Wowza Media Cache

User Guide

Wowza Media Cache User Guide

Version: 4

www.wowza.com

This document is for informational purposes only and in no way shall be interpreted or construed to create warranties of any kind, either express or implied, regarding the information contained herein.

No Endorsement or Warranty for Third-Party Links and Software

This document contains links to third-party websites ("Linked Sites") that are not under the control of Wowza Media Systems™, LLC ("Wowza™"). Wowza is not responsible for the content on or operation of Linked Sites. If you access Linked Sites, you do so at your own risk and understand that Wowza accepts no responsibility or liability for the content or operation of Linked Sites. Wowza provides these links only as a convenience, and the inclusion of a link does not imply that Wowza endorses such Linked Sites or any content, products, or services available from Linked Sites.

This document also refers to third-party software that is not licensed, sold, or distributed by Wowza (collectively, "Third-Party Software"). Wowza does not endorse, is not responsible for, and accepts no liability related to Third-Party Software. Please ensure that any and all use of Wowza software and third-party software is properly licensed.

Wowza Trademarks

Wowza™, Wowza Streaming Cloud™, Wowza Streaming Engine™, along with other trademarks, logos, trade dress, and other proprietary colors and markings, are each trademarks or registered trademarks of Wowza in the United States and in other countries (collectively, "Wowza Marks"). No right to use Wowza Marks in any way is granted hereunder. Contact sales@wowza.com for information about obtaining the right to use Wowza Marks. Any use of Wowza Marks, authorized or otherwise, shall inure to the sole benefit of Wowza.

Third-Party Trademarks and Copyrights

Trademarks, product names, logos, designs, trade dress, and other proprietary markings of non-Wowza third parties (collectively, "Third-Party Marks") may be trademarks or registered trademarks of their respective owners. Use of Third-Party Marks is for the sole purpose of identifying third-party products and services and does not represent endorsement, sponsorship, partnership, or other affiliation between Wowza and such third parties.

A list of applicable patent and copyright notices related to content in this document is available on the Wowza website at www.wowza.com/legal.

Document History

Version	Description	Date
Doc v4.0.0	Document release for Wowza Streaming Engine 4.0.0	09-22-2014
Doc v4.4.0	Updated with SSH (HTTPS) and Microsoft Azure source support in Wowza Streaming Engine 4.4.0	02-02-2016
Doc v4.5.0	Updated with Google Cloud Storage source support in Wowza Streaming Engine 4.5.0	06-22-2016
Doc v4.7.7.1	Updated for new Java requirements.	02-01-2019

Table of Contents

Chapter

1

Introduction

Media Cache is a read-through caching mechanism for video-on-demand (VOD) streaming. Media Cache lets you upload and manage content through a centralized point, allowing you to scale without having additional hardware, implement complex content management, tax bandwidth and network resources, or increase latency.

Media Cache is very flexible and extendable. It can retrieve content from HTTP-based servers that support HTTP/1.1 range requests, from network-attached file systems, and from cloud data storage services. Media Cache can also be used with edge server Wowza nDVR workflows to cache nDVR content from an origin server application that's running Wowza nDVR.

Media Cache is built in to Wowza Streaming Engine™ media server software and can be configured through Wowza Streaming Engine Manager. (Previously, it was available separately through the MediaCache AddOn for Wowza Media Server™ software releases.)

How Media Cache works

With a regular cache, the content is cached only once, after it has been completely downloaded. This is very inefficient for large on demand files—and, in fact, most on demand sources would be too large for a regular cache to handle anyway.

Instead, Media Cache only caches content that has actually been requested by a client. If someone only views a short portion of the video, Media Cache caches only that portion. When another client accesses the same video, Media Cache delivers the cached content first. When the cached content has been played, Media Cache starts caching and delivering the next portion of the content.

When content hasn't been watched for a specified period of time, Media Cache empties the cache to make room for new content.

Chapter

2

Installation and enablement

Media Cache installs with Wowza Streaming Engine and is available through the browser-based Wowza Streaming Engine Manager.

Wowza Streaming Engine automatically enables Media Cache when it starts. Leaving Media Cache enabled has no adverse effects on performance. You should leave it enabled unless you have a specific reason to turn it off.

Open Wowza Streaming Engine Manager

Wowza Streaming Engine and Wowza Streaming Engine Manager must be running before you can open and perform tasks in the manager. If you need help starting the software, see Start and stop Wowza Streaming Engine.

You can use Wowza Streaming Engine Manager with the latest versions of most modern web browsers that support HTML5 and Cascading Style Sheets level 3 (CSS 3). For best results, we recommend that you use the latest version of the Google Chrome browser.

1. Navigate to the following URL:

   ```
   http://[wowza-ip-address]:8088/enginemanager
   ```

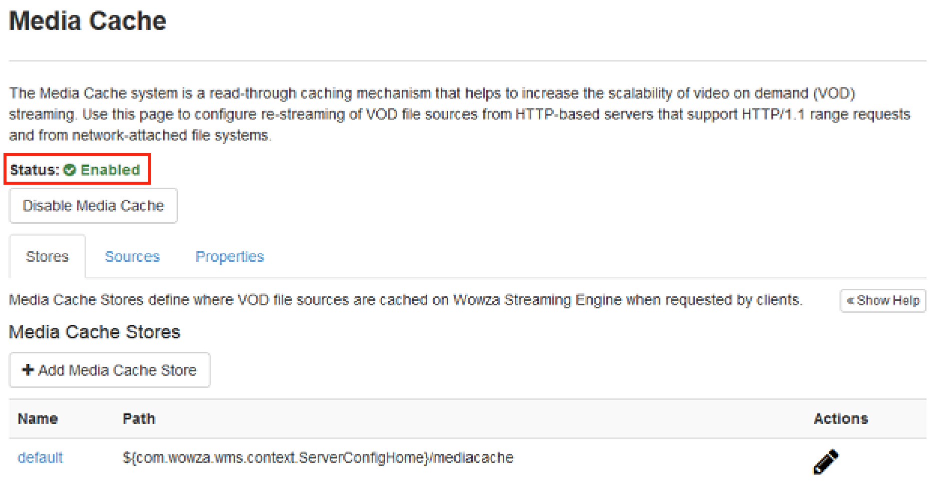
   Where **[wowza-ip-address]** is the IP address or domain name of the local or remote server running Wowza Streaming Engine.

2. Sign in to Wowza Streaming Engine Manager with the Administrator account you created when you installed the Wowza Streaming Engine.

Access Media Cache

Click the **Server** tab at the top of the manager, and then select **Media Cache** in contents panel.

The **Status** field indicates whether Media Cache is enabled or not. By default, it's enabled.



Media Cache

The Media Cache system is a read-through caching mechanism that helps to increase the scalability of video on demand (VOD) streaming. Use this page to configure re-streaming of VOD file sources from HTTP-based servers that support HTTP/1.1 range requests and from network-attached file systems.

Status: ✓ Enabled

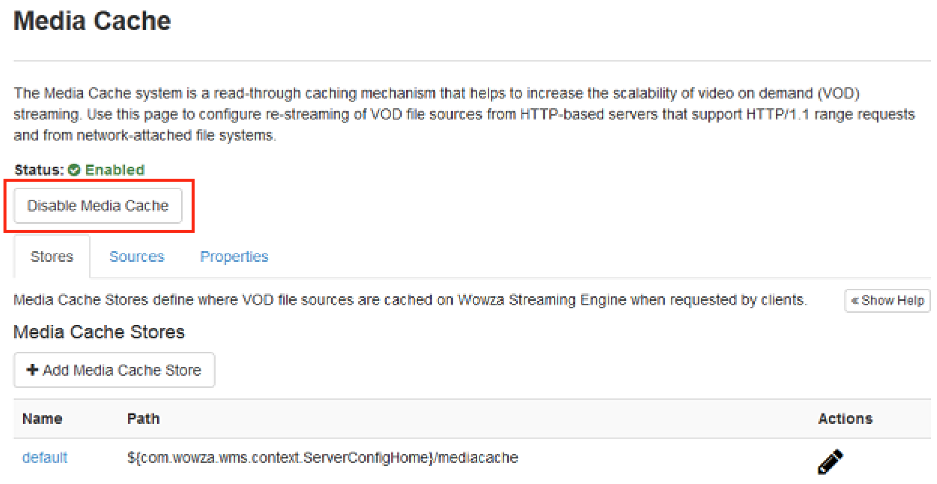
Media Cache Stores define where VOD file sources are cached on Wowza Streaming Engine when requested by clients.

Media Cache Stores

Name	Path	Actions
default	${com.wowza.wms.context.ServerConfigHome}/mediacache	✎

Disable Media Cache

1. Click **Disable Media Cache**.



Media Cache

The Media Cache system is a read-through caching mechanism that helps to increase the scalability of video on demand (VOD) streaming. Use this page to configure re-streaming of VOD file sources from HTTP-based servers that support HTTP/1.1 range requests and from network-attached file systems.

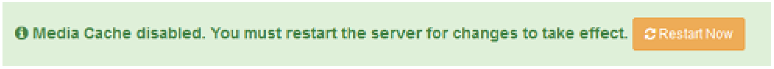
Status: ✓ Enabled

Media Cache Stores define where VOD file sources are cached on Wowza Streaming Engine when requested by clients.

Media Cache Stores

Name	Path	Actions
default	${com.wowza.wms.context.ServerConfigHome}/mediacache	✎

A disabled notice appears and prompts you to restart Wowza Streaming Engine.



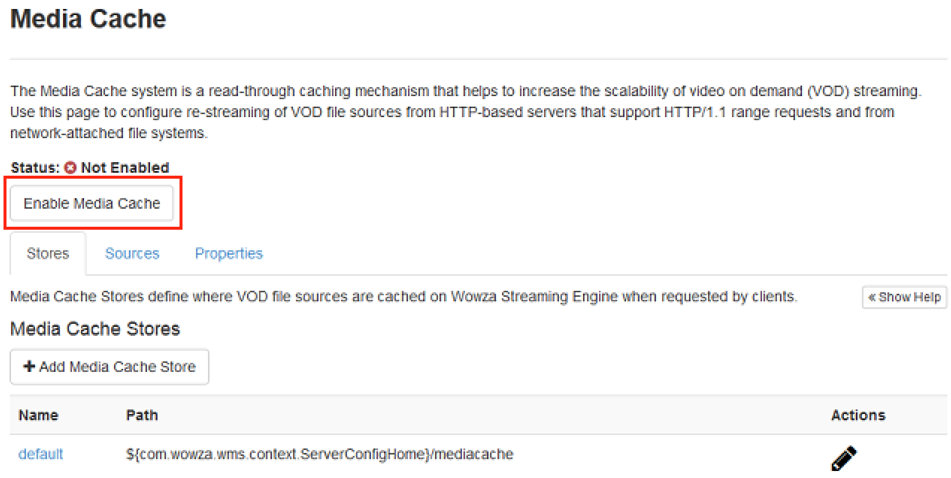
Media Cache disabled. You must restart the server for changes to take effect.

2. Click **Restart Now**.
3. In the **Restart Server** confirmation window, click **OK**.

> **Note**
>
> Wowza Streaming Engine Manager displays a "server will restart" message that persists even after the restart completes. To dismiss it, refresh your browser after a few seconds.

Enable Media Cache

1. Click **Enable Media Cache**.



Media Cache

The Media Cache system is a read-through caching mechanism that helps to increase the scalability of video on demand (VOD) streaming. Use this page to configure re-streaming of VOD file sources from HTTP-based servers that support HTTP/1.1 range requests and from network-attached file systems.

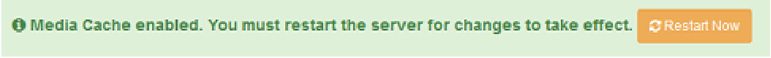
Status: ⊗ Not Enabled

Media Cache Stores define where VOD file sources are cached on Wowza Streaming Engine when requested by clients.

Media Cache Stores

Name	Path	Actions
default	${com.wowza.wms.context.ServerConfigHome}/mediacache	✎

An enabled notice appears and prompts you to restart the server.



Media Cache enabled. You must restart the server for changes to take effect.

2. Click **Restart Now**.
3. In the **Restart Server** confirmation window, click **OK**.

> **Note**
>
> Wowza Streaming Engine Manager displays a "server will restart" message that persists even after the restart completes. To dismiss it, refresh your browser after a few seconds.

Chapter

3

Basic configuration

For most production environments, some basic configuration is required. This involves specifying settings for Media Cache *stores* and *sources*.

Stores define where on demand file sources are cached by Wowza Streaming Engine when requested by clients.

Sources are the locations from which Wowza Streaming Engine retrieves the content to be cached.

Note

You can fine-tune Media Cache, store, and source performance by editing and adding properties. Editing and adding properties is covered in Chapter 4, "Advanced configuration."

Configure Media Cache stores

Media Cache stores define where Wowza Streaming Engine caches files. You can configure multiple stores to cache content on multiple volumes on a single computer. You can also use RAID storage devices and create a single store from multiples disks. Cached files are stored in a two-tier directory structure in the location that you specify.

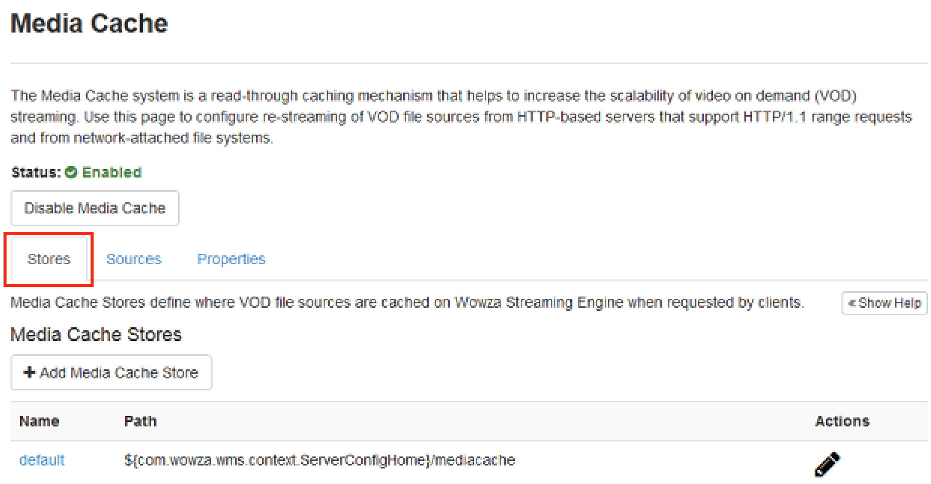
The default store location is **[*install-dir*]/mediacache**. This folder's default settings are adequate for testing and for low utilization; a production environment will probably require more space and resources.

We recommend that you do *not* change the default store unless you are an advanced user and have a specific need to do so. You can, however, create, edit, and delete other stores.

View Media Cache stores

1. If necessary, click the **Server** tab in the menu bar, and then select **Media Cache** in the contents panel.

   The **Stores** tab displays a list of available stores.



Media Cache

The Media Cache system is a read-through caching mechanism that helps to increase the scalability of video on demand (VOD) streaming. Use this page to configure re-streaming of VOD file sources from HTTP-based servers that support HTTP/1.1 range requests and from network-attached file systems.

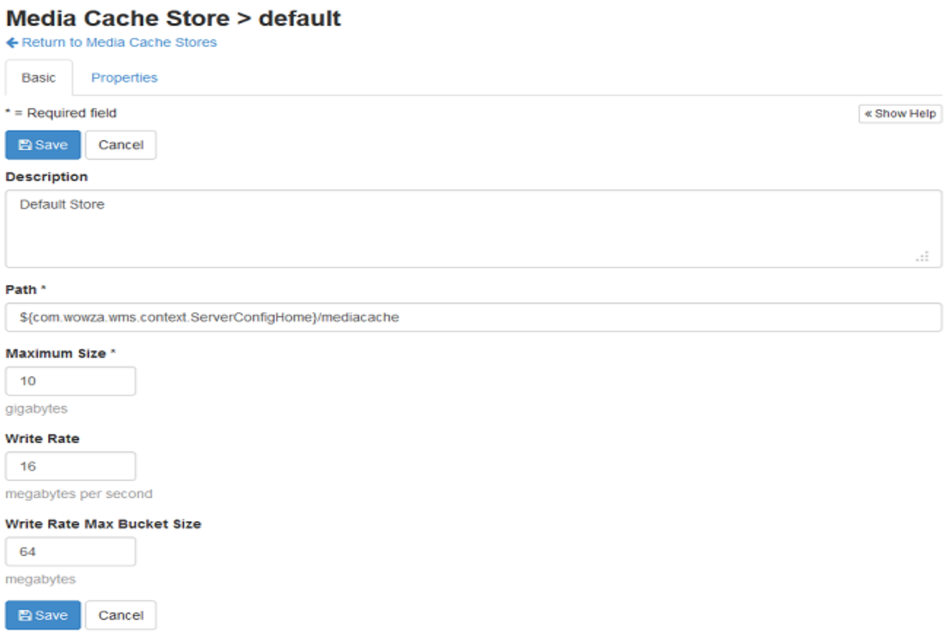
Status: ✓ Enabled

Media Cache Stores define where VOD file sources are cached on Wowza Streaming Engine when requested by clients.

Media Cache Stores

Name	Path	Actions
default	${com.wowza.wms.context.ServerConfigHome}/mediacache	✎

2. In the **Media Cache Stores** list, click a store **Name**, such as **default**, to see its **Description**, **Path**, **Maximum Size**, **Write Rate**, and **Write Rate Max Bucket Size**.



Media Cache Store > default

* = Required field

Description
Default Store

Path *
${com.wowza.wms.context.ServerConfigHome}/mediacache

Maximum Size *
10
gigabytes

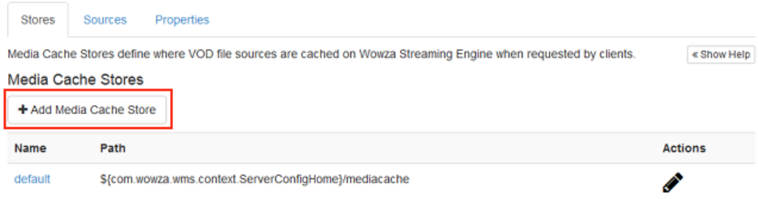
Write Rate
16
megabytes per second

Write Rate Max Bucket Size
64
megabytes

Add a Media Cache store

If you want to use multiple disks for cache storage, you must add a store for each location.

1. If necessary, click the **Server** tab in the menu bar, and then select **Media Cache** in the contents panel.
2. On the **Stores** tab of the Media Cache page, click **Add Media Cache Store**.



Media Cache Stores define where VOD file sources are cached on Wowza Streaming Engine when requested by clients.

Media Cache Stores

Name	Path	Actions
default	${com.wowza.wms.context.ServerConfigHome}/mediacache	✎

3. On the **Basic** tab, enter the following values:

   - **Store Name** and **Description** – Information about this store to help you identify it from other Media Cache stores. The **Store Name**, which is required, must be unique and can't contain less-than (<), greater-than (>), colon (:), quotation (' and "), forward slash (/), backslash (\), pipe (|), question mark (?), asterisk (*), double-dot (..), or tilde (~) characters.

     The **Description** is optional.

   - **Path** – The location where files are cached. You can enter the full path to this location or use variables in place of path elements. For example, **${com.wowza.wms.context. ServerConfigHome}/mediacache** defines a cache store in the server configuration location**.**

     The path value must be unique across all Media Cache stores and can't contain less-than (<), greater-than (>), quotation (' and "), backslash (\), pipe (|), question mark (?), and asterisk (*) characters. Always use the forward slash (/) character when defining paths.

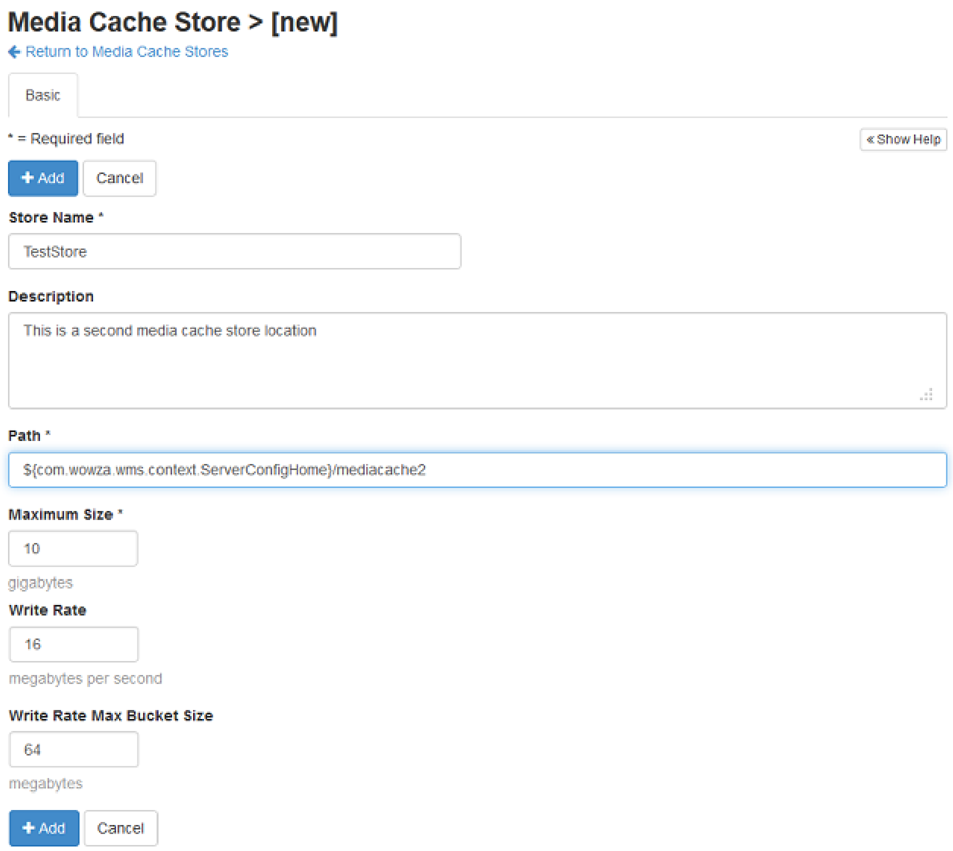
     If the folder you define doesn't exist and permissions allow it, Wowza Streaming Engine will create the folder.

   - **Maximum Size** – The maximum amount of data that can be stored in this cache store, in gigabytes.

   - **Write Rate** – The maximum rate that content can be written to the store. Throttling the write rate helps to control the flow of content entering the cache so that the file system isn't overwhelmed when content is also served from the cache (the read rate). The default value (**16 MB per second**) is tuned for a mechanical disk. For solid-state drives, **38 MB per second** is recommended.

> **Note**
>
> When adjusting the write rate, leave adequate network throughput capacity for read operations. Wowza recommends using a value that's 25 percent of the sustained write throughput rate of the storage device.

- **Write Rate Max Bucket Size** – The maximum amount of content that is cached before the write rate is throttled. This value should be set to 4 to 6 times the **Write Rate** value.



Media Cache Store > [new]

Store Name *: TestStore

Description: This is a second media cache store location

Path *: ${com.wowza.wms.context.ServerConfigHome}/mediacache2

Maximum Size *: 10 gigabytes

Write Rate: 16 megabytes per second

Write Rate Max Bucket Size: 64 megabytes

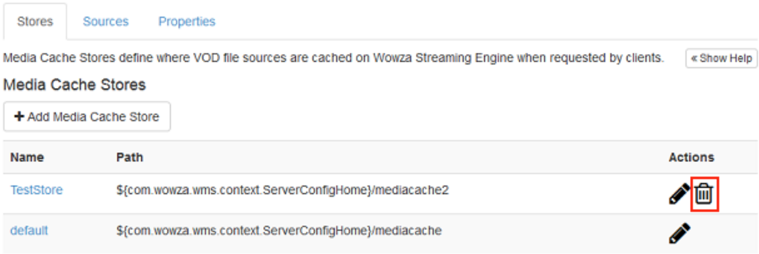
4. Click **Add**.

Edit a Media Cache store

1. On the **Stores** tab of the Media Cache page, do one of the following:
   - Click **Edit** (the pencil icon) for the store you want to modify.
   - Click a store name in the list, and then click **Edit**.
2. On the **Basic** tab, change the **Name**, **Description**, **Path**, **Maximum Size**, **Write Rate**, and/or the **Write Rate Max Bucket Size.**
3. Click **Save**.

Delete a Media Cache store

1. On the **Stores** tab of the **Media Cache** page, click **Delete** (the trash can icon) for the store you want to remove.



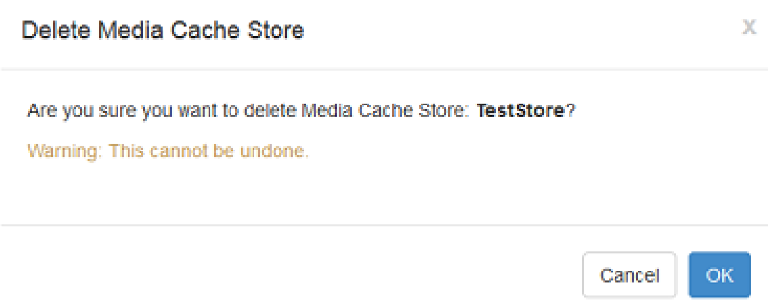
Media Cache Stores define where VOD file sources are cached on Wowza Streaming Engine when requested by clients.

Media Cache Stores

Name	Path	Actions
TestStore	${com.wowza.wms.context.ServerConfigHome}/mediacache2	✎ 🗑
default	${com.wowza.wms.context.ServerConfigHome}/mediacache	✎

> **Note**
>
> The default store can't be deleted.

2. In the **Delete Media Cache Store** confirmation window, click **OK**.

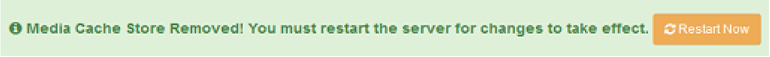


Are you sure you want to delete Media Cache Store: **TestStore**?

Warning: This cannot be undone.

Although the store has been deleted, it remains in use until the server is restarted.

3. Click **Restart Now**.



Media Cache Store Removed! You must restart the server for changes to take effect.

Configure Media Cache sources

Media Cache sources are the locations where Wowza Streaming Engine can find the incoming content to cache in stores and to use in VOD edge applications. Valid sources are:

- HTTP-based servers that support HTTP/1.1 range requests.
- Network-attached file systems, including network file systems and any device that's recognized as a disk by the operating system.
- The Amazon S3 cloud storage web service.
- The Microsoft Azure cloud computing platform.

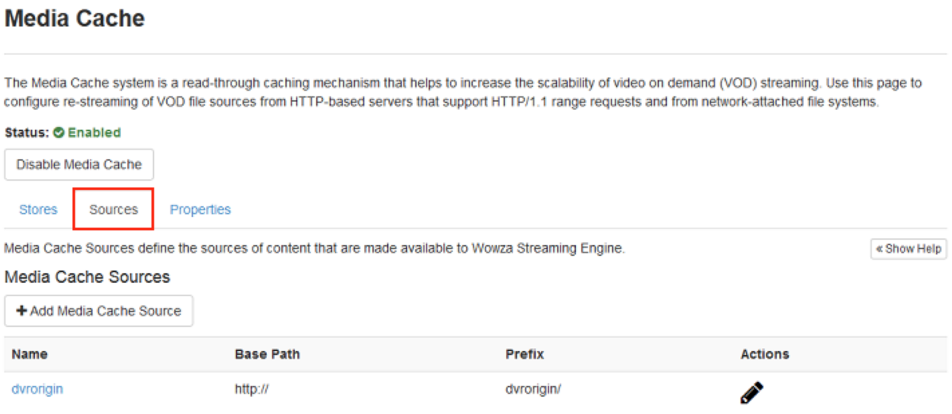
- The Google Cloud Storage web service.

The default Media Cache source is called **dvorigin**. The **dvrorigin** source is an HTTP source configured to work with a Wowza Streaming Engine origin server that has Wowza nDVR enabled. This enables DVR chunks to be re-streamed through the Media Cache store, increasing the scalability of Wowza nDVR streams. The **dvorigin** source can't be removed. You can, however, create, edit, and delete other sources.

View Media Cache sources

1. If necessary, click the **Server** tab in the menu bar, and then select **Media Cache** in the contents panel.
2. Click the **Sources** tab.

   The **Sources** tab displays a list of available sources.



Media Cache

The Media Cache system is a read-through caching mechanism that helps to increase the scalability of video on demand (VOD) streaming. Use this page to configure re-streaming of VOD file sources from HTTP-based servers that support HTTP/1.1 range requests and from network-attached file systems.

Status: ✓ Enabled

Media Cache Sources define the sources of content that are made available to Wowza Streaming Engine.

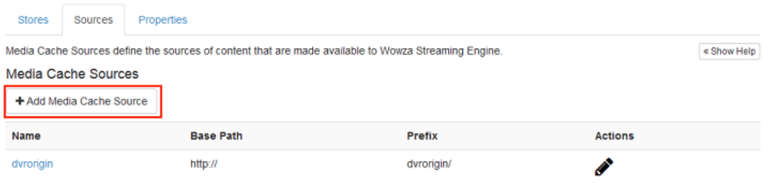
Media Cache Sources

Name	Base Path	Prefix	Actions
dvrorigin	http://	dvrorigin/	✎

3. In the **Media Cache Sources** list, click a source **Name**, such as the default **dvorigin**, to see its **Description**, **Prefix**, **Minimum Time To Live**, and **Maximum Time To Live**.

Add a Media Cache source

1. If necessary, click the **Server** tab in the menu bar, and then select **Media Cache** in the contents panel.
2. On the **Sources** tab of the **Media Cache** page, click **Add Media Cache Source**.



Media Cache Sources define the sources of content that are made available to Wowza Streaming Engine.

Media Cache Sources

Name	Base Path	Prefix	Actions
dvrorigin	http://	dvrorigin/	✎

3. On the **Basic** tab, enter the following values:

   - **Source Name** and **Description** –Information about this source to help you identify it from other Media Cache sources. The **Source Name**, which is required, must be unique and can't contain less-than (<), greater-than (>), colon (:), quotation (' and "), forward slash (/), backslash (\), pipe (|), question mark (?), asterisk (*), double-dot (..), or tilde (~) characters.

     The **Description** is optional.

   - **Source Type** – The type of source. Select one of the following:

     - **File** – Retrieves content from standard file reads. Use to re-stream content from disk sources such as a single large disk, network attached storage (NAS), or NFS storage locations.
     - **HTTP** – Uses HTTP/1.1 requests to retrieve content. Use to re-stream files from HTTP-based web servers. No authentication is performed for **HTTP** sources. Both HTTP and HTTPS (SSL) requests are supported.
     - **AmazonS3** – Uses HTTP/1.1 requests to retrieve content. Use to re-stream content from web servers that use S3-style authentication.
     - **Azure** – Retrieves content from a Microsoft Azure blob storage account. Use to manage your content outside of Wowza Streaming Engine.
     - **GoogleCloudStorage** – Retrieves content from a Google Cloud Storage bucket connected to a Google service account. Use to manage your content outside of Wowza Streaming Engine.

     ### Notes

     - You can't edit the **Source Type** after the source is created.
     - Microsoft Azure sources and HTTPS (SSL) requests are supported in Wowza Streaming Engine 4.4.0 and later. Google Cloud Storage Bucket sources are supported in Wowza Streaming Engine 4.5.0 and later.

   - **Prefix** – The first part of the URL that clients use to access content from the source. The **Prefix** field populates automatically depending on the selected **Source Type**:

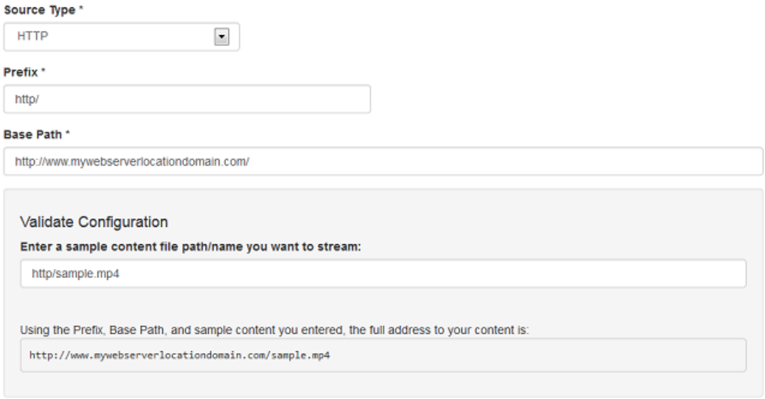
     - **File** uses the **path1/** prefix.
     - **HTTP** uses the **http/** prefix.
     - **AmazonS3** uses the **amazon3/** prefix.
     - **Azure** uses the **http/** prefix.
     - **GoogleCloudStorage** uses the **http/** prefix.

- **Base Path** – Forms the second part of the URL that clients use to access content from the source.
    - A **File** source base path is the file system location where the content can be found.
    - An **HTTP** source base path is the HTTP or HTTPS URL where the content can be found.
    - An **AmazonS3** source base path is **http://s3.amazonaws.com/**.
    - **Azure** and **GoogleCloudStorage** sources don't use a base path.

  The **Base Path** can't contain less-than (<), greater-than (>), quotation (' and "), backslash (\), pipe (|), question mark (?), or asterisk (*) characters.

- **Validate Configuration** – The location that Media Cache uses to find the source. This field updates automatically as you specify the **Prefix** and **Base Path**. Make sure the **Validate Configuration** field shows the full, correct address of your content.

  The following screenshot shows sample **Prefix**, **Base Path**, and **Validate Configuration** values for an HTTP source:



Source Type *
HTTP

Prefix *
http/

Base Path *
http://www.mywebserverlocationdomain.com/

Validate Configuration

Enter a sample content file path/name you want to stream:
http/sample.mp4

Using the Prefix, Base Path, and sample content you entered, the full address to your content is:
http://www.mywebserverlocationdomain.com/sample.mp4

Note

Microsoft Azure and Google Cloud Storage sources don't include the **Validate Configuration** field.

- **Minimum Time to Live** and **Maximum Time To Live** – Options that control how long unused content is cached. The time-to-live counter starts when the last client playing the content stops. If the cache is full and other source items are requested, content that isn't viewed within the specified **Minimum Time To Live** is purged from the cache. If the cache isn't full, content that isn't viewed remains in the cache until the specified **Maximum Time To Live** is reached, after which it's purged from the cache. The **Minimum Time To Live** value must be less than or equal to the **Maximum Time To Live** value.

4. For Amazon S3 sources, specify the following additional settings:
    - **AWS Access Key ID** and **AWS Secret Access Key** – Enable re-streaming from an Amazon S3 bucket that's not publicly available. Enter your Amazon S3 key ID and secret access key to enable Media Cache to sign HTTP requests so that they're authenticated for access to protected content. These values can't have spaces.
    - **Use Amazon S3 bucket name in domain** – Controls how the HTTP request URLs that Media Cache uses on Amazon S3 are formatted. Select this option to include the bucket name in the domain name. Otherwise, the bucket name is included in the URL path.
5. For Microsoft Azure sources, specify the following additional settings:
    - **Blob Storage Account Name** – Your Azure Blob account name. This value can't contain spaces.
    - **Blob Storage Container Name** – The name of a container in your Azure Blob storage. A Media Cache Source can point to only one container in Azure Blob storage.
    - **Blob Storage Account Access Key** – The primary or secondary access key for the Azure Blob account.
6. For Google Cloud Storage sources, specify the following additional settings:
    - **Google Service Account Email Address** – The email address (or account ID) used with your service account.
    - **Google Private Key Type** – The type of private key file certificate that you selected and downloaded when creating your service account (**JSON** or **P12**).
        i. **JSON:** If you're using a **JSON** private key file, open the file in a text editor and then copy and paste the file contents into the **Google JSON Private Key** box.

        ii. **P12:** If you're using a **P12** private key file, enter the file name (including the file extension) in **Google P12 Private Key File Name** and enter the password to access the private key in **Google P12 Private Key Password**.
7. Click **Add**.

Edit a Media Cache source

1. On the **Sources** tab of the **Media Cache** page, do one of the following:
    - Click **Edit** (the pencil icon) for the source you want to modify.
    - Click a source name in the list, and then click **Edit**.
2. On the **Basic** tab, change the **Description**, **Prefix**, **Base Path**, **Minimum Time To Live**, **Maximum Time To Live**, and/or the **AWS Options.**

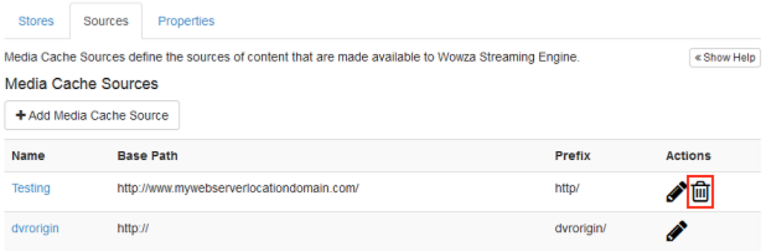
Note

You can't edit the **Source Type**.

3. Click **Save**.

Delete a Media Cache source

1. On the **Sources** tab of the **Media Cache** page, click **Delete** (the trash can icon) for the source you want to remove.



Media Cache Sources define the sources of content that are made available to Wowza Streaming Engine.

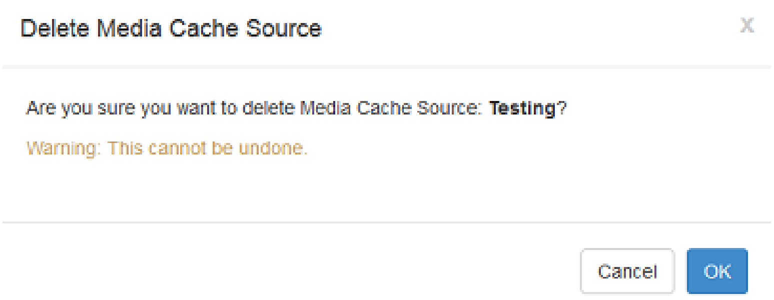
Media Cache Sources

Name	Base Path	Prefix	Actions
Testing	http://www.mywebserverlocationdomain.com/	http/	✎ 🗑
dvrorigin	http://	dvrorigin/	✎

Note

The default **dvorigin** source can't be deleted.

2. In the **Delete Media Cache Source** confirmation window, click **OK**.



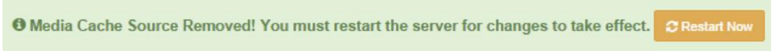
Delete Media Cache Source

Are you sure you want to delete Media Cache Source: **Testing**?

Warning: This cannot be undone.

Although the source has been deleted, it remains in use until the server is restarted.

3. Click **Restart Now**.



Media Cache Source Removed! You must restart the server for changes to take effect.

Chapter 4

Advanced configuration

Administrators can fine-tune how Media Cache processes and manages streams, stores, and sources using Media Cache properties.

Access to the **Properties** tabs requires an administrative user with advanced permissions. See Manage credentials in the article "Get started as a Wowza Streaming Engine Manager administrator" for instructions on setting up advanced permissions.

Important

Only users with advanced Media Cache knowledge should change properties. The primary reason to edit properties is to improve performance.

Configure Media Cache properties

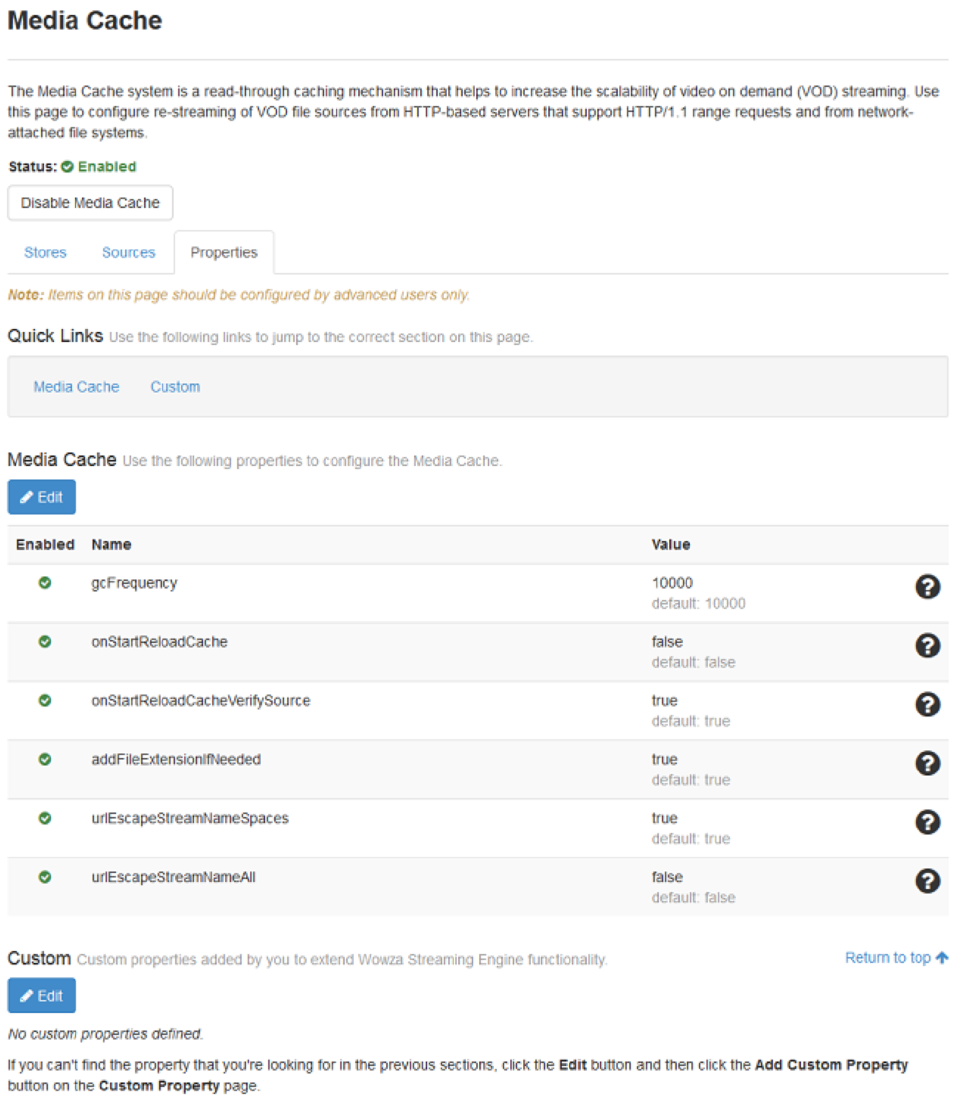
You can fine-tune overall Media Cache performance by editing Media Cache properties.

Note

Media Cache properties can also be edited directly in the <**MediaCache>** section of the **MediaCache.xml** file located at **[install-dir]/conf/**.

1. If necessary, click the **Server** tab in the menu bar, and then select **Media Cache** in the contents panel.
2. Click the **Properties** tab.

The **Properties** tab displays a list of available properties.



Media Cache

The Media Cache system is a read-through caching mechanism that helps to increase the scalability of video on demand (VOD) streaming. Use this page to configure re-streaming of VOD file sources from HTTP-based servers that support HTTP/1.1 range requests and from network-attached file systems.

Status: Enabled

Note: *Items on this page should be configured by advanced users only.*

Quick Links Use the following links to jump to the correct section on this page.

Media Cache Use the following properties to configure the Media Cache.

Enabled	Name	Value
✓	gcFrequency	10000 default: 10000
✓	onStartReloadCache	false default: false
✓	onStartReloadCacheVerifySource	true default: true
✓	addFileExtensionIfNeeded	true default: true
✓	urlEscapeStreamNameSpaces	true default: true
✓	urlEscapeStreamNameAll	false default: false

Custom Custom properties added by you to extend Wowza Streaming Engine functionality.

No custom properties defined.

If you can't find the property that you're looking for in the previous sections, click the **Edit** button and then click the **Add Custom Property** button on the **Custom Property** page.

The following Media Cache properties are enabled by default:

Property	Description
gcFrequency	The amount of time (in milliseconds) between cache-purging/cache-pruning sessions. Based on time-to-live settings, items stored in the cache are purged when they haven't been used in a given period of time (maximum time-to-live)—or if there's content waiting to enter the cache (minimum time-to-live). The default value is **10000** (10 seconds).

onStartReloadCache	This property determines whether Media Cache reloads the cache when Wowza Streaming Engine starts. The value **true** specifies that the cache is reloaded. The value **false** specifies that the cache is cleared. The default value is **false.**
onStartReloadCacheVerifySource	This property is triggered when **onStartReloadCache** is **true**. Given that condition, it specifies whether Media Cache verifies each reloaded item with the source. The value **true** specifies that the reloaded content is verified. The default value is **true.**
addFileExtensionIfNeeded	This property determines whether a filename extension is added. If **addFileExtensionIfNeeded** is **true** and if the stream name doesn't include a filename extension, then a filename extension is added based on the stream name prefix. The default value is **true**.
urlEscapeStreamNameSpaces	This property determines whether spaces in stream names are URL-escaped. If the value of this property is **true**, then spaces in stream names are URL-escaped before being sent to the source HTTP server. The default value is **true**.
urlEscapeStreamNameAll	This property determines whether the stream name is URL-escaped. If the value of this property is **true**, the entire stream name is URL-escaped before being sent to the source HTTP server. The default value is **false**.

Note

You can fine-tune additional Media Cache properties in the **Performance Tuning** area of Wowza Streaming Engine Manager. Click the **Server** tab and then, in the contents panel, select **Performance Tuning** > **Media Cache Tuning**. For more information about how to tune these properties, see Tune Media Cache in the article "Tune Wowza Streaming Engine for optimal performance."

3. Click **Edit** and:

   - Deselect **Enabled** to turn a property off.
   - Change a property's state from **True** to **False**, or enter a different **Value**.

4. Click **Save**.

Configure Media Cache store properties

You can fine-tune the performance of Media Cache stores by editing a store's properties.

Note

Media Cache store properties can also be edited directly in the <**MediaCacheStores>** section of the **MediaCache.xml** file located at **[install-dir]/conf/**.

1. If necessary, click the **Server** tab in the menu bar, and then select **Media Cache** in the contents panel.
2. In the **Stores** tab, click the name of the store whose properties you want to configure.
3. Click the **Properties** tab.

   The **Properties** tab displays a list of available store properties. The following store properties are enabled by default:

Property	Description
level1FolderCount	The number of folders created at the first level to store cached items. The default value is **24**.
level2FolderCount	The number of folders created at the second level to store cached items. The default value is **24.**
fileCount	The maximum number of files that can be stored in each level 2 folder. The default value is **1000**. This means that 576,000 files (24x24x1000) can be stored in this cache store by using default values.
writeRateFillFrequency	The interval, in milliseconds, at which the write rate control mechanism is refreshed. This value works with the **Write Rate** to control how the write rate is throttled. This value should be set to **100**, as it is by default.

5. Click **Edit** and:
   - Deselect **Enabled** to turn the property off.
   - Change a property's **Value**.
6. Click **Save**.

Configure Media Cache source properties

You can fine-tune Media Cache source performance by editing source properties.

Note

Media Cache source properties can also be edited directly in the <**MediaCacheSources>** section of the **MediaCache.xml** file located at **[install-dir]/conf/**.

1. If necessary, click the **Server** tab in the menu bar, and then select **Media Cache** in the contents panel.
2. In the **Sources** tab, click the name of the source whose properties you want to configure.
3. Click the **Properties** tab.

   The **Properties** tab displays two lists of available source properties:

   - A list of properties for the type of source you've selected—File source or HTTP source.
   - A list of properties that apply to all source types.

   No properties are available specifically for Amazon S3 sources.

File source properties

Property	Description
fileReadRetries	The number of times a file request is retried if it fails. The default value is **3**.

HTTP source properties

Property	Description
httpReadTimeout	The number of milliseconds that the server will wait for a read request to be fulfilled. The default value is **3000** (3 seconds).
maxPoolSize	The maximum number of connections allowed in the connection pool. The default value is **25**. Media Cache creates a pool of HTTP TCP connections per-source. Requests are sent to the source using this HTTP TCP pool.
httpConnectionTimeout	The number of milliseconds that the server will wait for an SSL connection. The default value is **6000** (6 seconds).

httpReadRetries	The number of times an HTTP request is retried if it fails. The default value is **3**.
httpSendBufferSize	The size, in bytes, of the send buffer, which is a low-level TCP buffer. The default value is **8000**.
httpReceiveBufferSize	The size, in bytes, of the receive buffer, which is a low-level TCP buffer. The default value is **65000**.
closeConnectOnBadContentLength	Switch that controls the socket connection when Media Cache makes a range request to the HTTP source and gets an incorrect content length in the response. If set to **true**, the server closes the connection when an unexpected content length is received. The default value is **false**.
proxyHost, proxyPort, requestFullURL	Properties that direct HTTP requests to a proxy. If these properties are set, Media Cache sends all HTTP requests to the defined proxy.

Media Cache Source properties

defaultBlockSize	The size of the blocks of data that are read from the source to populate the cache. Values are expressed in bytes and the following units are supported: K (kilobytes), M (megabytes), G (gigabytes), and T (terabytes). The default value is **1M**.
readAhead	Switch that controls whether content blocks are cached. If set to **true**, content blocks are pre-populated in the cache based on the **readAheadThreshold**. If set to **false**, no read ahead occurs. The default state is **true**.
readAheadThreshold	A percentage value that controls when the request is made to read the next block of data. For example, if set to **50,** then when any content is read beyond the halfway mark of the current block of content, the next block is queued to be read. The default value is **30**.

4. Click **Edit** and:
   - Deselect **Enabled** to turn the property off.
   - Change a property's state from **True** to **False**, or enter a different **Value**.
5. Click **Save**.

Add custom Media Cache HTTP source properties

You can also fine-tune Media Cache HTTP source performance by adding custom HTTP source properties.

Note

Media Cache source properties can also be added directly to the **<MediaCacheSources>/<Media CacheSource>/<Properties>** section of the **MediaCache.xml** file located at **[install-dir]/conf/**.

1. If necessary, click the **Server** tab in the menu bar, and then select **Media Cache** in the contents panel.
2. In the **Sources** tab, click the name of the source to which you want to add custom properties.
3. Click the **Properties** tab.
4. In the Quick Links section, click **Custom**.
5. Click **Edit**, and then click **Add Custom Property**.
6. Add any of the following custom properties:

- **closeConnectOnBadStatus** – Controls the socket connection when Wowza Streaming Engine makes a range request to the HTTP source but doesn't get the HTTP status code 206 (Partial Content) in the response. Any other status value that's returned is interpreted as an error, and if the property value is set to **true**, the server will close the connection. The default value is **false.**

Path	Name	Type	Value
/Root/MediaCacheSources/ MediaCacheSource	**closeConnectOnBadStatus**	Boolean	**True**

- **sendHTTPKeepAlive** – Controls whether Wowza Streaming Engine makes a keep-alive connection to an HTTP source. The source must support keep-alive connections. The default value (**true**) enables a keep-alive connection.

Path	Name	Type	Value
/Root/MediaCacheSources/ MediaCacheSource	**sendHTTPKeepAlive**	Boolean	**True**

- **sendHTTPProxyKeepAlive** – Controls whether Wowza Streaming Engine makes a keep-alive connection to an HTTP proxy. The proxy must support keep-alive connections. The default value (**true**) enables a keep-alive connection.

Path	Name	Type	Value
/Root/MediaCacheSources/ MediaCacheSource	**sendHTTPProxyKeepAlive**	Boolean	**true**

- **userHTTPHeaders** – Enables custom HTTP headers to be added to HTTP source requests. The **Value** is a pipe-separated list of key: value pairs that define the HTTP headers to add to requests to the source. For example: `User-Agent: Mozilla/5.0|Access-Control-Request-Method: <method>.`

Path	Name	Type	Value
/Root/MediaCacheSources/ MediaCacheSource	**userHTTPHeaders**	String	key1: value1|key 2: value2

- **debugHTTPRequests** – Enables all the HTTP requests made to the source to be shown and shows how the source responded. This allows problematic HTTP sources to be debugged and causes of content failures to be seen.

Path	Name	Type	Value
/Root/MediaCacheSources/ MediaCacheSource	**debugHTTPRequests**	Boolean	**false**

- **urlDecodeStreamName** – Enables content names with non-safe transportable characters such as spaces and dots to be URL encoded.

Path	Name	Type	Value
/Root/MediaCacheSources/ MediaCacheSource	**urlDecodeStreamName**	Boolean	**false**

- **base64DecodeStreamName** –Allows all content names to be base64 encoded, if set to **true**. This can aid in the transport of both unsafe and HTTP get variables to be passed through to the HTTP source. Note that passing HTTP get variables means the same piece of content could be cached more than once if the variable changes for new requests.

Path	Name	Type	Value
/Root/MediaCacheSources/ MediaCacheSource	**base64DecodeStreamName**	Boolean	**false**

Note

If your source is Amazon S3, the **isAmazonS3** property appears in the **Custom** list of source properties. The property, however, can't be edited.

7. Click **Add**, click **Save**, and then restart Wowza Streaming Engine.

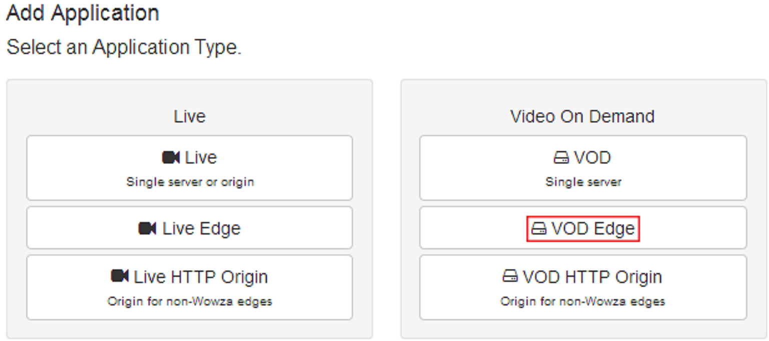
Chapter 5

Configure an application to use Media Cache

With Media Cache configured and enabled, you can use it in an application. Media Cache must use the **VOD Edge** application type.

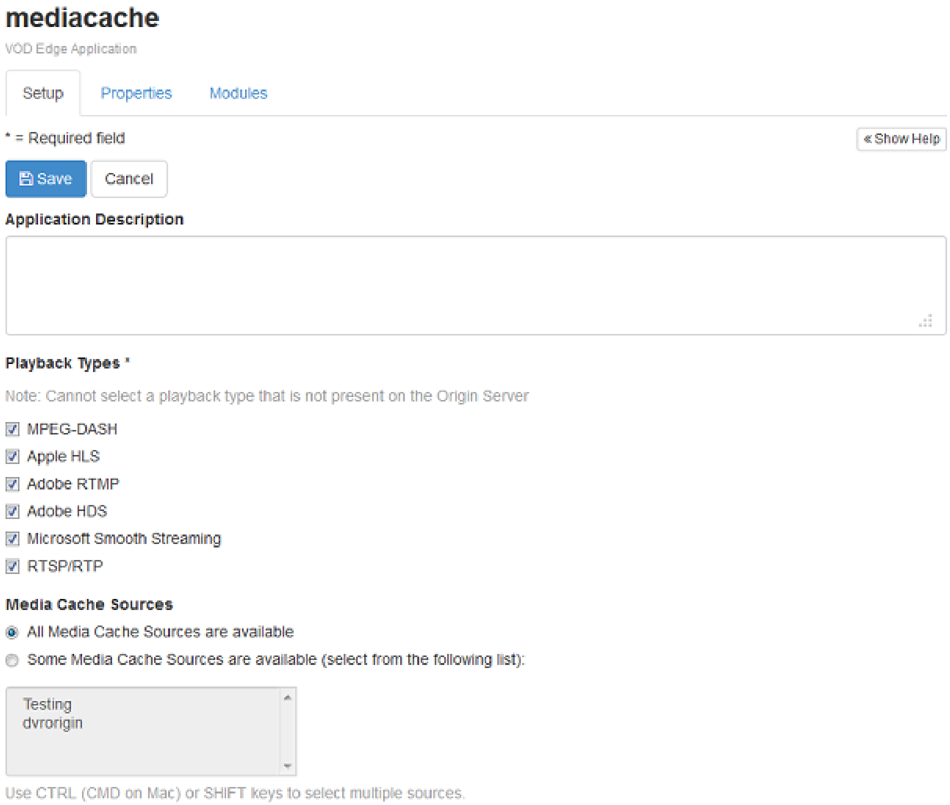
Create a VOD edge application that uses Media Cache

1. In Wowza Streaming Engine Manager, click the **Applications** tab, and then click **Add Application** in the contents panel.
2. On the **Add Application** page, click **VOD Edge**.



3. In the **New Application** dialog box, enter a name for the application (such as **mediacache**), and then click **Add**.

   The VOD edge application displays a **Setup** tab.



mediacache

VOD Edge Application

* = Required field

Application Description

Playback Types *

Note: Cannot select a playback type that is not present on the Origin Server

☑ MPEG-DASH
☑ Apple HLS
☑ Adobe RTMP
☑ Adobe HDS
☑ Microsoft Smooth Streaming
☑ RTSP/RTP

Media Cache Sources

● All Media Cache Sources are available
○ Some Media Cache Sources are available (select from the following list):

Testing
dvrorigin

Use CTRL (CMD on Mac) or SHIFT keys to select multiple sources.

4. Under **Playback Types**, select the output formats you want to stream to. By default, all supported playback types are selected.

   For more information about playback types, see Understanding protocols and output file formats supported by Wowza Streaming Engine.

5. Under **Media Cache Sources**, choose one of the following:

   - **All Media Cache Sources are available** – Makes all sources available to the application.
   - **Some Media Cache Sources are available** – Lets you select the source or sources you want to make available to the application.

6. Click **Save**, and then restart Wowza Streaming Engine.

Chapter

6

Playback URL examples

This chapter provides sample URLs for playing Media Cache content from a file source, an HTTP source, and an Amazon S3 source.

Playback using a file source

The following URLs show how to play content from a file source by using an example application named **mediacache** and an example file named **sample.mp4** in the source location.

Adobe Flash Player (RTMP)

```
Server: rtmp://[wowza-ip-address]/mediacache
Stream: mp4:path1/sample.mp4
```

Adobe Flash player (San Jose/Flash HTTP)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:path1/sample.mp4/manifest.f4m
```

Apple iOS device (Cupertino/Apple HTTP Live Streaming)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:path1/sample.mp4/playlist.m3u8
```

Microsoft Silverlight (Smooth Streaming)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:path1/sample.mp4/Manifest
```

MPEG-DASH

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:path1/sample.mp4/manifest.mpd
```

RTSP/RTP player or device

```
rtsp://[wowza-ip-address]:1935/mediacache/_definst_/mp4:path1/sample.mp4
```

Playback using an HTTP source

The following URLs show how to play content from an HTTP source by using an example application named **mediacache** and an example file named **sample.mp4** on a remote server.

Note

If using HTTPS (SSH) requests, you can substitute **https://** for **http://** in the URL examples below.

Adobe Flash Player (RTMP)

```
Server: rtmp://[wowza-ip-address]/mediacache
Stream: mp4:http/sample.mp4
```

Adobe Flash player (San Jose/Flash HTTP)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/sample.mp4/manifest.f4m
```

Apple iOS device (Cupertino/Apple HTTP Live Streaming)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/sample.mp4/playlist.m3u8
```

Microsoft Silverlight (Smooth Streaming)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/sample.mp4/Manifest
```

MPEG-DASH

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/sample.mp4/manifest.mpd
```

RTSP/RTP player or device

```
rtsp://[wowza-ip-address]:1935/mediacache/_definst_/mp4:http/sample.mp4
```

Playback using an Amazon S3 source

The following URLs show how to play content from an Amazon S3 source by using an example application named **mediacache** and an example file named **sample.mp4** on a remote S3 server.

Adobe Flash Player (RTMP)

```
Server: rtmp://[wowza-ip-address]/mediacache
Stream: mp4:amazons3/sample.mp4
```

Adobe Flash player (San Jose/Flash HTTP)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:amazons3/sample.mp4/manifest.f4m
```

Apple iOS device (Cupertino/Apple HTTP Live Streaming)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:amazons3/sample.mp4/playlist.m3u8
```

Microsoft Silverlight (Smooth Streaming)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:amazons3/sample.mp4/Manifest
```

MPEG-DASH

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:amazons3/sample.mp4/manifest.mpd
```

RTSP/RTP player or device

```
rtsp://[wowza-ip-address]:1935/mediacache/_definst_/mp4:amazons3/sample.mp4
```

Playback using a Microsoft Azure source

The following URLs show how to play content from a Microsoft Azure source by using an example application named **mediacache** and an example file named **sample.mp4** on a remote Azure server.

Adobe Flash Player (RTMP)

```
Server: rtmp://[wowza-ip-address]/mediacache
Stream: mp4:http/sample.mp4
```

Adobe Flash player (San Jose/Flash HTTP)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/sample.mp4/manifest.f4m
```

Apple iOS device (Cupertino/Apple HTTP Live Streaming)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/sample.mp4/playlist.m3u8
```

Microsoft Silverlight (Smooth Streaming)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/sample.mp4/Manifest
```

MPEG-DASH

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/sample.mp4/manifest.mpd
```

RTSP/RTP player or device

```
rtsp://[wowza-ip-address]:1935/mediacache/_definst_/mp4:http/sample.mp4
```

Playback using a Google Cloud Storage source

The following URLs show how to play content from a Google Cloud Storage source by using an example application named **mediacache** and an example file named **sample.mp4** in a remote Google Cloud Storage bucket named **myBucket**.

Adobe Flash Player (RTMP)

```
Server: rtmp://[wowza-ip-address]/mediacache
Stream: mp4:http/myBucket/sample.mp4
```

Adobe Flash player (San Jose/Flash HTTP)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/myBucket/sample.mp4/manifest.f4
m
```

Apple iOS device (Cupertino/Apple HTTP Live Streaming)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/myBucket/sample.mp4/playlist.m3
u8
```

Microsoft Silverlight (Smooth Streaming)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/myBucket/sample.mp4/Manifest
```

MPEG-DASH

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/myBucket/sample.mp4/manifest.mp
d
```

RTSP/RTP player or device

```
rtsp://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/myBucket/sample.mp4
```

Chapter

7

Optimizing performance and troubleshooting

This chapter includes information about how to optimize performance in advanced Media Cache deployments and suggests some basic troubleshooting techniques.

Optimizing source performance

Media Cache Source properties describe how data is retrieved from a source when content is requested. Understanding how the three source properties—**defaultBlockSize**, **readAhead**, and **readAheadThreshold**—work together will help you to choose the most effective settings for your deployment.

The default values for **defaultBlockSize** (**1M)** and **readAheadThreshold** (**30**) indicate that whenever content is required, it's retrieved in 1 MB blocks, and when 30 percent of one block is read, the next block is retrieved.

To select an appropriate **defaultBlockSize** value, determine the average size of all assets in use, and then divide by 1000. For example, if the average asset size is **1G**, the **defaultBlockSize** should be **1000K** (1M). This value is four times larger than the default value and will reduce the number of requests to the source for the complete asset to one-fourth the default rate.

Note

DefaultBlockSize values greater than **4M** might not increase performance.

To select an appropriate **readAheadThreshold** value, consider the speed of the source and the bandwidth between the source and the Wowza Streaming Engine server. The default value (**30**) signifies that when 30 percent of the current block is used, the next block is retrieved. If the source or connectivity is slow, you may need to reduce the value. For example, try a value of **25**.

Don't set the **readAheadThreshold** higher than 50 percent, even with an extremely fast and well-connected source.

Optimizing local cache performance

The speed of the local cache is critical to Media Cache performance. Use a third-party tool that tests storage speed to determine the maximum write speed for the cache to use. If the store's **Write Rate** value is too high, write failures and read errors may occur and cause performance to degrade.

Troubleshooting

This section describes some common issues that you might encounter while setting up your Media Cache deployment and provides suggestions for resolution.

Wowza Streaming Engine version

When troubleshooting Media Cache issues, it's important to know the version of Wowza Streaming Engine you're using. To identify the version, click the **Server** tab, and then click **About** in the contents panel.

It's best to use the latest version of Wowza Streaming Engine to get the latest bug fixes and feature enhancements.

Media Cache store errors

If Media Cache can't write to the defined store location, Wowza Streaming Engine displays the following error after it starts:

```
Media Cache failed to start
MediaCacheImpl.start[MediaCache]: Store not added to Media Cache. Directory
not found: /mediacache/temp
MediaCacheImpl.start[MediaCache]: No stores could be added. Media Cache is
not usable.
```

To resolve this error, make sure you have permission to read and write to the store location and that you have enough disk space to hold the cache.

MediaCache.xml file missing

If Wowza Streaming Engine can't find the **MediaCache.xml** file, it displays the following error:

```
MediaCacheImpl.start[Media Cache]ERROR server comment -
MediaCacheServerListener.loadConfig: MediaCache.xml missing:
/usr/local/WowzaStreamingEngine/conf/MediaCache.xml
MediaCacheImpl.start[Media Cache]: No stores could be added. Media cache is
not usable.
```

To resolve this error, make sure that **MediaCache.xml** is located in **[install-dir]/conf/**.

Debug logging

To help you resolve issues and to gain insight into problems, enable Media Cache debug logging.

1. If necessary, click the **Server** tab in the menu bar, and then select **Server Setup** in the contents panel.
2. Click the **Properties** tab.
3. In the Quick Links section, click **Custom**.
4. Click **Edit**, and then click **Add Custom Property**.
5. Add the following custom property:

Path	**Name**	**Type**	**Value**
/Root/Server/	**MediaCacheDebugLog**	Boolean	**true**

6. Click **Add**, click **Save**, and then restart the Wowza Streaming Engine instance.

Or, add the following property to the **<Properties>** container in the **[*install-dir*]/conf/Server.xml** file:

```
<Property>
      <Name>MediaCacheDebugLog</Name>
      <Value>true</Value>
      <Type>Boolean</Type>
</Property>
```

When debug logging is enabled, all entries are written to the **wowzastreamingengine_access.log** file, including HTTP attempts, failures, when content is purged from the Media Cache store, and time taken to retrieve content. It also records warnings so that Media Cache operations can be traced in the log file.

Note

Debug logging should only be used temporarily, for diagnostic purposes. Due to the increased load it puts on the server, it shouldn't be used in a production environment.

Disk allocation

On Linux operating systems, you may see a discrepancy between what Wowza Streaming Engine reports as the disk utilization/fullness of the cache disks and what's reported by the operating system. Media Cache pre-allocates the entire file size when an item enters the cache, and then fills the file in chunks based on what was viewed. If you add up all of the file sizes of the individual chunks on the disk, you'll see that this number matches the value reported by Media Cache.

Linux only allocates a disk block at the OS level when it's used. For example, if the file size is 100 MB, the OS reports 10 MB used if only the first 10 MB are filled. You'll see this discrepancy if you have a high level of abandonment, where viewers only watch the start of most files.

Media Cache performance on Windows

Media Cache may not perform as expected when running on Windows. If you encounter problems, try running Media Cache on Linux or OS X with the version of Java that automatically installs with Wowza Streaming Engine.

Chapter

8

XML-based configuration examples

This chapter provides some example source configurations that you can create by editing the **MediaCache.xml** file located at **[*install-dir*]/conf/**. These configurations can also be replicated using Media Cache in Wowza Streaming Engine Manager.

Basic HTTP source configuration

This basic HTTP configuration enables streaming of any HTTP-based content through Media Cache.

```
<MediaCacheSource>
      <Name>http</Name>
      <BasePath>http://</BasePath>
      <Prefix>http/</Prefix>
      <BaseClass>com.wowza.wms.mediacache.impl.MediaCacheItemHTTPImpl</Bas
eClass>
      <ReaderClass></ReaderClass>
      <DefaultBlockSize>256K</DefaultBlockSize>
      <MaxTimeToLive>1200000</MaxTimeToLive>
      <MinTimeToLive>600000</MinTimeToLive>
      <ReadAhead>true</ReadAhead>
      <ReadAheadThreshold>50</ReadAheadThreshold>
      <IsPassThru>false</IsPassThru>
      <Properties>
      </Properties>
</MediaCacheSource>
```

For example, assume you have a media file at the following URL:

```
http://mycompany.com/media/sample.mp4
```

To play this stream, use the following URLs and stream names:

To play using Adobe Flash Player (RTMP)

```
Server: rtmp://[wowza-ip-address]/mediacache
Stream: mp4:/sample.mp4
```

To play using Adobe Flash Player (San Jose/Adobe HDS)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/mycompany.com/media/sample.mp4/
manifest.f4m
```

To play using an Apple iOS device (Cupertino/Apple HLS)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/mycompany.com/media/sample.mp4/
playlist.m3u8
```

To play using Microsoft Silverlight (Microsoft Smooth Streaming)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/mycompany.com/media/sample.mp4/
Manifest
```

To play using MPEG-DASH

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/mycompany.com/media/sample.mp4/
manifest.mpd
```

To play using RTSP/RTP player or device

```
rtsp://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http/mycompany.com/media/sample.mp4
```

For more information, see Set up video-on-demand streaming in Wowza Streaming Engine.

HTTP source with domain restriction

This HTTP configuration limits streaming to a particular domain. In this example, the prefix **http1/** maps to the base path **http://mycompany.com**.

```
<MediaCacheSource>
	<Name>http1</Name>
	<BasePath>http://mycompany.com/</BasePath>
	<Prefix>http1/</Prefix>
	<BaseClass>com.wowza.wms.mediacache.impl.MediaCacheItemHTTPImpl</Bas
eClass>
	<ReaderClass></ReaderClass>
	<DefaultBlockSize>256K</DefaultBlockSize>
```

```
        <MaxTimeToLive>1200000</MaxTimeToLive>
        <MinTimeToLive>600000</MinTimeToLive>
        <ReadAhead>true</ReadAhead>
        <ReadAheadThreshold>50</ReadAheadThreshold>
        <IsPassThru>false</IsPassThru>
        <Properties>
        </Properties>
</MediaCacheSource>
```

For example, assume you have a media file at the following URL:

```
http://mycompany.com/media/sample.mp4
```

To play this stream, use the following URLs and stream names:

To play using Adobe Flash Player (RTMP)

```
Server: rtmp://[wowza-ip-address]/mediacache
Stream: mp4:http1/media/sample.mp4
```

To play using Adobe Flash Player (San Jose/Adobe HDS)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http1/media/sample.mp4/manifest.f4m
```

To play using an Apple iOS device (Cupertino/Apple HLS)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http1/media/sample.mp4/playlist.m3u8
```

To play using Microsoft Silverlight (Microsoft Smooth Streaming)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http1/media/sample.mp4/Manifest
```

To play using MPEG-DASH

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http1/media/sample.mp4/manifest.mpd
```

To play using RTSP/RTP player or device

```
rtsp://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:http1/media/sample.mp4
```

For more information, see Set up video-on-demand streaming in Wowza Streaming Engine.

HTTP source with forward-proxy routing

This HTTP configuration limits streaming to a particular domain and sends all requests through a forward proxy server.

```
<MediaCacheSource>
      <Name>http1</Name>
      <BasePath>http://www.mycompany.com/</BasePath>
      <Prefix>http1/</Prefix>
      <BaseClass>com.wowza.wms.mediacache.impl.MediaCacheItemHTTPImpl</Bas
eClass>
      <ReaderClass></ReaderClass>
      <DefaultBlockSize>256K</DefaultBlockSize>
      <MaxTimeToLive>1200000</MaxTimeToLive>
      <MinTimeToLive>600000</MinTimeToLive>
      <ReadAhead>true</ReadAhead>
      <ReadAheadThreshold>50</ReadAheadThreshold>
      <IsPassThru>false</IsPassThru>
      <Properties>
            <Property>
                  <Name>proxyHost</Name>
                  <Value>myproxy.mycompany.com</Value>
            </Property>
            <Property>
                  <Name>proxyPort</Name>
                  <Value>8080</Value>
                  <Type>Integer</Type>
            </Property>
            <Property>
                  <Name>requestFullURL</Name>
                  <Value>true</Value>
                  <Type>Boolean</Type>
            </Property>
      </Properties>
</MediaCacheSource>
```

The proxy server details are specified through the **proxyHost**, **proxyPort**, and **requestFullURL** properties. The **requestFullURL** property instructs Media Cache to use full URLs when making requests to the proxy server. This is a great way to create a multiple-tier caching system. For example, if you want to serve the same set of files from multiple data centers, you would set up a forward proxy server in each data center. You would then configure the edge servers in a particular data center to use that data center's forward proxy server. All requests would be sent to the data center proxy server. If this proxy server can't fulfill the request, then it would be sent to the origin for fulfillment.

Basic file source configuration

This configuration lets you stream content on a network-attached file system through Media Cache. The base class **com.wowza.wms.mediacache.impl.MediaCacheItemFileImpl** instructs Media Cache to use the **MediaCacheSource** file-based implementation to deliver content. In this example, the prefix **path1/** maps to the base path **C:/content/**.

```
<MediaCacheSource>
      <Name>content1</Name>
      <BasePath>C:/content/</BasePath>
      <Prefix>path1/</Prefix>
      <BaseClass>com.wowza.wms.mediacache.impl.MediaCacheItemFileImpl</Bas
eClass>
      <ReaderClass></ReaderClass>
      <DefaultBlockSize>256K</DefaultBlockSize>
      <MaxTimeToLive>1200000</MaxTimeToLive>
      <MinTimeToLive>600000</MinTimeToLive>
      <ReadAhead>true</ReadAhead>
      <ReadAheadThreshold>50</ReadAheadThreshold>
      <IsPassThru>false</IsPassThru>
      <Properties>
      </Properties>
</MediaCacheSource>
```

For example, assume you have a media file at the following URL:

```
C:/content/media/sample.mp4
```

To play this stream, use the following URLs and stream names:

To play using Adobe Flash Player (RTMP)

```
Server: rtmp://[wowza-ip-address]/mediacache
Stream: mp4:path1/media/sample.mp4
```

To play using Adobe Flash Player (San Jose/Adobe HDS)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:path1/media/sample.mp4/manifest.f4m
```

To play using an Apple iOS device (Cupertino/Apple HLS)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:path1/media/sample.mp4/playlist.m3u8
```

To play using Microsoft Silverlight (Microsoft Smooth Streaming)

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:path1/media/sample.mp4/Manifest
```

To play using MPEG-DASH

```
http://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:path1/media/sample.mp4/manifest.mpd
```

To play using RTSP/RTP player or device

```
rtsp://[wowza-ip-
address]:1935/mediacache/_definst_/mp4:path1/media/sample.mp4
```

For more information, see Set up video-on-demand streaming in Wowza Streaming Engine.

HTTP source with DVR repeater

This HTTP, multi-server configuration caches DVR chunks on edge servers. Media Cache is installed on each edge server and requests chunks from a Wowza Streaming Engine origin in a live stream repeater (origin/edge) configuration.

```
<MediaCacheSource>
      <Name>dvrorigin</Name>
      <BasePath>http://</BasePath>
      <Prefix>dvrorigin/</Prefix>
      <BaseClass>com.wowza.wms.mediacache.impl.MediaCacheItemHTTPImpl</Bas
eClass>
      <ReaderClass></ReaderClass>
      <DefaultBlockSize>262144</DefaultBlockSize>
      <MaxTimeToLive>1200000</MaxTimeToLive>
      <MinTimeToLive>600000</MinTimeToLive>
      <ReadAhead>true</ReadAhead>
      <ReadAheadThreshold>50</ReadAheadThreshold>
      <IsPassThru>false</IsPassThru>
      <Properties>
            <Property>
                  <Name>httpReadTimeout</Name>
                  <Value>3000</Value>
                  <Type>Integer</Type>
            </Property>
      </Properties>
</MediaCacheSource>
```

For more information, see Configure Wowza Streaming Engine Media Cache for Wowza nDVR.

HTTPS (SSL) source configuration

This basic HTTPS configuration enables streaming of any HTTP-based content through Media Cache using Secure Sockets Layer (SSL).

```
<MediaCacheSource>
      <Name>https</Name>
      <BasePath>https://</BasePath>
      <Prefix>https/</Prefix>
      <BaseClass>com.wowza.wms.mediacache.impl.MediaCacheItemHTTPImpl</Bas
eClass>
      <ReaderClass></ReaderClass>
      <DefaultBlockSize>262144</DefaultBlockSize>
      <MaxTimeToLive>1200000</MaxTimeToLive>
      <MinTimeToLive>600000</MinTimeToLive>
      <ReadAhead>true</ReadAhead>
      <ReadAheadThreshold>50</ReadAheadThreshold>
      <IsPassThru>false</IsPassThru>
      <Properties>
            <Property>
                  <Name>httpReaderFactoryClass</Name>
                  <Value>com.wowza.wms.mediacache.impl.MediaCacheItem
HTTPSSLReaderFactory</Value>
                  <Type>String</Type>
                  </Property>
      </Properties>
</MediaCacheSource>
```