**Path**

**Name**

**Type**

**Value**

/Root/Application/MediaReader

randomAccessReaderClass

String

com.wowza.wms.plugin.RandomAccessReaderMediaCacheLocalContent

---

*MediaCacheLocalContent* is a custom *IRandomAccessReader* implementation for Wowza Streaming Engine™ media server software that looks for a piece of content in an on-demand application’s content storage directory and, if it can’t find it, tries to retrieve the content from a remote Media Cache source. This is useful when you need to stream locally stored content alongside remote content. For example, you could use this module if you have popular content that you need to store locally or if you have content that was just recorded and you need to make it available before transferring it to the remote server.

---

**Prerequisites**

Wowza Streaming Engine 4.0.0 or later is required.

---

**Installation**

1. Download *wse-plugin-mediacachelocalcontent.zip*.
2. Extract the contents from the downloaded (zipped) package, and then copy the *lib/wse-plugin-mediacachelocalcontent.jar* file from the package to the *lib* folder in your Wowza Streaming Engine installation (*[install-dir]/lib*).

---

**Configuration**

This Media Reader can be added to new or existing on-demand (VOD) applications and is enabled by setting a custom property in the application. See [*Configure properties*](#) for details.

---

**Properties**

After enabling the Media Reader, you can adjust the default settings by adding the following properties to your application. See [*Configure properties*](#) for details.

---

**Notes:**

- The *Path* setting isn’t the same as for a regular module property.
- If the application is a VOD Edge application, this property will have been set already and you should update the *Value*. There can be only one *randomAccessReaderClass* property.

---

**Path**

**Name**

**Type**

**Value**

/Root/Application/MediaReader

mediaCacheLocalContentMediaCachePath

String

http/

/Root/Application

mediaCacheLocalContentClassName

String

com.wowza.wms.mediacache.impl.MediaCacheRandomAccessReaderVODEdge
Setting the mediaCacheLocalContentMediaCachePath property

The `mediaCacheLocalContentMediaCachePath` property can use the following variable placeholders, which are replaced automatically while the content is being located.

- `${com.wowza.wms.context.VHost}` - Virtual host name
- `${com.wowza.wms.context.Application}` - Application name
- `${com.wowza.wms.context.ApplicationInstance}` - Application instance name

**Notes:**

- If the application is configured to only work with some Media Cache sources, the `mediaCacheLocalContentMediaCachePath` property value must begin with the prefix of one of those sources.
- If the content can't be located locally or on the Media Cache source, a "Stream Not Found (404)" error is sent to the player.

Playback

When a player requests a piece of content, the following sequence of events occur:

1. The application looks in the local content storage directory for the piece of content, and delivers it to the player if it finds it.
2. If the content isn't in the local content storage directory, the requested stream path is compared to the `mediaCacheLocalContentMediaCachePath` property value. One of the following sets of events occur, depending on the comparison:

   **Note:** If the path set in the property has variable placeholders, the variables are resolved first, resulting in the value of the actual path.

   - If the requested stream path starts with the path resolved from the `mediaCacheLocalContentMediaCachePath` property or if the property value is empty, the requested stream path is passed directly to the Media Cache Reader as a full path to the content on the Media Cache source. It's assumed that the path set in the player is a valid path in the Media Cache source.
   - If the requested stream path doesn't start with the path resolved from the `mediaCacheLocalContentMediaCachePath` property, the property path value is added to the start of the requested stream path. This creates the full path to the content on the Media Cache source.

Examples

1. **Condition:** Application content directory contains `sample.mp4`.
   **Result:** A player requests `sample.mp4`, which is returned from the local content directory.

2. **Condition:** Application content directory is empty, the `mediaCacheLocalContentMediaCachePath` property is set to `mysource/`, and there's a Media Cache source stored in the top-level directory on the Media Cache source with the prefix set to `mysource/sample.mp4`.
   **Result:** A player requests `sample.mp4`, which isn't found in the local content directory. So `mysource/` is added to the beginning of the path or the requested stream name and `mysource/sample.mp4` is requested from the Media Cache.

3. **Condition:** Application content directory contains `sample.mp4`, the `mediaCacheLocalContentMediaCachePath` property is set to `mysource/`, and `sample.mp4` is stored in the top-level directory on the Media Cache source.
   **Result:** A player requests `mysource/sample.mp4`, which isn't found in the local content directory (there's no local folder named `mysource`). So `mysource/sample.mp4` is requested from the Media Cache.

4. **Condition:** A live application records a live stream and temporarily stores the recorded files in the local content directory. Periodically, the content is uploaded to the Media Cache Source (with prefix set to `mysource/`) and the local copies are deleted. The `mediaCacheLocalContentMediaCachePath` property is set to `mysource/`.
   **Result:** A player requests `myStream.mp4`, which was just recorded and is in the local content directory. The stream is returned from the local copy. Later, a player requests the same stream name (`myStream.mp4`). This stream isn't found in the local content directory because it was uploaded to the Media Cache source. The `mysource/` prefix is prepended to the requested stream name and the resulting stream name is requested from the Media Cache.

5. **Condition:** A live application is configured to record streams that are sent to two different Media Cache sources. The application records the first stream (`myStream.mp4`) to a folder named `mysource/` in the local content directory. It records the second stream (`stream1.mp4`) to a folder named `myothersource/monday/` in the content directory.
The first Media Cache source has the prefix set to `mysource/` and the content from this stream is uploaded to the top-level folder. The other Media Cache source has a prefix set to `myothersource/` and the content uploads to a folder on the source named `monday`.

**Notes:**
- In this scenario, it's important that the paths (relative to the content directory) where the content is stored locally match the prefix and any internal paths defined by the Media Cache sources.
- Because players can access multiple Media Cache sources, the `mediaCacheLocalContentMediaCachePath` property value should be empty. Players must request the full Media Cache source paths (including prefixes).

**Result:** A player requests `mysource/myStream.mp4`, which is returned from the local copy if it's still available locally. Otherwise, it's returned from the Media Cache. The application tries to find the content (`myStream.mp4`) on the Media Cache source that has the prefix `mysource/`.

Another player can request `myothersource/monday/stream1.mp4` and if it's not available locally, it's requested from the Media Cache, which will look for the content (`monday/stream1.mp4`) on the Media Cache source that has the prefix `myothersource/`.

**Note:** In a situation where the same content could be in either local or remote sources, the content should be fully uploaded to the Media Cache source and verified before the local copy is deleted. This will ensure uninterrupted playback.

**For developers**

- Get the source code on [GitHub](#)

**More resources**

- [Scale video-on-demand streaming with Wowza Streaming Engine Media Cache](#)
- [Wowza Media Cache Deployment Architecture (PDF)](#)