

# Design Elements



---

Published: 2013-06-20

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

Copyright © 2013, Juniper Networks, Inc. All rights reserved.

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

Juniper Networks assumes no responsibility for any inaccuracies in this document. Juniper Networks reserves the right to change, modify, transfer, or otherwise revise this publication without notice.

Products made or sold by Juniper Networks or components thereof might be covered by one or more of the following patents that are owned by or licensed to Juniper Networks: U.S. Patent Nos. 5,473,599, 5,905,725, 5,909,440, 6,192,051, 6,333,650, 6,359,479, 6,406,312, 6,429,706, 6,459,579, 6,493,347, 6,538,518, 6,538,899, 6,552,918, 6,567,902, 6,578,186, and 6,590,785.

#### *Design Elements*

Copyright © 2013, Juniper Networks, Inc.  
All rights reserved.

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

#### YEAR 2000 NOTICE

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

#### END USER LICENSE AGREEMENT

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 the Documentation . . . . .	ix
	Documentation and Release Notes . . . . .	ix
	Documentation Conventions . . . . .	ix
	Documentation Feedback . . . . .	xi
	Requesting Technical Support . . . . .	xi
	Self-Help Online Tools and Resources . . . . .	xii
	Opening a Case with JTAC . . . . .	xii
<b>Part 1</b>	<b>Overview</b>	
<b>Chapter 1</b>	<b>Understanding Media Flow Activate Design Elements . . . . .</b>	<b>3</b>
	Understanding Access Log Profiles . . . . .	3
	Understanding Cache-Tuning Policies . . . . .	7
	Understanding Origin Maps . . . . .	8
	Understanding Policy Scripts . . . . .	9
	Understanding Virtual Players . . . . .	10
<b>Part 2</b>	<b>Administration</b>	
<b>Chapter 2</b>	<b>Managing Media Flow Activate Design Elements . . . . .</b>	<b>15</b>
	Actions on Access Log Profiles . . . . .	15
	Actions on Cache-Tuning Policies . . . . .	16
	Actions on Origin Maps . . . . .	16
	Actions on Policy Scripts . . . . .	17
	Actions on Virtual Players . . . . .	18
<b>Part 3</b>	<b>Configuration</b>	
<b>Chapter 3</b>	<b>Configuring Media Flow Activate Design Elements . . . . .</b>	<b>21</b>
	Creating Access Log Profiles . . . . .	21
	Creating Cache-Tuning Policies . . . . .	25
	Creating Consistent Hash Maps . . . . .	39
	Creating Escalation Maps . . . . .	43
	Adding Policy Scripts . . . . .	46
	Creating Virtual Players . . . . .	47
<b>Part 4</b>	<b>Index</b>	
	Index . . . . .	55



# List of Figures

<b>Part 3</b>	<b>Configuration</b>	
<b>Chapter 3</b>	<b>Configuring Media Flow Activate Design Elements . . . . .</b>	<b>21</b>
	Figure 1: Add Policy—Cache tier Tab . . . . .	29
	Figure 2: Add Policy—Expiry & revalidation Tab . . . . .	33
	Figure 3: Add Policy—Tunnel override decision Tab . . . . .	36
	Figure 4: Add Virtual Player Window—Connection Properties Tab . . . . .	49



# List of Tables

	<b>About the Documentation . . . . . ix</b>
	Table 1: Notice Icons . . . . . x
	Table 2: Text and Syntax Conventions . . . . . x
<b>Part 1</b>	<b>Overview</b>
<b>Chapter 1</b>	<b>Understanding Media Flow Activate Design Elements . . . . . 3</b>
	Table 3: Reason Codes Recorded on System Log for Access Log Files . . . . . 6





# About the Documentation

- Documentation and Release Notes on page ix
- Documentation Conventions on page ix
- Documentation Feedback on page xi
- Requesting Technical Support on page xi

## Documentation and Release Notes

---

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

If the information in the latest release notes differs from the information in the documentation, follow the product Release Notes.

Juniper Networks Books publishes books by Juniper Networks engineers and subject matter experts. These books go beyond the technical documentation to explore the nuances of network architecture, deployment, and administration. The current list can be viewed at <http://www.juniper.net/books>.

## Documentation Conventions

---

Table 1 on page x 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 x defines the text and syntax conventions used in this guide.

Table 2: Text and Syntax Conventions

Convention	Description	Examples
<b>Bold text like this</b>	Represents text that you type.	To enter configuration mode, type the <b>configure</b> command:  <code>user@host&gt; configure</code>
Fixed-width text like this	Represents output that appears on the terminal screen.	<code>user@host&gt; show chassis alarms</code> <code>No alarms currently active</code>
<i>Italic text like this</i>	<ul style="list-style-type: none"> <li>Introduces or emphasizes important new terms.</li> <li>Identifies book names.</li> <li>Identifies RFC and Internet draft titles.</li> </ul>	<ul style="list-style-type: none"> <li>A policy <i>term</i> is a named structure that defines match conditions and actions.</li> <li><i>Junos OS System Basics Configuration Guide</i></li> <li>RFC 1997, <i>BGP Communities Attribute</i></li> </ul>
<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:  <code>[edit]</code> <code>root@# set system domain-name <i>domain-name</i></code>
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"> <li>To configure a stub area, include the <b>stub</b> statement at the <code>[edit protocols ospf area area-id]</code> hierarchy level.</li> <li>The console port is labeled <b>CONSOLE</b>.</li> </ul>
< > (angle brackets)	Enclose optional keywords or variables.	<code>stub &lt;default-metric <i>metric</i>&gt;;</code>

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

Convention	Description	Examples
(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.	<b>broadcast   multicast</b>  <i>(string1   string2   string3)</i>
# (pound sign)	Indicates a comment specified on the same line as the configuration statement to which it applies.	<b>rsvp { # Required for dynamic MPLS only</b>
[ ] (square brackets)	Enclose a variable for which you can substitute one or more values.	<b>community name members [ community-ids ]</b>
Indentation and braces ( { } )	Identify a level in the configuration hierarchy.	<b>[edit]</b> routing-options { static { route default { nexthop <i>address</i> ; retain; } } }
;(semicolon)	Identifies a leaf statement at a configuration hierarchy level.	
<b>GUI Conventions</b>		
<b>Bold text like this</b>	Represents graphical user interface (GUI) items you click or select.	<ul style="list-style-type: none"> <li>In the Logical Interfaces box, select <b>All Interfaces</b>.</li> <li>To cancel the configuration, click <b>Cancel</b>.</li> </ul>
> (bold right angle bracket)	Separates levels in a hierarchy of menu selections.	In the configuration editor hierarchy, select <b>Protocols&gt;Ospf</b> .

## 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](mailto: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

# Overview

- [Understanding Media Flow Activate Design Elements on page 3](#)



## CHAPTER 1

# Understanding Media Flow Activate Design Elements

- [Understanding Access Log Profiles on page 3](#)
- [Understanding Cache-Tuning Policies on page 7](#)
- [Understanding Origin Maps on page 8](#)
- [Understanding Policy Scripts on page 9](#)
- [Understanding Virtual Players on page 10](#)

## 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 21](#)
- [Actions on Access Log Profiles on page 15](#)
- [Media Flow Activate Overview](#)
- [Understanding Media Flow Controller Management with Media Flow Activate](#)
- [Quick Reference to Tasks in Media Flow Activate](#)

## 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 25](#)
- [Actions on Cache-Tuning Policies on page 16](#)
- [Media Flow Activate Overview](#)
- [Understanding Media Flow Controller Management with Media Flow Activate](#)
- [Quick Reference to Tasks in Media Flow Activate](#)

## 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 39](#)
- [Creating Escalation Maps on page 43](#)
- [Actions on Origin Maps on page 16](#)
- [Media Flow Activate Overview](#)
- [Quick Reference to Tasks in Media Flow Activate](#)

## 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 46](#)
- [Actions on Policy Scripts on page 17](#)
- [Media Flow Activate Overview](#)
- [Quick Reference to Tasks in Media Flow Activate](#)

## 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 47](#)
- [Actions on Virtual Players on page 18](#)
- [Media Flow Activate Overview](#)
- [Understanding Media Flow Controller Management with Media Flow Activate](#)
- [Quick Reference to Tasks in Media Flow Activate](#)





## PART 2

# Administration

- [Managing Media Flow Activate Design Elements on page 15](#)



## CHAPTER 2

# Managing Media Flow Activate Design Elements

- [Actions on Access Log Profiles on page 15](#)
- [Actions on Cache-Tuning Policies on page 16](#)
- [Actions on Origin Maps on page 16](#)
- [Actions on Policy Scripts on page 17](#)
- [Actions on Virtual Players on page 18](#)

### 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 3](#)
- [Creating Access Log Profiles on page 21](#)
- *Media Flow Activate Overview*
- *Understanding Media Flow Controller Management with Media Flow Activate*
- *Quick Reference to Tasks in Media Flow Activate*

---

## 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 7](#)
- [Creating Cache-Tuning Policies on page 25](#)
- *Media Flow Activate Overview*
- *Understanding Media Flow Controller Management with Media Flow Activate*
- *Quick Reference to Tasks in Media Flow Activate*

---

## 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 39](#)
- [Creating Escalation Maps on page 43](#)
- [Understanding Origin Maps on page 8](#)
- [Media Flow Activate Overview](#)
- [Quick Reference to Tasks in Media Flow Activate](#)

---

## 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 9](#)
- [Adding Policy Scripts on page 46](#)
- [Media Flow Activate Overview](#)

- *Quick Reference to Tasks in Media Flow Activate*

## 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 47](#)
- [Understanding Virtual Players on page 10](#)
- *Media Flow Activate Overview*
- *Understanding Media Flow Controller Management with Media Flow Activate*
- *Quick Reference to Tasks in Media Flow Activate*

## PART 3

# Configuration

- [Configuring Media Flow Activate Design Elements on page 21](#)





## CHAPTER 3

# Configuring Media Flow Activate Design Elements

- [Creating Access Log Profiles on page 21](#)
- [Creating Cache-Tuning Policies on page 25](#)
- [Creating Consistent Hash Maps on page 39](#)
- [Creating Escalation Maps on page 43](#)
- [Adding Policy Scripts on page 46](#)
- [Creating Virtual Players on page 47](#)

## 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*.



**CAUTION:** No diskspace 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 15](#)
- [Understanding Access Log Profiles on page 3](#)
- [Media Flow Activate Overview](#)
- [Understanding Media Flow Controller Management with Media Flow Activate](#)
- [Quick Reference to Tasks in Media Flow Activate](#)

## 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 7](#)
- [Actions on Cache-Tuning Policies on page 16](#)
- [Media Flow Activate Overview](#)
- [Understanding Media Flow Controller Management with Media Flow Activate](#)
- [Quick Reference to Tasks in Media Flow Activate](#)

## 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.

- 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 MFC Node(s)** dialog box is displayed. Click **Yes** to remove the nodes or click **No** to exit without making any changes.
- 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.
  - 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 16](#)
- [Creating Escalation Maps on page 43](#)
- [Understanding Origin Maps on page 8](#)
- [Creating HTTP Reverse Proxy Services](#)
- [Media Flow Activate Overview](#)
- [Quick Reference to Tasks in Media Flow Activate](#)

## 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 16](#)
- [Creating Consistent Hash Maps on page 39](#)
- [Understanding Origin Maps on page 8](#)
- *Creating HTTP Reverse Proxy Services*
- *Media Flow Activate Overview*
- *Quick Reference to Tasks in Media Flow Activate*

## 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 17](#)
- [Understanding Policy Scripts on page 9](#)
- *Media Flow Activate Overview*
- *Quick Reference to Tasks in Media Flow Activate*

---

## 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 10 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 4,294,967,295 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 18](#)
- [Understanding Virtual Players on page 10](#)
- *Media Flow Activate Overview*
- *Understanding Media Flow Controller Management with Media Flow Activate*
- *Quick Reference to Tasks in Media Flow Activate*



## PART 4

# Index

- [Index on page 55](#)



# Index

## Symbols

#, comments in configuration statements.....	xi
( ), in syntax descriptions.....	xi
< >, in syntax descriptions.....	x
[ ], in configuration statements.....	xi
{ }, in configuration statements.....	xi
(pipe), in syntax descriptions.....	xi

## A

access log profile	
actions .....	15
configuring .....	21
understanding .....	3
action	
access log profile.....	15
cache-tuning policy.....	16
origin map.....	16
policy script.....	17
virtual player.....	18
adding	
policy script.....	46

## B

braces, in configuration statements.....	xi
brackets	
angle, in syntax descriptions.....	x
square, in configuration statements.....	xi

## C

cache-tuning policy	
about cache handling options.....	7
about “Hotness”.....	7
actions .....	16
configuring .....	25
understanding .....	7
comments, in configuration statements.....	xi
configuring	
access log profile.....	21
cache-tuning policy.....	25
consistent hash map.....	39

escalation map.....	43
virtual player.....	47
consistent hash map	
configuring .....	39
conventions	
text and syntax.....	x
curly braces, in configuration statements.....	xi
customer support.....	xi
contacting JTAC.....	xi

## D

documentation	
comments on.....	xi

## E

escalation map	
configuring .....	43

## F

font conventions.....	x
-----------------------	---

## M

manuals	
comments on.....	xi
Media Flow Activate	
access log profile.....	3
adding	
policy script.....	46
cache-tuning policy.....	7
configuring	
access log profile.....	21
cache-tuning policy.....	25
consistent hash map.....	39
escalation map.....	43
virtual player.....	47
origin map.....	8
policy script.....	9
virtual player.....	10
actions.....	18

## O

origin map	
actions .....	16
understanding .....	8

## P

parentheses, in syntax descriptions.....	xi
--	----

policy script	
actions .....	17
adding .....	46
understanding .....	9

## S

support, technical	See technical support
syntax conventions.....	x

## T

technical support	
contacting JTAC.....	xi

## V

virtual player	
actions .....	18
configuring .....	47
understanding.....	10
using Hash Verify.....	10