace <name>
domain	none	none	Enter the name of the domain against which incoming requests are referenced; the media files are located in this domain.	namespace <name> domain regex <regex>
match	regexPath	none	Specify a path to be matched in the incoming request URL.	namespace <name> match uri (regex <regex> <uri-prefix>)[precedence number]
	queryStringMatch	name	Specify a query-string parameter (name-value pair) to be matched in the incoming request URL.	namespace <name> match query-string (regex <regex> <name> (any <value>)) [precedence number]
		value		
	regexQueryStringMatch	none		
	headerMatch	name	Specify a header to be matched in the incoming request.	namespace <name> match header (regex <regex> <name> (any <value>)) [precedence number]
		value		
	regexHeaderMatch	none		
	path	allFiles	Specify that all files in the given domain may be requested. Alternatively, use the specific files option.	namespace <name> match uri regex <regex>
		specificFiles	Specify a regex path to the files that are requested. Alternatively, use the all files option.	namespace <name> match uri <uri-prefix> /

Table 6: rproxyDefinition Options and Elements (*continued*)

Option	Option or Sub-Element	Option or Sub-Element	Description	CLI Equivalent
precedence	none	none	The lower the value, the higher the precedence for this namespace, should there be a conflict. See “Understanding Precedence” in “Service Design Overview” on page 92 on page 70 for more information about configuring the “precedence” parameter.	namespace <name> match uri (regex <regex> <uri-prefix> [precedence number] or namespace <name> match query-string (regex <regex> <name> (any <value>)) [precedence number] or namespace <name> match header (regex <regex> <name> (any <value>)) [precedence number]

Table 6: rproxyDefinition Options and Elements (*continued*)

Option	Option or Sub-Element	Option or Sub-Element	Description	CLI Equivalent
clientRequest	cacheIndex	urlToMatch	Configure a regex expression to match the request URL. The regex value for url to match has a maximum character limit of 1024 characters (including NULL); if the URI exceeds this limit, the request is tunneled. Only one url to match expression per service is allowed. No PCRE regular expression is allowed; only GNU regular expression is allowed.	namespace <name> delivery protocol http client-request cache-index url-match <regex> map-to <map_string> no-match-tunnel
		mapTo	Configure a string to map or rewrite the URL portion of the request when a match is found. The mapTo value has a maximum character limit of 2048 characters (including NULL).	
		tunnelUnMatched	Tunnel the request when no regex match for url to match is found.	

Table 6: rproxyDefinition Options and Elements (*continued*)

Option	Option or Sub-Element	Option or Sub-Element	Description	CLI Equivalent
originServer	address	IPAddress	Specify the origin server IP address that Media Flow Controller must access to fetch content upon a cache miss.	namespace <name> origin-server http <hostname/IP address>/<port>
		port	Specify the port through which the server listens for any requests.	
	originMap		Specify the name of the origin map to be used with this reverse proxy service configuration.	namespace <name> origin-server http server-map <map_name>
	useSecureMode	none	Authenticate the origin servers before downloading content from the origin servers.	namespace <name> delivery protocol http origin-request secure

Table 6: rproxyDefinition Options and Elements (*continued*)

Option	Option or Sub-Element	Option or Sub-Element	Description	CLI Equivalent
cachePinning	enableAutoPin	none	Specify whether the objects should be automatically pinned to cache.	namespace <name> pinned-object auto-pin
	pinHeaderName	none	Specify the header name for pinning to cache.	namespace <name> pinned-object pin-header <name>
	maxObjectSizeInKB	none	Specify the maximum size of the object that can be pinned (in KB).	namespace <name>pinned-object max-obj-size KB
	maxCacheCapacityInGB	none	Specify the maximum size, in GB, that can be allocated for storing pinned objects for a namespace.	namespace <name> pinned-object cache-capacity GB
	validityBeginHdr	none	Specify whether to use the header timestamp to determine whether an object can be delivered when a request to the object has been received.	namespace <name> object validity-begin-header
maximumCacheCapacity	cacheTier	none	Specify a cache tier (SSD, SAS, or SATA) and specify the maximum cache capacity, in MB, to be allocated for caching objects in the selected tier. It is recommended that you select the lowest tier for caching the objects.	namespace <name> media-cache disk cache-tier [sas (free-block-threshold <number> group-read (enable disable) read-size <number>) sata (free-block-threshold <number> group-read (enable disable) read-size <number>) ssd (free-block-threshold <number> group-read (enable disable) read-size <number>)]
	sizeInMB	none		

Table 6: rproxyDefinition Options and Elements (*continued*)

Option	Option or Sub-Element	Option or Sub-Element	Description	CLI Equivalent
clientResponse	headersToDelete	value	Enter the name of the header that should be deleted from the outgoing response—that is, from Media Flow Controller to the client.	namespace <name> delivery protocol http client-response header <name> [<value>] action delete
	headersToAdd	name and value	Enter the name of the header that should be added to the outgoing response—that is, from Media Flow Controller to the client.	namespace <name> delivery protocol http client-response header <name> [<value>] action add
	dscp	none	Media Flow Controller sets the DSCP field of the IPv4 packets by using the value specified for all the HTTP responses sent from Media Flow Controller to the client for this service.	namespace <name> delivery protocol http client-response dscp <number>

Table 6: rproxyDefinition Options and Elements (*continued*)

Option	Option or Sub-Element	Option or Sub-Element	Description	CLI Equivalent
originRequest	inheritHostHeaderValue	none	Allow 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.	namespace <name> delivery protocol http origin-request host-header inherit incoming-req (deny permit)
	headersToAdd	name and value	Enter the name of the header that should be added to the incoming request.	namespace <name> delivery protocol http origin-request header <name> [<value>] action add
	setHostHeader	none	Media Flow Controller uses the value provided for setting the host header instead of using the origin server's IP address or hostname in its request to the origin server.	namespace <name> delivery protocol http origin-request host-header set <header-value>
tunnelAll	none	none	Select to tunnel all incoming client requests directly to the origin server without further processing.	namespace <name> delivery protocol http client-request tunnel-all



NOTE: See the *Juniper Networks Media Flow Controller Administrator's Guide* and *Juniper Networks Media Flow Controller CLI Command Reference* for Media Flow Controller CLI command descriptions.

See these topics for additional information:

- [Sample XML Schema for Network Optimization and Reverse Proxy Services on page 179](#)
- [Provisioning Services Overview on page 141](#)
- [Actions on Services on page 137](#)
- [Creating HTTP Reverse Proxy Services on page 112](#)
- [Creating Reverse Proxy Service XML Files for Export on page 132](#)

- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Creating Reverse Proxy Service XML Files for Export

Reverse Proxy Service XML Schema

You can export a created service as an XML file. You can use your preferred XML editor to make the necessary changes and later import the XML file back to Media Flow Activate. You can then reprovision the service on Media Flow Controller with the updated configuration. However, if you want to import this configuration on another MFA server, you have to remove the service ID information before the import.



CAUTION: Server map information cannot be imported on a different Media Flow Activate server because it contains the device ID information that is specific to the Media Flow Activate server.

After you have customized and saved a Reverse Proxy XML configuration file, go to the Service Design > HTTP Reverse Proxy Services page, and select the service, which you had recently customized. From the **Actions** list, select **Export Service**. In the **Export Service** dialog box, navigate to the XML file you created and click **Yes**. Save the file to a suitable location. Because the filename is saved with the service ID rather than the service name, rename the file with a suitable name and save it with an xml extension for your later use.



NOTE: See the *Juniper Networks Media Flow Controller Administrator's Guide* and *Juniper Networks Media Flow Controller CLI Command Reference* for Media Flow Controller CLI command descriptions.

See these topics for additional information:

- [Creating HTTP Reverse Proxy Services on page 112](#)
- [Creating Reverse Proxy Service XML Files for Import on page 123](#)
- [HTTP Reverse Proxy Services Overview on page 111](#)
- [Service Design Overview on page 92](#)
- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Purging Content from Media Flow Controller Devices

You can purge all or specific content, which you no longer need to cache, from selected Media Flow Controller devices.

To purge content from selected Media Flow Controller devices:

1. From the left navigation panel, click the plus sign (+) adjacent to **Service Design**.
2. Click **HTTP Reverse Proxy Services**. The **HTTP Reverse Proxy Services** page is displayed.
3. Select the services for which you want to purge content. From the **Actions** list, select **Purge Content**. The **Purge Content** configuration page is displayed.
4. In the **Content to Purge** area, select one of the following options:

- **All**—Purge all content.
- **Directories**—Enter a regular expression. Enter a URL until the folder name. For example, <http://abc.com/videos>.

You may choose to purge directories that exactly match this value or at least contain a part of this value.

- **Files**—Enter a regular expression. Enter a URL that includes the filename. For example, <http://abc.com/videos/hello.flv>.

You may choose to purge files that exactly match this value or at least contain a part of this value.

- Select the **Soft Purge** check box. You can select this option to revalidate the content with the origin server instead of deleting content from Media Flow Controllers. Select this option only if you have selected the **All** option.
5. In the **Select Devices** area, select the Media Flow Controllers.
 6. Click one of the following buttons:
 - **Delete**—Proceed with the purge and close the page. The specified content from the selected Media Flow Controllers is purged.



CAUTION: The content is not purged if you have configured a query string or a header in the HTTP reverse proxy service to be matched in the incoming request URL.

- **Cancel**—Cancel the purge and close the page.

Related Documentation

- [HTTP Reverse Proxy Services Overview on page 111](#)
- [Service Design Overview on page 92](#)
- [Provisioning Services Overview on page 141](#)
- [Media Flow Activate Overview on page 3](#)

- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

CHAPTER 12

Content Ingest Service

- [Content Ingest Services Overview on page 135](#)
- [Creating Content Ingest Services on page 135](#)
- [Actions on Services on page 137](#)

Content Ingest Services Overview

With the content ingest service (crawler), you can preload content from origin servers at predefined time intervals. The crawler searches parent node—that is, Media Staging Servers (MSSs) or other origin servers—for content that can be downloaded to Media Flow Controllers. (MSSs are repositories where service providers store permanent content.) The content on parent nodes are then crawled by edge nodes running the standard Media Flow Controller software.

The crawler discovers the requested URLs at the origin server, downloads the discovered URL content from the origin server, and caches the content on Media Flow Controllers.

Related Documentation

- [Creating Content Ingest Services on page 135](#)

Creating Content Ingest Services

Use the **Service Design** workspace to create content delivery services. You create these services for the types of media you are delivering.

Before you begin configuring a **Content Ingest** media delivery service, you need the following information:

- Protocol
- Origin server
- Link depth
- Crawler schedule
- Content types

To configure content ingest services:

1. From the left navigation panel, click the plus sign (+) adjacent to **Service Design**.
2. Click the plus sign (+) adjacent to **Content Ingest Service**.
3. Click **Add Service Instance**. The **Add Service** page is displayed.
4. On the **Basic Properties** tab, specify the following information:
 - **Name**—Enter the name of the crawler instance.
 - **Description**—(Optional) Enter the description of the crawler instance.
 - **Protocol**—Enter the protocol to fetch content.
 - **Origin Server URL**—Enter the complete URL of the origin server including the domain name, port number, and path from where the content must be downloaded (for example, www.foo.com:8080/video/).

The origin for parent caches is the Media Staging Servers (MSSs) or third-party origin servers. The origin for edge caches is always the parent cache.

 - **Link Depth**—Enter an integer that specifies the level of directory or links that the crawler needs to follow to obtain the content. A selected link depth of 1 means that the crawler needs to follow the link or directory one level down.

You can enter a value from 0 through 10. The default value is 10.

 - In the Crawler Schedule area, specify the following information:
 - **Start Time**—Enter time in GMT at which the crawler instance should be started. The crawler begins crawling at the specified start time.
 - **Stop Time**—Enter time in GMT after which the new crawler instances are not started. If a crawling operation is in progress after the stop time is reached, the crawling operation is gracefully completed.
 - **Refresh Interval**—Enter interval at which the crawler refreshes content from the origin server. Specify the time in minutes. If a stop time is not specified, the crawler automatically continues to crawl at the specified refresh intervals.- 5. On the **Content Types** tab, you can set the crawl file types that the crawler must accept for the particular instance. The maximum number of extensions that can be added is 10. The default is to download or preload objects that match the extension.
 - Click **Add** to add the content types. The **Select File Extensions** dialog box is displayed.
 - In the **Available** pane of the dialog box, double-click the content types you want the crawler to accept. Each double-clicked content type moves to the **Selected** pane. Click **OK**. You are returned to the **Content Types** tab. You can view the selected file extensions in the **Accept Content Types** area.



NOTE: You can also add new content types. To add new content types, enter a content type in the **Add File Type** field and click **Add**.

- By default, **Preload** is set to **Yes**. If you do not want to preload the content, set this value to **No**.
- You can sort the data by clicking the arrow next to the column name.
- Click **Remove** to remove a selected content type.
- **Generate ASX file for every wmv file**—By default, this check box is selected, which means that whenever the crawler finds a .wmv file, a corresponding .asx file is auto-generated. Clear this check box if you do not intend to cache any .wmv files. All the applications that need to be delivered via MS-WMSP need the .asx file. The .asx file specifies the application domain and path pointing to the target video.



NOTE: The ASX file contains data about the actual media file (ASF file), containing video, audio, and so on. The purpose of an ASX file is to start the ASF file streaming.

6. Click **OK**, **Cancel**, or **Reset**. **OK** instantiates the values you set, **Cancel** closes the dialog box, and **Reset** returns all values to their defaults.

Related Documentation

- [Content Ingest Services Overview on page 135](#)
- [Actions on Services on page 137](#)

Actions on Services

From the **Content Ingest Services**, **Network Optimization Services**, or **Rproxy Services** page, you can perform some or all of the following actions on services by clicking the links on the **Actions** list. You have to select the services before performing any actions on them:

- **Modify Service**—Click this link to modify all configuration settings other than the service name.

You can modify only one service at a time.

- **Delete Service(s)**—Click this link to delete one or more services. Deletion removes the services totally from Media Flow Activate.

If the service is provisioned, deprovision the service from all the devices that are running the service before deleting the service. For more information about deprovisioning a service, see [“Managing Provisioned Services” on page 144](#).

- **Copy Service**—Click this link to create a duplicate of the service with a different name. When you select this action, you are prompted for a name for the new service.
- **Export Service**—Click this link to export a created service as an XML file.

You can use your preferred XML editor to make the necessary changes to the XML file and later import it back to Media Flow Activate. You can then reprovision the service to Media Flow Controller with the updated configuration. However, if you want to import this configuration to another Media Flow Activate server, you have to remove the service ID information before the import.

- **Import Service**—Click this link to import a service as an XML file and provision the service on to Media Flow Controllers.
- **Manage Provisioned Devices**—Click this link to view the status of all devices provisioned with a service and, if needed, perform further actions, such as reprovisioning, deprovisioning, activating, or deactivating the service on specific Media Flow Controllers. For more information about managing the services that are provisioned, see [“Managing Provisioned Services” on page 144](#).
- **Tag Service**—Click this link to tag the services.
- **Untag Service**—Click this link to untag the selected tags from the specific service.
- **View Tags**—Click this link to view the tags associated with a specific service.
- **Purge Content**—Click this link to purge all or specific content, which you no longer need to cache, from selected Media Flow Controller devices. For more information about purging content, see [“Purging Content from Media Flow Controller Devices” on page 133](#).
- **Show Service Details**—Click this link to view the configuration of the specific service.

**Related
Documentation**

- [Network Optimization Services Overview on page 97](#)
- [HTTP Reverse Proxy Services Overview on page 111](#)
- [Content Ingest Services Overview on page 135](#)
- [Service Design Overview on page 92](#)
- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

PART 5

Service Provisioning

- [Overview on page 141](#)

CHAPTER 13

Overview

- [Provisioning Services Overview on page 141](#)
- [Provisioning Services on page 142](#)
- [Managing Provisioned Services on page 144](#)

Provisioning Services Overview

This topic describes what needs to be in place before you can provision a service. With the **Service Provisioning** workspace, you can push your configured service design to selected Media Flow Controllers.

Before you can provision sites, you must create services to provision by using the **Service Design** workspace. See [“Creating Network Optimization Services” on page 98](#), [“Creating HTTP Reverse Proxy Services” on page 112](#), and [“Creating Content Ingest Services” on page 135](#) for more information about creating these services.

Provisioning services pushes all the configurations of a service to the selected Media Flow Controllers.



NOTE: See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about provisioning transparent or reverse proxy services (deployments).

Related Documentation

- [Provisioning Services on page 142](#)
- [Managing Provisioned Services on page 144](#)
- [Service Design Overview on page 92](#)
- [Media Flow Activate Overview on page 3](#)
- [Understanding Media Flow Controller Management with Media Flow Activate on page 5](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Provisioning Services

Use the **Service Provisioning** workspace to push your service design to selected Media Flow Controllers.

Before you begin, you must have discovered Media Flow Controller devices and created at least one service from the **Service Design** workspace. See “[Creating Network Optimization Services](#)” on page 98, “[Creating HTTP Reverse Proxy Services](#)” on page 112, and “[Creating Content Ingest Services](#)” on page 135 for more information about creating a Network Optimization, HTTP Reverse Proxy, or Content Ingest service, respectively.

To provision services to managed Media Flow Controllers:

1. From the left navigation panel, click the plus sign (+) adjacent to **Service Provisioning**.
2. Click one of the following links:
 - **Network Optimization**—To provision a transparent proxy service
 - **HTTP Reverse Proxy**—To provision a reverse proxy service
 - **Content Ingest**—To provision a content ingest service

When you provision a reverse proxy service, you can bind the service to a resource pool. The resources that are made available to the domains or websites configured in the service are then governed by the resource pool parameters. However, if you do not bind any resource pool, the service is bound to the global resource pool.

To bind a service to a resource pool, select the resource pool in the **Resource pool** column.

For a reverse proxy or content ingest service, you can also view the services that have been previously provisioned on the device by clicking **View** in the **Current Service List** column.

Figure 7: HTTP Reverse Proxy Provisioning - Select Devices

Service Provisioning > HTTP Reverse Proxy

Select Devices 0 Items Selected Select: Page | None Search: Name

Name	Running Image	IP Address	Connection St...	Managed Status	Resource pool	Current Servic...
<input type="checkbox"/> mfc-165	mfc-12.2.4-qa	10.157.34.1...	up	In Sync	Global Select	View
<input type="checkbox"/> mfc-163	mfc-12.2.4-qa	10.157.34.1...	up	In Sync	Global Select	View
<input type="checkbox"/> mfc-152	mfc-12.2.3-qa	10.157.43.1...	up	In Sync	Global Select	View
<input type="checkbox"/> mfc-164	mfc-12.2.4-qa	10.157.34.1...	up	In Sync	Global Select	View
<input type="checkbox"/> mfc-191	mfc-12.2.3-qa	10.157.43.1...	up	In Sync	Global Select	View

User resource pool appears here

3. Select the Media Flow Controller devices that you want to provision and click **Next**. The **Select Service Instances** page is displayed.

4. Select the **Service Instances** that you designed and click **Next**. The selected services are provisioned to the selected Media Flow Controllers.
5. Click **Finish**. A pop-up is displayed with a status message and the Job ID: “**Please click on the Job ID link for details.**”

To view the status of provisioned services:

1. From the left navigation panel, click the plus sign (+) adjacent to **Service Design**. Click **Network Opt Services** to display the configured transparent proxy services. Click **HTTP Reverse Proxy Services** to display the configured reverse proxy services. Click **Content Ingest Service** to display the configured content ingest services.
2. Select the service for which you want to view the status.
3. On the **Actions** list, select **Manage Provisioned Devices**. Verify the progress of the services that you have provisioned.

You can also track the provisioning of the services with the **Job Management** workspace. To view the status of provisioned services:

1. From the left navigation panel, click the plus sign (+) adjacent to **Job Management**. The **Job Management** inventory landing page is displayed. You can see graphs for current **Job Types**, **State of Jobs Run**, and **Average Execution Time per Completed Job**.
2. Click **Manage Jobs**. The **Manage Jobs** page is displayed. You see the job ID, name, percentage of job completed, state of the job, job type (Provisioning, Restart Service, Software Upgrade, or Restart Devices), summary of the job, scheduled start time, user details, recurrence details (if applicable), and the retry group Id.



NOTE: In the Job Management workspace, a Media Flow Activate Provisioning Job shows a list of services that are provisioned, list of devices on which these services are provisioned, and whether the services were successfully initiated or not. If the services were successfully initiated, the Job Management workspace shows the corresponding job ID. The Restart Service job shows information about the set of devices, that this operation was initiated, and whether the operation is successfully initiated or not.

See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about provisioning transparent or reverse proxy services (deployments).

Related Documentation

- [Managing Provisioned Services on page 144](#)
- [Provisioning Services Overview on page 141](#)
- [Job Management Workspace Overview on page 175](#)
- [Resource Pools Overview on page 33](#)
- [Provisioning Resource Pools to MFC Devices on page 36](#)
- [Media Flow Activate Overview on page 3](#)

- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Managing Provisioned Services

Purpose View the status of all devices provisioned with a service on the basis of the status of the provisioned service.

Action To view the status of a provisioned service:

1. Select a service from the **Service Design** workspace (that is, from the **Network Optimization Services** page, **HTTP Reverse Proxy Services** page, or the **Content Ingest Services** page).
2. On the **Actions** list, select **Manage Provisioned Devices**. The **Manage Provisioned Devices** page is displayed, showing each Media Flow Controller provisioned with the service you selected in Step 1.

Meaning The **Manage Provisioned Devices** page displays the following status information for the selected service:

- **Name**—Media Flow Controllers provisioned with the service
- **IP Address**—IP addresses of the Media Flow Controllers provisioned with the service
- **Resource pool**—Resource pools associated with the Media Flow Controllers for the specific service (reverse proxy)



NOTE: If the service is not associated with a user-defined resource pool, the service is associated with the “Global” resource pool, by default. For more information about resource pools, see [“Resource Pools Overview” on page 33](#).

- **Service Status**—Whether the service is active or inactive
- **Managed Status**—Whether the device inventory information in the Junos Space database matches the current configuration information on the Media Flow Controller device. The “Managed Status” of the device can be one of the following:
 - **In Sync**—Indicates that the Media Flow Controller device configuration and the configuration information in the Junos Space database are in sync. It is recommended that you provision the services, only when the configurations are in sync.
 - **Out of Sync**—Indicates that the Media Flow Controller device configuration and the configuration information in the Junos Space database are out of sync. This usually happens when configurations are made to the device but are not yet committed. Wait till the configurations are in sync before you provision a service to a device.
 - **Synchronizing**—Indicates that the resync job has begun. The “Managed Status” of the device changes to “In Sync” after the resync job has completed.
 - **Sync Failed**—Indicates that resynchronization has failed.

To resolve this issue, try resynchronizing the managed device by following the steps mentioned in the "Resynchronizing Managed Devices" section of the *Junos Space Network Application Platform User Guide*. If resynchronization does not rectify the issue, you must delete the device and rediscover it.

- **Device Status**—Whether the device is Up (discovered and functioning); or Down (not functioning)
- **Provisioning Status**—Whether the provisioning job is Successful (completed) or Failed (not completed) and the Provision Job identification number (Click the Provision job ID on the Job Management > Manage Jobs page for more details about that provisioning job.)

In the **Manage Provisioned Devices** page, you can sort the data and even choose what columns you want to display by:

- Mousing over a column and clicking the list.
- Selecting **Sort Ascending** or **Sort Descending** to sort the data in ascending or descending order.
- Selecting **Columns** and choosing the columns to display. By default, the following columns are not displayed on the **Manage Provisioned Devices** page: **Id**, **Configuration**, and **Ref**. Select these columns, if you want the information for these columns to be displayed, as well.

Use the **Search** option to display specific Media Flow Controllers by filtering using their names or tags.

From this page, you can select Media Flow Controllers and then select one of the following options:

- **Provision Again**—Provision the service again. Select this option, if the **Provisioning Status** is Failed, or if you have modified the service.
- **De-provision**—Remove the association of the device with this service. A confirmation dialog box is displayed; click **Ok** to complete the deletion. First, the service configurations are deleted from the selected Media Flow Controllers; then the service association with the selected Media Flow Controllers is deleted from Media Flow Activate.

You may consider deprovisioning a service for the following reasons:

- When you no longer want to cache the content for a website and you want to remove the service completely from Media Flow Controller.
- When any of the Media Flow Controllers is in inconsistent state due to some error.
- **Activate**—Activate the service. A newly created service is inactive by default and you must explicitly activate it.

When a service is provisioned for the first time, the service is in the active state.

- **De-activate**—Deactivate the service. Media Flow Controller drains the connections when a service is deactivated. No new connections are accepted and no new requests

are accepted in the current connections. Any existing traffic is brought to a graceful shutdown.

You can deactivate a service when you want to make numerous changes to the service configuration—for example, when you want to update the website and you do not want visitors to the website during the update.



CAUTION: Deactivation of a service disrupts the service.

- **Cancel**—Exit the **Manage Provisioned Devices** page; no changes are made.

See the *Juniper Networks Media Flow Controller Administrators Guide* for detailed information about provisioning the transparent or reverse proxy services (deployments).

**Related
Documentation**

- [Service Design Overview on page 92](#)
- [Creating Network Optimization Services on page 98](#)
- [Creating HTTP Reverse Proxy Services on page 112](#)
- [Creating Content Ingest Services on page 135](#)
- [Provisioning Services Overview on page 141](#)
- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

PART 6

Configuration Templates

- [Overview on page 149](#)

Overview

- [Configuration Templates Overview on page 149](#)

Configuration Templates Overview

Using the Device Templates feature in Junos Space Network Application Platform, you can create Media Flow Controller–specific device templates to provision platform attributes in multiple Media Flow Controller devices. [Table 7 on page 152](#) lists the Media Flow Controller–specific attributes or CLI commands that are supported through this feature.

You use this feature when you want to configure those aspects of infrastructure that need to be provisioned before the Content Delivery Infrastructure (CDI) can provide content delivery services. This is because it is necessary that the server (physical or virtual) and the delivery application (Media Flow Controller) are configured correctly before the services are provisioned. Some of the configuration that you can set up before provisioning the services (namespaces) and their related attributes are: configuring the management IP address, and the TACACS and SNMP server addresses, bonding the Ethernet interfaces, and so on.

As a network operator, use device templates to:

- Develop and maintain a set of basic platform provisioning CLI commands as a template and provision the template to all new devices at the receiving (preparation) center before sending the devices for rack installation at the deployment location.
- Develop and maintain templates for different geographic locations or regions, server types (such as VXA or MX Series service card), or deployment types (such as an edge or midtier parent).

When you have device-specific values, you might want to consider a comma-separated value (CSV) file for a template definition. After you have created a CSV file, you must import it into Junos Space. See “Specifying Device-Specific Values in Definitions” in the *Junos Space Network Application Platform User Guide* for more information about creating a device-specific template definition.

Such one-time configuration commands are best suited to be part of device templates. You can also create a golden configuration for all devices in the network on the basis of locations, cache tier, and so on, which you can use to readily bring up a new device.



NOTE:

- The commands that are available through the Device Templates feature and the commands that are available through the MFA GUI are mutually exclusive. This ensures that there are no mismatches between the device template configuration and the configuration done through the MFA GUI.
 - Template definition and deployment is Media Flow Controller version specific. That is, there are different template definition schemas published for different Media Flow Controller versions.
-

“Device Templates” in the *Junos Space Network Application Platform User Guide* contains detailed instructions on creating and deploying template definitions and templates. Briefly, the process to deploy a template to Media Flow Controller devices is as follows:

Before you begin:

- Make sure you have the appropriate permissions. You need the **Template Design Manager** role to create and publish template definitions. Typically, the user with the **Template Manager** role selects a template definition and creates a template from it to configure one or more devices.
- You must not use your browser’s Back and Forward buttons to navigate the **Device Templates** pages.
- You must configure all the supported Media Flow Controller properties in one single template.
- You must upload the MFC 12.2.4 configuration schema file (config.xsd file) to Junos Space Network Application platform and set this schema as the default schema. You must contact the Juniper Networks technical support team to obtain this file.

Perform the following steps to upload this configuration schema file to Junos Space Network Application platform:

- a. Download this configuration schema file to your system.
- b. From the Junos Space Network Platform GUI, select **Platform > Administration > Manage DMI Schemas > Update Schema**. The **Schema Update** page displays.
- c. From the **Schema Update** page, click the **Browse** button adjacent to the **Archived Schemas File** field and select the configuration file from your system.
- d. Click **Upload**. This file appears under the **Available Updates (already installed versions are pre-selected)** area.
- e. Select the **Enable Schema Overwrite** check box.
- f. From the **Available Updates (already installed versions are pre-selected)** area, select this configuration file and click **Install**.
- g. After the successful installation of the schema, click **Manage DMI Schemas**. The **Manage DMI Schemas** page is displayed.

To verify whether the schema has been successfully installed, see the corresponding job in the Job Management workspace.

- h. From the **Manage DMI Schemas** page, select this configuration file.
- i. From the **Actions** list, click **Set Default Schema** to set this schema as the default schema.
1. On the **Create Definition** page, create a template definition.

To complete this task, follow the instructions mentioned in “Creating a Template Definition” in the *Junos Space Network Application Platform User Guide*.

When you create a template definition, make sure that you:

- a. Select the **Media Flow** device family.
- b. Select the appropriate **OS Version**. The Device Templates feature is supported for Media Flow Controller devices running version 12.2.4 or later.
- c. Use the search function to quickly locate a specific configuration option. For more information about locating a specific configuration option, see “Finding Configuration Options” in the *Junos Space Network Application Platform User Guide*.
- d. Set device-specific values, if needed. For more information about setting device-specific values, see “Specifying Device-Specific Values in Definitions” in the *Junos Space Network Application Platform User Guide*.

The template definition file that you create is an XML file. The template definition file contains its own schema and is different from the device configuration schema. A template definition can be exported, modified, and imported to the same or a new Junos Space server. For more information about template definitions, see “Template Definitions” in the *Junos Space Network Application Platform User Guide*.

2. Publish the template definition. For more information about publishing a template definition, see “Publishing and Unpublishing a Template Definition” in the *Junos Space Network Application Platform User Guide*.
3. Define a configuration template or a variable-based configuration template with a fixed set of configuration commands from the template definition.

For example, to map a license to a device, you need to provision a variable-based configuration template. Junos Space configuration templates provide you with an option of marking this configuration value as “device-specific.” Typically, template designers use a comma-separated value (CSV) file to provide device-specific values in a template definition.

You can also use rules to supplement the device-specific value capability supplied by CSV files. Specify rules to resolve device-specific values at the time of deployment. You can use rules in addition to or instead of CSV files.

To create a template, follow the instructions mentioned in “Creating a Template” in the *Junos Space Network Application Platform User Guide*. Ensure that you select a **MEDIA-FLOW** template.

**TIP:**

Before you deploy the device template:

- Select the template that is most appropriate for your requirement from the **Manage Templates** page.
- User must create a single template for each “mfc-cluster” object as the template deployment is not additive and if you create and deploy a new template for each feature under the same “mfc-cluster” object to a device, the previous template deployed for that ‘mfc-cluster’ object is undeployed, even if the subsequent template contains only additional parameter settings.

4. Deploy the template to one or more Media Flow Controllers either on demand or at a scheduled time in the future. You can select the Media Flow Controllers one by one or filter them using their tags. Each publishing is handled as a Junos Space job. When a configuration template is published to multiple Media Flow Controllers, the publishing status for each Media Flow Controller is displayed in a separate row in the Job Management workspace.

This deployment action also allows you to validate the template against the device family and against the device. For more information about deploying a template, see “Deploying a Template” in the *Junos Space Network Application Platform User Guide*.

After you deploy the template, if you want to:

- View the list of devices to which a template has been deployed, select **View Template Deployment** from the **Actions** list.
- Verify the extent to which a template and the device to which the template has been deployed match, select **Audit Log Config** from the **Actions** list.

For troubleshooting, see “Troubleshooting” under “Device Templates” in the *Junos Space Network Application Platform User Guide*.

[Table 7 on page 152](#) lists the Media Flow Controller CLI commands that are supported through the Device Templates feature:

Table 7: MFC CLI Commands Supported Through Junos Space Configuration Templates

CLI Commands	Description
license install	Activate features with license keys.
Hostname	Set the system's hostname.
ip name-server	Add a name server.
ip default-gateway	Set the default gateway.
ip route	Add a static route.

Table 7: MFC CLI Commands Supported Through Junos Space Configuration Templates (*continued*)

CLI Commands	Description
ip domain-list	Add a domain name to use when resolving hostnames.
ntp server	Set the NTP server. This command can be deleted or disabled.
clock timezone	Set the timezone. This command can be deleted or disabled.
ssh client user <user> authorized-key sshv2 <key>	Set SSH client authentication.
ssh server enable listen	Set SSH server configuration.
snmp-server	Set up the SNMP server, such as IPv4 SNMP Host, community string, listen interface, syscontact, and syslocation. These parameters are configurable through device templates. Trap port, trap version, and enabling all traps cannot be configured through device templates and are set to their default values. This command can be deleted or disabled.
tacacs-server host	Allow configuration of up to three TACACS server hosts and their attributes—that is, shared secret and timeout for every individual host. The remaining attributes are set to their default values. This command can be deleted or disabled.
logging	Set the system log configuration to specify local and remote logging and severity levels. This command can be deleted or disabled.
ram-cache cache-size-MB dict-size-MB small-buffers scale-factor <i>number</i> small-attribute size <i>number</i> count <i>number</i>	Configure RAM cache options. The small-attribute command can be deleted or disabled.
telnet-server enable	Enable telnet.
service snapshot mod-delivery enable service status mod-delivery include disk	Tune mod-delivery service–related parameters. Enable snapshot (core file) to be generated when the mod-delivery service crashes. Enable considering pre-read status before declaring the system status UP.

Table 7: MFC CLI Commands Supported Through Junos Space Configuration Templates (*continued*)

CLI Commands	Description
<code>network connection concurrent session</code>	Configure network layer parameters.
<code>network connection origin failover</code> <code>use-dns-response</code>	The <code>network connection concurrent session</code> command can be deleted or disabled.
<code>name-resolver cache-timeout (auto random seconds)</code>	This command is not needed if the Media Flow Controller CLI command is fixed to set "auto" as the default value.

- Related Documentation**
- [Media Flow Activate Overview on page 3](#)
 - [Quick Reference to Tasks in Media Flow Activate on page 191](#)

PART 7

Network Monitoring

- [Fault Monitoring on page 157](#)

Fault Monitoring

- [Fault Monitoring with SNMP on page 157](#)

Fault Monitoring with SNMP

To provide network monitoring capabilities, Junos Space Network Application Platform is integrated with a third-party tool called OpenNMS. The OpenNMS network management application platform provides solutions for enterprises and carriers. Junos Space Network Application Platform on which OpenNMS is installed exposes some functionality of OpenNMS through the Network Monitoring workspace. The default performance management configuration of OpenNMS for Media Flow Activate enables you to quickly view basic device statistics, such as device availability and interface availability for the entire Media Flow Network through the OpenNMS dashboard landing page. As a Media Flow Network administrator or Data Center administrator, you can monitor the performance statistics of individual Media Flow Controller devices as well as networkwide aggregated statistics using the OpenNMS dashboard. You can also configure OpenNMS to display critical events, such as delivery outages, on the same dashboard.

For more information about OpenNMS configuration-related information, see the “Network Monitoring” section in the *Junos Space Network Application Platform User Guide*.



CAUTION: Although you can access additional OpenNMS functionality by customizing its XML files, editing these files can affect the functionality of the Network Monitoring workspace. Juniper Networks does not support changes to OpenNMS.

In order to facilitate fault monitoring from the **Network Monitoring** workspace:

- When Media Flow Controllers are discovered from the Junos Space GUI, they are automatically added to the list of monitored devices or nodes on OpenNMS.
- All the basic monitoring capabilities, such as the ICMP ping for device outage, interface availability, and service outages, are supported for Media Flow Controllers.
- Media Flow Controllers are configured to send traps or notifications to OpenNMS when significant events occur on the Media Flow Controller devices. The list of Media Flow Controller events that can be tracked from the OpenNMS dashboard are listed in [Table 8 on page 159](#) and [Table 9 on page 161](#).

Before you start monitoring the Media Flow Controller, make sure that the SNMP server host IP address of the Media Flow Controller is set to the Junos Space server IP address. You can use the Junos Space device template feature to provision this configuration to multiple Media Flow Controllers (**Device Templates > Create Definition > Media Flow > Configuration > Services > Mfc cluster > System > Monitoring > Snmp server > Host > Ip address**).

For more information about all the operations that you can perform from the **Network Monitoring** workspace, see the “Network Monitoring UI” section in the *Junos Space Network Application Platform User Guide*. The following list of actions should help you get started:

- Select **Node List** and click one of the displayed nodes to view:
 - General status of the node
 - Recent events that occurred in the node
 - Recent outages that occurred in the node
 - Notifications
 - SNMP attributes
 - Availability of the interfaces



TIP: If you have recently modified the host name of a Media Flow Controller, it is possible that this node may not be displayed in the nodes list. Resynchronize the nodes to update this list. Select **Network Monitoring > Node List > Resync Nodes > Confirm** to complete this task.

- Select **Search** to search for a specific node.
- Select **Dashboard** to open the OpenNMS dashboard.
- Select **Events** to view the events that occurred within the network. From the **Events** page that is displayed, select **View all events** to view all the events. In the tabular view, click the **Severity**, **Node**, or **Time** column heading to sort the data in ascending or descending order. For more information about these events, see the “Events” section in the *Junos Space Network Application Platform User Guide*.



NOTE: Every change in the network can be considered an event. An event is raised in OpenNMS as a result of receiving an SNMP Trap from a Media Flow Controller device. If you do not receive an event that occurred in any of the Media Flow Controller device that you are monitoring, make sure that the corresponding SNMP trap is enabled through the Media Flow Controller CLI command (mentioned at the end of this topic).

- Select **Alarms** to view significant events that occurred on the network. From the **Alarms** page that is displayed, select **All alarms (summary)** or **All alarms (detail)** to view the shorter or detailed version of all the alarms. For more information about alarms, see the “Alarms” section in the *Junos Space Network Application Platform User Guide*.

From Media Flow Activate, you can view the alarms that are specific to each Media Flow Controller from the **Media Flow Devices** inventory landing page:

1. Click **Media Flow Devices** to view the list of Media Flow Controllers.
2. Select the Media Flow Controller whose alarms you want to view.
3. From the **Actions** list, select **View Service Alarms**. The alarms that are specific to the device are displayed.

[Table 8 on page 159](#) lists the SNMP alarm events of severity levels—critical, major, and minor (in that order), which are displayed on the OpenNMS dashboard. To view the corresponding clear alarm events (alarms that indicate that the system has recovered from events of higher severity levels), see [Table 9 on page 161](#).

Table 8: SNMP Alarm Events

SNMP Event or Trap (Alarm Event)	Severity Level	Description	Log Message
jmfcFanFailure	Critical	The system fan has stopped functioning.	jmfcFanFailure trap received
jmfcPowerSupplyFailure	Critical	The system power supply has failed.	jmfcPowerSupplyFailure trap received
jmfcSmartError	Critical	SMART has sent an event about a possible disk error.	jmfcSmartError trap received
jmfcUnexpectedShutdown	Critical	The system has shut down unexpectedly.	jmfcUnexpectedShutdown trap received
jmfcServiceCrash	Major	One of the monitored services is down due to a crash.	jmfcServiceCrash trap received jmfcServiceName= <i>name</i>
jmfcCpuUtilHigh	Major	The aggregate CPU utilization across all CPUs is high.	jmfcCpuUtilHigh trap received jmfcCpuUtil=%parm[#1]% jmfcCpuUtilErrThreshold=%parm[#2]%;
jmfcDiskSpaceLow	Major	Free disk space is low.	jmfcDiskSpaceLow trap received jmfcDiskName=%parm[#1]% jmfcDiskFreeSpace=%parm[#2]% jmfcDiskSpaceErrThreshold=%parm[#3]
jmfcOriginNodeDown	Major	One of the nodes in the cluster is down.	jmfcOriginNodeDown trap received
jmfcApplCpuUtilHigh	Major	CPU utilization of core HTTP Engine is high.	jmfcApplCpuUtilHigh trap received jmfcAppMaxCpuUtil=%parm[#1]% jmfcAppMaxCpuUtilErrThreshold=%parm[#2]%

Table 8: SNMP Alarm Events (*continued*)

SNMP Event or Trap (Alarm Event)	Severity Level	Description	Log Message
jmfcServiceExit	Major	One of the services managed by Process Manager has exited unexpectedly, but has not left a core file.	jmfcServiceExit trap received jmfcServiceName=%parm[#1]%
jmfcServiceLivenessFailure	Major	Process Manager has detected that a process has hung and is set to restart.	jmfcServiceLivenessFailure trap received jmfcServiceName=%parm[#1]%
jmfcCacheHitRatioLow	Minor	Cache hit ratio is low.	jmfcCacheHitRatioLow trap received jmfcCacheHitRatio=%parm[#1]% jmfcCacheHitRatioErrThreshold=%parm[#2]%
jmfcMemUtilizationHigh	Minor	Memory utilization on the system is high.	jmfcMemUtilizationHigh trap received jmfcMemUtil=%parm[#1]% jmfcMemUtilErrThreshold=%parm[#2]
jmfcNetUtilizationHigh	Minor	Network utilization on the system is high.	jmfcNetUtilizationHigh trap received jmfcNetIntfUtilCurrent=%parm[#1]% jmfcNetIntfUtilErrThreshold=%parm[#2]%
jmfcDiskIOHigh	Minor	Disk I/O on the system is high.	jmfcDiskIOHigh trap received jmfcDiskName=%parm[#1]% jmfcDiskIORate=%parm[#2]% jmfcDiskIORateErrThreshold=%parm[#3]%
jmfcCacheBandwidthUsageHigh	Minor	The cache bandwidth usage is high.	jmfcCacheBandwidthUsageHigh trap received jmfcCacheBw=%parm[#1]% jmfcCacheBwErrThreshold=%parm[#2]%s
jmfcOriginBandwidthUsageHigh	Minor	The origin bandwidth usage is high.	jmfcOriginBandwidthUsageHigh trap received jmfcOriginBw=%parm[#1]% jmfcOriginBwErrThreshold=%parm[#2]%
jmfcDiskBandwidthUsageHigh	Minor	The disk bandwidth usage is high.	jmfcDiskBandwidthUsageHigh trap received. jmfcDiskName=%parm[#1]% jmfcDiskBW=%parm[#2]% jmfcDiskBWErrThreshold=%parm[#3]%
jmfcConnectionRateHigh	Minor	The connection rate is high.	jmfcConnectionRateHigh trap received jmfcConnectionRateCurrent=%parm[#1]% jmfcConnectionRateErrThreshold=%parm[#2]%
jmfcTransactionRateHigh	Minor	The HTTP transaction rate is high.	jmfcTransactionRateHigh trap received jmfcTransactionRateCurrent=%parm[#1]% jmfcTransactionRateErrThreshold=%parm[#2]%

Table 8: SNMP Alarm Events (*continued*)

SNMP Event or Trap (Alarm Event)	Severity Level	Description	Log Message
jmfcPagingHigh	Minor	The paging activity is high.	jmfcPagingHigh trap received jmfcPagingCurrent=%parm[#1]% jmfcPagingErrThreshold=%parm[#2]%
jmfcResourcePoolUsageHigh	Minor	The usage of a resource pool has exceeded its defined upper limit.	jmfcResourcePoolUsageHigh trap received jmfcResourcePoolName=%parm[#1]% jmfcResourcePoolBandwidth=%parm[#2]% jmfcResourcePoolActiveConns=%parm[#3]%
jmfcResourcePoolUsageLow	Minor	The usage of a resource pool has fallen lower than its defined lower limit.	jmfcResourcePoolUsageLow trap received jmfcResourcePoolName=%parm[#1]% jmfcResourcePoolBandwidth=%parm[#2]% jmfcResourcePoolActiveConns=%parm[#3]%

Table 9 on page 161 lists the alarm events that indicate that the system has recovered from events of higher severity levels.

Table 9: SNMP Clear Alarm Events

SNMP Event or Trap (Clear Alarm Event)	Severity Level	Description	Log Message
jmfcServiceUp	Cleared	One of the monitored services is restarted.	jmfcServiceUp trap received, Clearing Service Crash Alarm. jmfcServiceName= <i>name</i>
jmfcFanStatusOK	Cleared	The fan status is okay.	jmfcFanStatusOK trap received, Clearing Fan Failure Alarm.
jmfcPowerSupplyOk	Cleared	The system power supply is restored.	jmfcPowerSupplyOk trap received, Clearing Power Supply Alarm.
jmfcCacheHitRatioOk	Cleared	Cache hit ratio is normal.	jmfcCacheHitRatioOk trap received, Clearing Cache Hit Ratio Low Alarm. jmfcCacheHitRatio=%parm[#1]% jmfcCacheHitRatioClrThreshold=%parm[#2]%
jmfcCpuUtilOk	Cleared	The aggregate CPU utilization across all CPUs has fallen back to normal.	jmfcCpuUtilOk trap received, Clearing CPU Utilization High Alarm. jmfcCpuUtil=%parm[#1]% jmfcCpuUtilClrThreshold=%parm[#2]%
jmfcDiskSpaceOk	Cleared	Free disk space is normal.	jmfcDiskSpaceOk trap received, Clearing Disk Space Alarm. jmfcDiskName=%parm[#1]% jmfcDiskFreeSpace=%parm[#2]% jmfcDiskSpaceClrThreshold=%parm[#3]%
jmfcMemUtilizationOk	Cleared	Memory utilization on the system has come down to the normal level.	jmfcMemUtilizationOk trap received, Clearing High Memory Utilization Alarm. jmfcMemUtil=%parm[#1]% jmfcMemUtilClrThreshold=%parm[#2]%

Table 9: SNMP Clear Alarm Events (*continued*)

SNMP Event or Trap (Clear Alarm Event)	Severity Level	Description	Log Message
jmfcNetUtilizationOk	Cleared	Network utilization on the system has come down to the normal level.	jmfcNetUtilizationOk trap received, Clearing Net Utilization High Alarm. jmfcNetIntfUtilCurrent=%parm[#1]% jmfcNetIntfUtilClrThreshold=%parm[#2]%
jmfcDiskIOOk	Cleared	Disk I/O on the system has come down to the normal level.	jmfcDiskIOOk trap received, Clearing Disk IO High Alarm. jmfcDiskName=%parm[#1]% jmfcDiskIORate=%parm[#2]% jmfcDiskIORateClrThreshold=%parm[#3]%
jmfcOriginNodeUp	Cleared	One of the failed nodes in the cluster is up, clearing Origin Node Down Alarm.	jmfcOriginNodeUp trap received
jmfcApplCpuUtilOk	Cleared	CPU utilization of core HTTP Engine has fallen back to normal.	jmfcApplCpuUtilOk trap received, Clearing App. Utilization High Alarm. jmfcAppMaxCpuUtil=%parm[#1]% jmfcAppMaxCpuUtilClrThreshold=%parm[#2]%
jmfcCacheBandwidthUsageOk	Cleared	The cache bandwidth usage has come down within normal limits.	jmfcCacheBandwidthUsageOk trap received, Clearing Cache Bandwidth Usage High Alarm. jmfcCacheBw=%parm[#1]% jmfcCacheBwClrThreshold=%parm[#2]%
jmfcOriginBandwidthUsageOk	Cleared	The origin bandwidth usage has come down within normal limits.	jmfcOriginBandwidthUsageOk trap received, Clearing Origin Bandwidth Usage High Alarm. jmfcOriginBw=%parm[#1]% jmfcOriginBwClrThreshold=%parm[#2]%
jmfcDiskBandwidthUsageOk	Cleared	The disk bandwidth usage has come down within normal limits.	jmfcDiskBandwidthUsageOk trap received, Clearing Disk Bandwidth Usage High Alarm. jmfcDiskName=%parm[#1]% jmfcDiskBW=%parm[#2]% jmfcDiskBWClrThreshold=%parm[#3]%
jmfcConnectionRateOk	Cleared	The connection rate has come down within normal limits.	jmfcConnectionRateOk trap received, Clearing Connection Rate High Alarm. jmfcConnectionRateCurrent=%parm[#1]% jmfcConnectionRateClrThreshold=%parm[#2]%
jmfcTransactionRateOk	Cleared	The HTTP transaction rate has come down within normal limits.	jmfcTransactionRateOk trap received, Clearing Transaction Rate High Alarm. jmfcTransactionRateCurrent=%parm[#1]% jmfcTransactionRateClrThreshold=%parm[#2]%
jmfcPagingOk	Cleared	The paging activity has come down to normal level.	jmfcPagingOk trap received, Clearing Paging High Alarm. jmfcPagingCurrent=%parm[#1]% jmfcPagingClrThreshold=%parm[#2]%

Table 9: SNMP Clear Alarm Events (*continued*)

SNMP Event or Trap (Clear Alarm Event)	Severity Level	Description	Log Message
<code>jmfcResourcePoolHighUsageOK</code>	Cleared	The usage of a resource pool has fallen back to its normal limit.	<code>jmfcResourcePoolHighUsageOK trap received, Clearing Resource Pool usage High Alarm. jmfcResourcePoolName=%parm[#1]% jmfcResourcePoolBandwidth=%parm[#2]% jmfcResourcePoolActiveConns=%parm[#3]%</code>
<code>jmfcResourcePoolLowUsageOK</code>	Cleared	The usage of a resource pool has come up to its normal limit.	<code>jmfcResourcePoolLowUsageOK trap received, Clearing Resource Pool Usage Low Alarm. jmfcResourcePoolName=%parm[#1]% jmfcResourcePoolBandwidth=%parm[#2]% jmfcResourcePoolActiveConns=%parm[#3]%</code>

You can execute the following commands to view and enable events and alarms in a Media Flow Controller device:

- **show snmp events *cr***—View the list of SNMP events that are enabled.
- **snmp-server traps event *event-name cr***—Enable a specific SNMP event.
- **show stats alarm *cr***—View the list of alarms that are enabled.
- **stats alarm *alarm-ID* [enable|clear|event-repeat|rate-limit|falling|rising]**—Enable a specific alarm.

Related Documentation

- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

PART 8

Audit Logs

- [Overview on page 167](#)

CHAPTER 16

Overview

- [Audit Logs Workspace Overview on page 167](#)

Audit Logs Workspace Overview

From the Audit Logs workspace, with the Audit Log Administrator role, you can monitor tasks initiated by users from the Junos Space GUI. Any user-initiated task that is performed from the Media Flow Activate GUI is recorded in the Audit Log database with information about the user who initiated the task, the time of the request, the device that was used, a list of modifications or changes, and so on. The Audit Logs workspace displays this information in two views: graphical and tabular. In the graphical view, you can view data on a daily, weekly, or monthly basis. The tabular view displays audit log entries.

The following actions generate audit logs in Junos Space:

- User logins and logouts
- User timeouts
- Authentication failures
- Each operation attempted by a logged-in MFA GUI user

Non-user-initiated activities, such as device-driven activities, are not logged in the Audit Log database.

From this workspace, you can perform the following tasks:

- **Viewing audit log statistics**—You can use the following graphs to monitor user activity.
 - The **Audit Log Statistical Graph** pie chart displays all tasks that have been performed and logged in all Junos Space applications over a specific period of time. You can view Audit Log statistics by task type, user, workspace, and application.

You can control how data is displayed on the pie chart by selecting the category and time scale. The category determines what statistical log graph is displayed, whereas the time scale bar allows you to select the time period for which the log data is displayed. You can click a sector within the pie to drill down to audit log details. For more information about the audit log statistical graph, see the "Viewing the Dynamic Audit Log Statistical Graph" section in the *Junos Space Network Application Platform User Guide*. Following is a brief overview of the data that is displayed when you choose a category:

- **Task**—Shows all tasks within the selected time frame. You can click a task within the pie to view the users who performed this task or the IP addresses from which this task was performed. Click **Overview** to go back to the first-level chart—that is, to view all the tasks within the selected time frame.
- **User**—Shows all users using the system within the selected time frame. Click a user to view the tasks performed by that user. Click **Overview** to go back to the first-level chart.
- **Workspace**—Shows all workspaces used in the selected time frame
- **Application**—Shows all applications for which audit logs were logged within the selected time frame. Click **REST** to view the audit logs generated for Media Flow Activate—this is because, internally, all user-initiated tasks from the Media Flow Activate GUI invokes the corresponding REST APIs to complete the tasks. In the tabular display of log entries, the **Description** column provides details about the API that was invoked. For example, this column displays something like **POST: /api/juniper/mfa/definitions-management/rproxy-definitions/**.
- The **Top 10 Active Users in 24 hours** graph displays the top 10 Junos Space users who performed the most tasks over 24 hours. For more information about this graph, see the “Viewing Audit Log Statistics” section in the *Junos Space Network Application Platform User Guide*.
- **Viewing audit logs**—This workspace provides a tabular view of audit log entries, displaying the following information:
 - **User Name**—Login ID of the user
 - **User IP**—IP address of the machine from which the user logged in
 - **Task**—Name of the performed task
Examples: **Login**, **Logout**, **HTTP POST**, and so on
 - **Timestamp**—Time when the task is executed or when the job is scheduled
Example: **Oct 20, 2012 11:12:03 AM UTC+05:30**
 - **Result**—Result of the executed task
Examples: **Success**, **200**, **202**, and so on
 - **Description**—Simple description of the audit log
Examples:
 - **Login Succeeded**
 - **POST: /api/space/device-management/devices/262148/exec-rpc**

- **Change request created successfully with Id : 295053**
- **Job: POST:**
/api/juniper/mfa/device-management/mfdevices/rproxy/bulk-provision/—Such log entries typically include a job ID, which you click to view details about the job.
- **Job ID**—ID of the job being scheduled (If the task performed is a scheduled job, you can view the job details by clicking the Job ID link in the Audit Log table.)

On the **View Audit Logs** page, you can perform the following actions:

- Click the column headings to sort the data in ascending or descending order.
- Enter a search criterion in the textbox adjacent to the **Search** icon to filter the logs.
- Select an audit log entry and click the **Display Quick View** icon next to the **Actions** list to view the summary of the audit log entry, which includes the list of affected objects. If no objects are affected, then the summary view displays **None**.

For more information, see the “Viewing Audit Logs” section in the *Junos Space Network Application Platform User Guide*.

- **Archiving and purging jobs**—With the **Archive/Purge** feature, you can manage your Junos Space log volume, which enables you to archive log files and then purge those log files from the Junos Space database. For each **Archive/Purge** operation, the archived log files are saved in a single file, in CSV format. The audit logs can be saved to a local server or a remote network host. When you archive data to a local server, the archived log files are saved to the default directory, `/var/lib/mysql/archive`. To specify the remote archive location, use the IP address of the remote machine. The default filename of the archived file is `JunosSpaceAuditLog_yyyy-mm-dd_hh-mm-ss.csv.gz`, where `yyyy-mm-dd_hh-mm-ss` is the date and time up to when all the audit logs recorded are archived and purged from the database.



NOTE: The date and time in the archive filename may differ from your local client's time zone. This is because the Audit Logs workspace displays the Junos Space server time zone, whereas your local client may be located in a different time zone.

The archived file includes information such as:

- Timestamp
- UTC Time
- User IP
- Application
- Task
- Result
- Description

- Job ID
- Username

To view the local time instead of the UTC time in the archived audit log file, follow the instructions in the “Converting the Audit Log File UTC Timestamp to Local Time in Microsoft Excel” section in the *Junos Space Network Application Platform User Guide*.

For more information about archiving and purging audit logs, see the “Archiving and Purging Audit Logs” section in the *Junos Space Network Application Platform User Guide*.

- **Exporting audit logs**—The audit logs export feature enables you to download entire or partial audit logs in CSV format so that you can view the audit logs in a separate application or save them on another machine for future use, without purging them from the system. For more information about exporting audit logs, see the “Exporting Audit Logs” section in the *Junos Space Network Application Platform User Guide*.

From the Media Flow Activate GUI, select one of the following options:

- **Export all audit logs**—To export all audit logs
- **Export audit logs filtered by date range**—To export audit logs that are logged within the specified time frame
- **Export audit logs as displayed on View Audit Logs table**—(Default) On the View Audit Logs page, you can filter audit logs on the basis of multiple criteria. The criteria you choose determine which audit log data is exported.

The audit logs are exported as CSV files (for example, **AuditLogs_2012-10-25_11-14-56.csv**). They are not removed from the database after they are exported.

Troubleshooting and Additional Information About Audit Logs

Some troubleshooting tips and other sundry information for audit logs are as follows:

- If an action is not completed due to validation errors, the action is not logged in the audit log.
- If an action fails due to database errors, the action is logged in the audit log with a result of “Failure”.
- If an audit log fails to be recorded due to exceptions, the corresponding error logs and debug logs are created in the **server.log** file. In this case, search for error logs and debug logs under “AuditlogRecorderBean” class.
- Audit log archive and purge actions may fail due to insufficient server disk space or due to errors in performing scp to remote server. The error details can be viewed from the Job Management workspace for the specific archive and purge jobs.
- In case of scp errors, the archived file is still created in the **/var/lib/mysql/archive** subdirectory in the Junos Space server.

Related Documentation

- [Media Flow Activate Overview on page 3](#)
- [Job Management Workspace Overview on page 175](#)

- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

PART 9

Job Management

- [Overview on page 175](#)

CHAPTER 17

Overview

- [Job Management Workspace Overview on page 175](#)

Job Management Workspace Overview

With the Job Management workspace, you can monitor the status of all jobs. A job is a user-initiated action that is performed on a Junos Space object. Each job is assigned a unique job ID that serves to identify the job. For more information about managing jobs from this workspace, see “Job Management” in the *Junos Space Network Application Platform User Guide*.

From Media Flow Activate, the following actions initiate jobs:

- From the **Media Flow Devices** inventory landing page:
 - Upgrading or rolling back a software
 - Replicating a device configuration
 - Restarting a service
 - Restarting devices
 - Restoring a device configuration
 - Provisioning, reprovisioning, or deprovisioning the resource pools
 - Configuring an interface for transparent or reverse proxy services
 - Configuring a bonded interface
- From the **Service Design** inventory landing page:
 - Provisioning, reprovisioning, or deprovisioning services
- From the **Service Provisioning** inventory landing page:
 - Provisioning services

When you initiate an action on multiple devices from Media Flow Activate, the main job that is created comprises several subjobs. Each subjob represents a user-initiated action for a single device. For example, when you provision a service on three devices, four jobs are created, of which three are subjobs representing the provisioning of the specific service on a single device. Starting from MFA Release 3.0, you can view the status of the subjobs

(whether they are successful or not) by clicking the main job. The pop-up that appears lists the subjobs with job IDs, the devices on which the action is performed, whether the job is a success or a failure, and any other special messages that provide additional information when the job is a failure. The status of the main job is shown as successful only when all its subjobs have been completed successfully.

**Related
Documentation**

- [Upgrading or Rolling Back the Media Flow Controller Software Image on page 21](#)
- [Restarting Media Flow Controller Devices or Services on page 29](#)
- [Provisioning Resource Pools to MFC Devices on page 36](#)
- [Managing Resource Pool Associations on page 37](#)
- [Provisioning Services on page 142](#)
- [Managing Provisioned Services on page 144](#)
- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

PART 10

Reference

- [Sample XML Schema on page 179](#)
- [Quick Reference on page 191](#)

Sample XML Schema

- [Sample XML Schema for Network Optimization and Reverse Proxy Services on page 179](#)

Sample XML Schema for Network Optimization and Reverse Proxy Services

This topic contains XML schema for Network Optimization and HTTP Reverse Proxy services, which you may use as a pointer to create your own XML schema. Using the Media Flow Activate GUI, you may import this XML schema to create services, which can then be provisioned to selected Media Flow Controllers. See the following sample XML files for:

- Network Optimization service
- HTTP Reverse Proxy service

Sample XML Schema for Network Optimization Service

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<contentDirectDefinitionCatalog>
  <virtualPlayerDefinitions>
    <genericPlayerDefinitions>
      <id>1</id>
      <name>vp1</name>
      <createdBy>super</createdBy>
      <lastModifiedBy>super</lastModifiedBy>
      <lastModifiedDateTime>2012-11-07T06:25:30Z</lastModifiedDateTime>
      <systemDefined>false</systemDefined>
      <type>generic</type>
      <seekConfiguration>
        <startIdentifier>start</startIdentifier>
        <endIdentifier>end</endIdentifier>
        <tunnelSeekRequest>true</tunnelSeekRequest>
      </seekConfiguration>
      <maxConnBandwidth>
        <noLimit>true</noLimit>
      </maxConnBandwidth>
      <fastStart>
        <defaultSizeinKB>100</defaultSizeinKB>
      </fastStart>
      <fullDownload>
        <always>true</always>
      </fullDownload>
```

```

    </genericPlayerDefinitions>
  </virtualPlayerDefinitions>
  <cacheTuningPolicies>
    <cacheTuningPolicy>
      <id>1</id>
      <name>ctp1</name>
      <description></description>
      <createdBy>super</createdBy>
      <lastModifiedBy>super</lastModifiedBy>
      <lastModifiedDateTime>2012-11-07T06:26:31Z</lastModifiedDateTime>
      <systemDefined>false</systemDefined>
      <ageSizeSettings>
        <ageThreshold>120</ageThreshold>
        <defaultCacheAge>120</defaultCacheAge>
        <minObjSize>4096</minObjSize>
        <diskIngestThreshold>0</diskIngestThreshold>
      </ageSizeSettings>
      <cacheExclusion>
        <queryString>true</queryString>
        <cacheRequestsWithCookies>
          <enable>true</enable>
          <option>VALIDATE_WITH_ORIGIN</option>
        </cacheRequestsWithCookies>
        <isDomainName>true</isDomainName>
        <queryStringInclusion>INCLUDE_QUERY_STRING</queryStringInclusion>
        <cacheRequestWithSpecificHeaders>
          <enable>true</enable>
          <headers>
            <header>AUTH_HEADER</header>
            <header>COOKIE_HEADER</header>
            <header>CACHECONTROL_HEADER</header>
          </headers>
        </cacheRequestWithSpecificHeaders>
      </cacheExclusion>
      <originFetch>
        <cacheAgeSettings>
          <cacheAgeList/>
          <contentAnyAgeInSecs>120</contentAnyAgeInSecs>
        </cacheAgeSettings>
        <noCacheDirective>FOLLOW</noCacheDirective>
        <overrideTunnelObjectExpired>true</overrideTunnelObjectExpired>
        <objectSizeMinimumThreshold>0</objectSizeMinimumThreshold>
        <objectSizeMaximumThreshold>0</objectSizeMaximumThreshold>
        <cacheFill>AGGRESSIVE</cacheFill>
        <ignoreNoTransformHeader>true</ignoreNoTransformHeader>
      </originFetch>
    </cacheTuningPolicy>
  </cacheTuningPolicies>
  <responsesWithHTTP302Status>PASS_THROUGH_TO</responsesWithHTTP302Status>

  <cacheIngestHotnessThreshold>3</cacheIngestHotnessThreshold>
  <uriDepthThreshold>10</uriDepthThreshold>
</originFetch>
<originRequest>
  <cacheRevalidation>
    <permit>true</permit>
    <useHEADMethod></useHEADMethod>
  </cacheRevalidation>

```

```

        <setXForwardedForHeader>true</setXForwardedForHeader>
    </originRequest>
    <clientRequest>
        <revalidationOverride>
            <overrideMaxAgeHeader>>false</overrideMaxAgeHeader>
        </revalidationOverride>
    </clientRequest>
    <diskCacheProperties>
        <SAS/>
        <SATA/>
        <SSD/>
    </diskCacheProperties>
</cacheTuningPolicy>
</cacheTuningPolicies>
<contentDirectDefinitions>
    <contentDirectDefinition>
        <id>1</id>
        <name>tp1</name>
        <description></description>
        <createdBy>super</createdBy>
        <lastModifiedBy>super</lastModifiedBy>
        <lastModifiedDateTime>2012-11-07T06:29:18Z</lastModifiedDateTime>
        <systemDefined>>false</systemDefined>
        <properties>
            <domain>google</domain>
            <isDomainRegex>>false</isDomainRegex>
            <orderOfServing>LIFO</orderOfServing>
        </properties>
    </contentDirectDefinition>
</contentDirectDefinitions>
<ignoreAllObjectsCorrelationValidators>true</ignoreAllObjectsCorrelationValidators>

    <match>
        <path>
            <allFiles>true</allFiles>
        </path>
    </match>
    <precedence>0</precedence>
    <clientRequest>
        <cacheIndex>
            <tunnelUnMatched>>false</tunnelUnMatched>
        </cacheIndex>
    </clientRequest>
    <originServerResolution>USE_HOST_HEADER</originServerResolution>
    <useClientIP>true</useClientIP>
</properties>
<VPControlDefinition>
    <id>1</id>
    <name>vp1</name>
</VPControlDefinition>
<cacheTuningPolicy>
    <id>1</id>
    <name>ctp1</name>
</cacheTuningPolicy>
<tunnelAll>>false</tunnelAll>
<revalidateCacheHit>
    <enable>>false</enable>
</revalidateCacheHit>

```

```
</contentDirectDefinition>
</contentDirectDefinitions>
</contentDirectDefinitionCatalog>
```

Sample Schema for HTTP Reverse Proxy Service

```
<?xml version="1.0" encoding="UTF-8" standalone="yes"?>
<rproxyDefinitionCatalog>
  <accessLogProfiles>
    <accessLogProfile>
      <id>1</id>
      <name>alp1</name>
      <description></description>
      <createdBy>super</createdBy>
      <lastModifiedBy>super</lastModifiedBy>
      <lastModifiedDateTime>2012-11-07T06:27:49Z</lastModifiedDateTime>
      <systemDefined>>false</systemDefined>
      <logProfileType>ACCESSLOG_PROFILE</logProfileType>
      <fileName>access.log</fileName>
      <logRotation>
        <maxLogFileSizeInMiB>100</maxLogFileSizeInMiB>
        <timeIntervalInMins>15</timeIntervalInMins>
      </logRotation>
      <exportSetting>
        <path>scp://1.1.1.1/path</path>
      </exportSetting>
      <logRecordFormat>
        <type>CUSTOM</type>
        <columns size="34">
          <column order="1">
            <fixedColumn>
              <field>BYTES_OUT_NO_HEADER</field>
            </fixedColumn>
          </column>
          <column order="2">
            <fixedColumn>
              <field>CACHE_HIT_INDICATOR</field>
            </fixedColumn>
          </column>
          <column order="3">
            <fixedColumn>
              <field>CACHE_HIT_HISTORY</field>
            </fixedColumn>
          </column>
          <column order="4">
            <fixedColumn>
              <field>FILENAME</field>
            </fixedColumn>
          </column>
          <column order="5">
            <fixedColumn>
              <field>REMOTE_HOST</field>
            </fixedColumn>
          </column>
          <column order="6">
            <fixedColumn>
```

```
        <field>REQUEST_METHOD</field>
      </fixedColumn>
    </column>
    <column order="7">
      <fixedColumn>
        <field>OBJECT_HOTNESS</field>
      </fixedColumn>
    </column>
    <column order="8">
      <fixedColumn>
        <field>QUERY_STRING</field>
      </fixedColumn>
    </column>
    <column order="9">
      <fixedColumn>
        <field>STATUS</field>
      </fixedColumn>
    </column>
    <column order="10">
      <fixedColumn>
        <field>TIMESTAMP</field>
      </fixedColumn>
    </column>
    <column order="11">
      <fixedColumn>
        <field>REMOTE_USER</field>
      </fixedColumn>
    </column>
    <column order="12">
      <fixedColumn>
        <field>SERVER_NAME</field>
      </fixedColumn>
    </column>
    <column order="13">
      <fixedColumn>
        <field>STATUS_SUBCODE</field>
      </fixedColumn>
    </column>
    <column order="14">
      <fixedColumn>
        <field>CONNECTION_TYPE</field>
      </fixedColumn>
    </column>
    <column order="15">
      <fixedColumn>
        <field>REQUEST_IN_TIME</field>
      </fixedColumn>
    </column>
    <column order="16">
      <fixedColumn>
        <field>FIRST_BYTE_OUT_TIME</field>
      </fixedColumn>
    </column>
    <column order="17">
      <fixedColumn>
        <field>TIME_USED_MS</field>
      </fixedColumn>
    </column>
```

```
</fixedColumn>
</column>
<column order="18">
  <fixedColumn>
    <field>TIME_USED_SEC</field>
  </fixedColumn>
</column>
<column order="19">
  <fixedColumn>
    <field>LAST_BYTE_OUT_TIME</field>
  </fixedColumn>
</column>
<column order="20">
  <fixedColumn>
    <field>REQUEST_PROTOCOL</field>
  </fixedColumn>
</column>
<column order="21">
  <fixedColumn>
    <field>BYTES_IN</field>
  </fixedColumn>
</column>
<column order="22">
  <fixedColumn>
    <field>LATENCY_TO_FIRST_BYTE_OUT</field>
  </fixedColumn>
</column>
<column order="23">
  <fixedColumn>
    <field>DATA_OUT_MS</field>
  </fixedColumn>
</column>
<column order="24">
  <fixedColumn>
    <field>NAMESPACE_NAME</field>
  </fixedColumn>
</column>
<column order="25">
  <fixedColumn>
    <field>BYTES_OUT</field>
  </fixedColumn>
</column>
<column order="26">
  <fixedColumn>
    <field>CACHE_REVALIDATE</field>
  </fixedColumn>
</column>
<column order="27">
  <fixedColumn>
    <field>URL</field>
  </fixedColumn>
</column>
<column order="28">
  <fixedColumn>
    <field>HTTP_HOST</field>
  </fixedColumn>
</column>
```



```

</column>
<column order="29">
  <fixedColumn>
    <field>REMOTE_ADDR</field>
  </fixedColumn>
</column>
<column order="30">
  <fixedColumn>
    <field>LOCAL_ADDR</field>
  </fixedColumn>
</column>
<column order="31">
  <fixedColumn>
    <field>SERVER_PORT</field>
  </fixedColumn>
</column>
<column order="32">
  <fixedColumn>
    <field>ORIGIN_FETCHED_SIZE</field>
  </fixedColumn>
</column>
<column order="33">
  <fixedColumn>
    <field>REQUEST_LINE</field>
  </fixedColumn>
</column>
<column order="34">
  <fixedColumn>
    <field>ORIGIN_SERVER_NAME</field>
  </fixedColumn>
</column>
</columns>
</logRecordFormat>
<doNotLog>
  <objectSizeInBytes>100</objectSizeInBytes>
  <httpResponseCodes size="0"/>
</doNotLog>
</accessLogProfile>
</accessLogProfiles>
<virtualPlayerDefinitions>
  <genericPlayerDefinitions>
    <id>1</id>
    <name>vp1</name>
    <createdBy>super</createdBy>
    <lastModifiedBy>super</lastModifiedBy>
    <lastModifiedDateTime>2012-11-07T06:25:30Z</lastModifiedDateTime>
    <systemDefined>>false</systemDefined>
    <type>generic</type>
    <seekConfiguration>
      <startIdentifier>start</startIdentifier>
      <endIdentifier>end</endIdentifier>
      <tunnelSeekRequest>true</tunnelSeekRequest>
    </seekConfiguration>
    <maxConnBandwidth>
      <noLimit>true</noLimit>
    </maxConnBandwidth>
  </genericPlayerDefinitions>
</virtualPlayerDefinitions>

```

```

    <fastStart>
      <defaultSizeinKB>100</defaultSizeinKB>
    </fastStart>
    <fullDownload>
      <always>true</always>
    </fullDownload>
  </genericPlayerDefinitions>
</virtualPlayerDefinitions>
<cacheTuningPolicies>
  <cacheTuningPolicy>
    <id>1</id>
    <name>ctp1</name>
    <description></description>
    <createdBy>super</createdBy>
    <lastModifiedBy>super</lastModifiedBy>
    <lastModifiedDateTime>2012-11-07T06:26:31Z</lastModifiedDateTime>
    <systemDefined>>false</systemDefined>
    <ageSizeSettings>
      <ageThreshold>120</ageThreshold>
      <defaultCacheAge>120</defaultCacheAge>
      <minObjSize>4096</minObjSize>
      <diskIngestThreshold>0</diskIngestThreshold>
    </ageSizeSettings>
    <cacheExclusion>
      <queryString>true</queryString>
      <cacheRequestsWithCookies>
        <enable>true</enable>
        <option>VALIDATE_WITH_ORIGIN</option>
      </cacheRequestsWithCookies>
      <isDomainName>true</isDomainName>
      <queryStringInclusion>INCLUDE_QUERY_STRING</queryStringInclusion>
      <cacheRequestWithSpecificHeaders>
        <enable>true</enable>
        <headers>
          <header>AUTH_HEADER</header>
          <header>COOKIE_HEADER</header>
          <header>CACHECONTROL_HEADER</header>
        </headers>
      </cacheRequestWithSpecificHeaders>
    </cacheExclusion>
    <originFetch>
      <cacheAgeSettings>
        <cacheAgeList/>
        <contentAnyAgeInSecs>120</contentAnyAgeInSecs>
      </cacheAgeSettings>
      <noCacheDirective>FOLLOW</noCacheDirective>
      <overrideTunnelObjectExpired>true</overrideTunnelObjectExpired>
      <objectSizeMinimumThreshold>0</objectSizeMinimumThreshold>
      <objectSizeMaximumThreshold>0</objectSizeMaximumThreshold>
      <cacheFill>AGGRESSIVE</cacheFill>
      <ignoreNoTransformHeader>true</ignoreNoTransformHeader>
    </originFetch>
  </cacheTuningPolicy>
</cacheTuningPolicies>
<responsesWithHTTP302Status>PASS_THROUGH_TO</responsesWithHTTP302Status>

  <cacheIngestHotnessThreshold>3</cacheIngestHotnessThreshold>
  <uriDepthThreshold>10</uriDepthThreshold>

```

```

</originFetch>
<originRequest>
  <cacheRevalidation>
    <permit>true</permit>
    <useHEADMethod></useHEADMethod>
  </cacheRevalidation>
  <setXForwardedForHeader>true</setXForwardedForHeader>
</originRequest>
<clientRequest>
  <revalidationOverride>
    <overrideMaxAgeHeader>>false</overrideMaxAgeHeader>
  </revalidationOverride>
</clientRequest>
<diskCacheProperties>
  <SAS/>
  <SATA/>
  <SSD/>
</diskCacheProperties>
</cacheTuningPolicy>
</cacheTuningPolicies>
<originMaps>
  <originEscalationMap>
    <id>2</id>
    <name>Em1</name>
    <description></description>
    <type>ORIGIN_ESCALATION_MAP</type>
    <createdBy>super</createdBy>
    <lastModifiedBy>super</lastModifiedBy>
    <lastModifiedDateTime>2012-11-07T06:33:51Z</lastModifiedDateTime>
    <nodeMonitoring>
      <allowedFails>3</allowedFails>
      <heartBeatIntervalInMillis>100</heartBeatIntervalInMillis>
      <connectTimeoutInMillis>100</connectTimeoutInMillis>
      <readTimeoutInMillis>100</readTimeoutInMillis>
    </nodeMonitoring>
    <connectionSettings>
      <connectTimeOut>100</connectTimeOut>
      <readTimeOut>100</readTimeOut>
      <connectRetryDelay>100</connectRetryDelay>
      <readRetryDelay>100</readRetryDelay>
    </connectionSettings>
    <devices size="1">
      <device order="0">
        <nonMFDevice>
          <address>20.1.1.11</address>
          <type>NON_MFDEVICE</type>
          <port>80</port>
          <heartBeatPath>/root/heartbeat.html</heartBeatPath>
          <failureCodes size="3">
            <value>404</value>
            <value>500</value>
            <value>505</value>
          </failureCodes>
        </nonMFDevice>
        <type>NON_MFDEVICE</type>
      </device>
    </devices>
  </originEscalationMap>
</originMaps>

```

```
</devices>
</originEscalationMap>
</originMaps>
<rproxyDefinitions>
  <rproxyDefinition>
    <id>1</id>
    <name>rrp1</name>
    <description></description>
    <createdBy>super</createdBy>
    <lastModifiedBy>super</lastModifiedBy>
    <lastModifiedDateTime>2012-11-07T06:35:59Z</lastModifiedDateTime>
    <systemDefined>false</systemDefined>
    <properties>
      <domain>google</domain>
      <isDomainRegex>false</isDomainRegex>
      <orderOfServing>LIFO</orderOfServing>

<ignoreAllObjectsCorrelationValidators>true</ignoreAllObjectsCorrelationValidators>

    <match>
      <path>
        <allFiles>true</allFiles>
      </path>
    </match>
    <precedence>0</precedence>
    <clientRequest/>
    <originServer>
      <originMap>
        <id>2</id>
        <name>Em1</name>
      </originMap>
      <useSecureMode>true</useSecureMode>
    </originServer>
    <cachePinning>
      <enableAutoPin>false</enableAutoPin>
      <pinHeaderName>header</pinHeaderName>
      <maxObjectSizeInKB>100</maxObjectSizeInKB>
      <maxCacheCapacityInGB>10</maxCacheCapacityInGB>
      <validityBeginHdr>VAL_BEGIN</validityBeginHdr>
    </cachePinning>
    <clientResponse>
      <headersToDelete size="1">
        <value>via</value>
      </headersToDelete>
      <headersToAdd size="0"/>
      <addRetryHeaderAfterSecs>
        <addRetryHeader>true</addRetryHeader>
        <retryHeaderValueInSecs>100</retryHeaderValueInSecs>
      </addRetryHeaderAfterSecs>
    </clientResponse>
    <originRequest>
      <inheritHostHeaderValue>true</inheritHostHeaderValue>
      <headersToAdd size="0"/>
    </originRequest>
  </properties>
</VPCControlDefinition>
```

```
<id>1</id>
<name>vp1</name>
</VPControlDefinition>
<cacheTuningPolicy>
  <id>1</id>
  <name>ctp1</name>
</cacheTuningPolicy>
<tunnelAll>true</tunnelAll>
<logProfile>
  <id>1</id>
  <name>alp1</name>
</logProfile>
<policyScript>
  <id>1</id>
  <name>dest_ip</name>
</policyScript>
</rproxyDefinition>
</rproxyDefinitions>
</rproxyDefinitionCatalog>
```

See these topics for additional information:

- [Table 5 on page 106](#)
- [Creating Network Optimization Service XML Files for Import on page 105](#)
- [Table 6 on page 125](#)
- [Creating Reverse Proxy Service XML Files for Import on page 123](#)
- [Media Flow Activate Overview on page 3](#)
- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

CHAPTER 19

Quick Reference

- [Quick Reference to Tasks in Media Flow Activate on page 191](#)

Quick Reference to Tasks in Media Flow Activate

This topic provides a list of tasks that you can perform using Media Flow Activate and links to corresponding sections within the *Junos Space Media Flow Activate Management Guide*.

Feature	Task	Link
Access log profile	Associating an access log profile with an HTTP reverse proxy service	<ul style="list-style-type: none">• Creating HTTP Reverse Proxy Services on page 112• Sample XML Schema for Network Optimization and Reverse Proxy Services on page 179
	Creating an access log profile	“Creating Access Log Profiles” on page 45
	Deleting an access log profile	“Actions on Access Log Profiles” on page 49
	Modifying an access log profile	
	Viewing an access log profile	
Audit logs	Monitoring audit logs	“Audit Logs Workspace Overview” on page 167
BGP-based traffic steering	Configuring BGP-based traffic steering in a Media Flow Controller device	“Configuring BGP-Based Traffic Steering” on page 27
Bonded interface	Configuring a bonded interface	“Configuring Bonded Interfaces” on page 16

Feature	Task	Link
Cache-tuning policy	Associating a cache-tuning policy with a Network Optimization service	<ul style="list-style-type: none"> • Creating Network Optimization Services on page 98 • Sample XML Schema for Network Optimization and Reverse Proxy Services on page 179
	Associating a cache-tuning policy with an HTTP reverse proxy service	<ul style="list-style-type: none"> • Creating HTTP Reverse Proxy Services on page 112 • Sample XML Schema for Network Optimization and Reverse Proxy Services on page 179
	Creating a cache-tuning policy	“Creating Cache-Tuning Policies” on page 52
	Modifying a cache-tuning policy	“Actions on Cache-Tuning Policies” on page 65
	Deleting a cache-tuning policy	
Configuration templates	Managing configuration templates	“Configuration Templates Overview” on page 149
Content ingest service	Activating a content ingest service	“Managing Provisioned Services” on page 144
	Creating a content ingest service	“Creating Content Ingest Services” on page 135
	Deactivating a content ingest service	“Managing Provisioned Services” on page 144
	Deleting a content ingest service	“Actions on Services” on page 137
	Deprovisioning a content ingest service	“Managing Provisioned Services” on page 144
	Modifying a content ingest service	“Actions on Services” on page 137
	Provisioning a content ingest service	“Provisioning Services” on page 142
	Reprovisioning a content ingest service	“Managing Provisioned Services” on page 144
	Tagging, untagging, and viewing tags	<ul style="list-style-type: none"> • Actions on Services on page 137 • Tagging Media Flow Controller Objects on page 30
	Viewing a content ingest service	“Actions on Services” on page 137
Fault monitoring		“Fault Monitoring with SNMP” on page 157

Feature	Task	Link
HTTP reverse proxy service	Activating an HTTP reverse proxy service	“Managing Provisioned Services” on page 144
	Creating an HTTP reverse proxy service	“Creating HTTP Reverse Proxy Services” on page 112
	Copying an HTTP reverse proxy service	“Actions on Services” on page 137
	Deactivating an HTTP reverse proxy service	“Managing Provisioned Services” on page 144
	Deleting an HTTP reverse proxy service	“Actions on Services” on page 137
	Deprovisioning an HTTP reverse proxy service	“Managing Provisioned Services” on page 144
	Exporting an HTTP reverse proxy service	<ul style="list-style-type: none"> • Actions on Services on page 137 • Creating Reverse Proxy Service XML Files for Export on page 132
	Importing an HTTP reverse proxy service	<ul style="list-style-type: none"> • Actions on Services on page 137 • Creating Reverse Proxy Service XML Files for Import on page 123 • Sample XML Schema for Network Optimization and Reverse Proxy Services on page 179
	Modifying an HTTP reverse proxy service	“Actions on Services” on page 137
	Provisioning: <ul style="list-style-type: none"> • A service • A resource pool 	“Provisioning Services” on page 142
	Purging content	“Purging Content from Media Flow Controller Devices” on page 133
	Reprovisioning an HTTP reverse proxy service	“Managing Provisioned Services” on page 144
	Tagging, untagging, and viewing tags	<ul style="list-style-type: none"> • Actions on Services on page 137 • Tagging Media Flow Controller Objects on page 30

Feature	Task	Link
Media Flow Controller device	Configuring BGP-based traffic steering in a Media Flow Controller device	“Configuring BGP-Based Traffic Steering” on page 27
	Configuring a bonded interface	“Configuring Bonded Interfaces” on page 16
	Configuring an interface	“Configuring MFC Device Interfaces” on page 14
	Configuring the log pull method	“Configuring Devices” on page 18
	Launching Media Flow Controller Console from Media Flow Activate	“Launching the Media Flow Controller Secure Console from MFA” on page 12
	Monitoring the Media Flow Controller dashboard	“Monitoring a Media Flow Controller Dashboard from MFA” on page 11
	Replicating device configuration	“Replicating Media Flow Controller Device Configuration” on page 24
	Restarting a device	“Restarting Media Flow Controller Devices or Services” on page 29
	Restarting a service	
	Restoring device configuration	“Restoring Media Flow Controller Device Configuration” on page 22
	Tagging, untagging, and viewing tags	“Tagging Media Flow Controller Objects” on page 30
	Upgrading or rolling back the Media Flow Controller software image	“Upgrading or Rolling Back the Media Flow Controller Software Image” on page 21

Feature	Task	Link
Network Optimization service	Activating a Network Optimization proxy service	“Managing Provisioned Services” on page 144
	Creating a Network Optimization proxy service	“Creating Network Optimization Services” on page 98
	Copying a Network Optimization proxy service	“Actions on Services” on page 137
	Deactivating a Network Optimization proxy service	“Managing Provisioned Services” on page 144
	Deleting a Network Optimization proxy service	“Actions on Services” on page 137
	Deprovisioning a Network Optimization proxy service	“Managing Provisioned Services” on page 144
	Exporting a Network Optimization proxy service	“Actions on Services” on page 137
	Importing a Network Optimization proxy service	<ul style="list-style-type: none"> • Actions on Services on page 137 • Creating Network Optimization Service XML Files for Import on page 105 • Sample XML Schema for Network Optimization and Reverse Proxy Services on page 179
	Modifying a Network Optimization proxy service	“Actions on Services” on page 137
	Provisioning a Network Optimization proxy service	“Provisioning Services” on page 142
	Reprovisioning a Network Optimization proxy service	“Managing Provisioned Services” on page 144
	Tagging, untagging, and viewing tags	<ul style="list-style-type: none"> • Actions on Services on page 137 • Tagging Media Flow Controller Objects on page 30

Feature	Task	Link
Origin map	Associating an origin map with an HTTP reverse proxy service	<ul style="list-style-type: none"> • Creating HTTP Reverse Proxy Services on page 112 • Sample XML Schema for Network Optimization and Reverse Proxy Services on page 179
	Creating a consistent hash map	“Creating Consistent Hash Maps” on page 69
	Creating an escalation map	“Creating Escalation Maps” on page 72
	Deleting an origin map	“Actions on Origin Maps” on page 75
	Modifying an origin map	
	Viewing an origin map	
Policy script	Associating a policy script with a Network Optimization proxy service	“Creating Network Optimization Services” on page 98
	Associating a policy script with an HTTP reverse proxy service	“Creating HTTP Reverse Proxy Services” on page 112
	Adding a policy script	“Adding Policy Scripts” on page 78
	Deleting a policy script	“Actions on Policy Scripts” on page 79
	Exporting a policy script	
	Modifying a policy script	

Feature	Task	Link
Resource pool	Associating a resource pool with an HTTP reverse proxy service	“Provisioning Services” on page 142
	Creating a resource pool	“Creating Resource Pools” on page 34
	Deleting a resource pool	“Actions on Resource Pools” on page 35
	Deprovisioning a resource pool	“Managing Resource Pool Associations” on page 37
	Modifying a resource pool	“Actions on Resource Pools” on page 35
	Managing resource pool associations	“Managing Resource Pool Associations” on page 37
	Provisioning a resource pool	“Provisioning Resource Pools to MFC Devices” on page 36
	Reprovisioning a resource pool	“Managing Resource Pool Associations” on page 37
	Tagging, untagging, and viewing tags	<ul style="list-style-type: none"> • Actions on Resource Pools on page 35 • Tagging Media Flow Controller Objects on page 30
Virtual player	Associating a virtual player with a Network Optimization service	<ul style="list-style-type: none"> • Creating Network Optimization Services on page 98 • Sample XML Schema for Network Optimization and Reverse Proxy Services on page 179
	Associating a virtual player with an HTTP reverse proxy service	<ul style="list-style-type: none"> • Creating HTTP Reverse Proxy Services on page 112 • Sample XML Schema for Network Optimization and Reverse Proxy Services on page 179
	Creating a virtual player	“Creating Virtual Players” on page 83
	Copying a virtual player	“Actions on Virtual Players” on page 88
	Modifying a virtual player	
	Deleting a virtual player	

- Related Documentation**
- [Media Flow Activate Overview on page 3](#)
 - [Service Design Overview on page 92](#)
 - [Provisioning Services Overview on page 141](#)

- [Job Management Workspace Overview on page 175](#)

PART 11

Index

- [Index on page 201](#)

Index

Symbols

#, comments in configuration statements.....	xiii
(), in syntax descriptions.....	xiii
< >, in syntax descriptions.....	xiii
[], in configuration statements.....	xiii
{ }, in configuration statements.....	xiii
(pipe), in syntax descriptions.....	xiii

A

access log profile	
actions	49
configuring	45
understanding	41
action	
access log profile.....	49
cache-tuning policy.....	65
origin map.....	75
policy script.....	79
resource pool.....	35
service.....	137
virtual player.....	88
adding	
policy script.....	78
audit log.....	167

B

BGP traffic steering	
overview	26
bonded interface	
configuring for Media Flow Controller.....	16
braces, in configuration statements.....	xiii
brackets	
angle, in syntax descriptions.....	xiii
square, in configuration statements.....	xiii

C

cache-tuning policy	
about cache handling options.....	51
about “Hotness”.....	51
actions	65

configuring	52
understanding	51
comments, in configuration statements.....	xiii
configuration template.....	149
configuring	
access log profile.....	45
BGP based traffic steering.....	27
cache-tuning policy.....	52
consistent hash map.....	69
Content Ingest services.....	135
escalation map.....	72
HTTP Reverse Proxy service.....	112
XML file for export.....	132
XML file for import.....	123
Media Flow Controller bonded interface.....	16
Media Flow Controller device.....	18
Media Flow Controller device interface.....	14
Network Optimization service.....	98
XML file for import.....	105
resource pool.....	34
virtual player.....	83
consistent hash map	
configuring	69
content	
purging	133
Content Ingest services	
configuring.....	135
conventions	
text and syntax.....	xii
curly braces, in configuration statements.....	xiii
customer support.....	xiv
contacting JTAC.....	xiv

D

designing	
HTTP Reverse Proxy service.....	111
Network Optimization service.....	97
device	
configuring BGP based traffic steering.....	27
configuring for Media Flow Controller.....	18
device interface	
configuring for Media Flow Controller.....	14
device template.....	149
dynamic URI remapping	
understanding.....	92

E

escalation map	
configuring	72

exporting	
reverse proxy service XML file.....	132

F

fault	
monitoring.....	157
font conventions.....	xii

H

HTTP Reverse Proxy service	
about	111
configuring.....	112
provisioning.....	142

I

importing	
Network Optimization service XML file.....	105
reverse proxy service XML file.....	123

J

job management.....	175
---------------------	-----

L

launching	
a secure console.....	12
Media Flow Controller dashboard.....	11

M

managing	
provisioned sites.....	144
resource pool.....	37
Media Flow Activate	
about HTTP Reverse Proxy service.....	111
about Network Optimization service.....	97
access log profile.....	41
adding	
policy script.....	78
audit log.....	167
BGP traffic steering.....	26
cache-tuning policy.....	51
configuration template.....	149
configuring	
access log profile.....	45
BGP based traffic steering.....	27
cache-tuning policy.....	52
consistent hash map.....	69
escalation map.....	72
HTTP Reverse Proxy service.....	112

Media Flow Controller bonded	
interface.....	16
Media Flow Controller device.....	18
Media Flow Controller device	
interface.....	14
Network Optimization service.....	98
resource pool.....	34
virtual player.....	83
configuring Content Ingest services.....	135
device template.....	149
discovering device.....	3
fault monitoring.....	157
job management.....	175
launch Media Flow Controller dashboard.....	11
launch secure console.....	12
managing	
provisioned device.....	144
resource pool.....	37
Media Flow Controller management.....	5
Media Flow Controller version requirement.....	3
Network Optimization service XML file for	
import.....	105
origin map.....	68
overview.....	3
policy script.....	77
provisioning	
resource pool.....	36
services.....	142
purging content.....	133
removing Media Flow Controller	
configuration.....	22
replicating Media Flow Controller	
configuration.....	24
resource pool.....	33
actions.....	35
restart Media Flow Controller device or	
service.....	29
restoring Media Flow Controller	
configuration.....	22
reverse proxy service XML file for	
export.....	132
import.....	123
rolling back Media Flow Controller.....	21
sample XML schema.....	179
service actions.....	137
Service Design.....	92
Service Provisioning.....	141
tagging object.....	30
task list.....	191

upgrading Media Flow Controller.....	21	overview	33
virtual player.....	81	provisioning.....	36
actions.....	88	restarting	
XML schema.....	179	Media Flow Controller device.....	29
monitoring		Media Flow Controller service.....	29
Media Flow Controller.....	11	restoring	
performance.....	157	Media Flow Controller configuration.....	22
N		rolling back	
Network Optimization		Media Flow Controller.....	21
provisioning service.....	142	S	
Network Optimization service		secure console	
about	97	launching.....	12
configuring.....	98	service	
O		actions	137
object		provisioning.....	142
tagging	30	Service Design	
origin map		managing provisioned device.....	144
actions	75	overview.....	92
understanding	68	understanding dynamic URI remapping.....	92
P		using Precedence.....	92
parentheses, in syntax descriptions.....	xiii	Service Provisioning	
policy script		overview.....	141
actions	79	support, technical See technical support	
adding	78	syntax conventions.....	xii
understanding	77	T	
Precedence		tagging	
using in Service Design.....	92	object.....	30
provisioning		task list.....	191
resource pool.....	36	technical support	
service.....	142	contacting JTAC.....	xiv
purging		U	
content.....	133	upgrading	
R		Media Flow Controller.....	21
reference		V	
sample XML schema.....	179	virtual player	
task list.....	191	actions	88
removing		configuring	83
Media Flow Controller configuration.....	22	understanding.....	81
replicating		using Hash Verify.....	81
Media Flow Controller configuration.....	24	X	
resource pool		XML schema.....	179
actions	35		
configuring	34		
managing.....	37		

