

## **Purpose**

---

This implementation specification will outline what streaming media protocols, devices, and file formats the RUcore repository will support using the Wowza streaming media server. Also, this document will specify how the Wowza streaming media server will connect to and address the supported datastreams stored in the repository. Finally, the mechanism involved in providing the streaming media file to the media player will be explained.

## **Protocols**

---

Real Time Messaging Protocol Encrypted (RTMPE) will be used to support devices that have Adobe Flash installed. These devices are more commonly known as “desktops” but also include laptops or any device with Adobe Flash. The encrypted portion of the RTMPE protocol is handled using a shared secret/token between the media player, JWPlayer in this case, and the Wowza streaming media server.

HTTP Live Streaming (HLS) will be implemented as fallback when Adobe Flash is not detected on the device. HLS is support on all iOS devices, iPads & iPhones. HLS support on Android devices is minimal.

## **Devices**

---

Any devices with Adobe Flash installed will be supported. Any devices with iOS installed will be supported. Devices with the Android operating system will not be supported in this release. As Android HLS support grows and JWPlayer support progresses in this area Android support will be offered.

## **File Formats**

---

MPEG-4 (MP4) support will be offered using both the RTMPE and HLS protocols.

Flash Video (FLV) support will be limited to devices that have Adobe Flash installed. The HLS protocol does not support Flash Video files.

QuickTime Movie (MOV) support will be offered using both the RTMPE and HLS protocols.

MPEG-1 or MPEG-2 Audio Layer III (MP3) will be offered using both RTMPE and HLS protocols.

## **Wowza/Repository Connection**

---

The simplest and most direct way of providing Wowza access to the repository would be to mount the repository data storage location to a path in the Wowza content directory. This will allow the Wowza application to address all of the possible datastreams it needs to access. Figure 1 demonstrates how this would be accomplished using a Network File System (NFS) mount. The mount point would point at the base Fedora datastreams directory, i.e the directory with the listings of the datastreams starting with the year.

## **Providing Media File Location**

---

The video player will need to provide the Wowza media server with a relative path to the media file at the repository storage NFS mount point. The Get API which is used to mitigate access to datastreams and information about a

resources datastreams can be used to accomplish this. This is accomplished by configuring a datastream in the Workflow Management System (WMS) to have its content-disposition set to “inline” and setting the media server value to “Wowza.” This will inform the Get API via the file policy created to block direct downloads of the datastream and instead offer path information and Wowza streaming media server information.

Below is an example that compares the Get API’s list methods pre and post this configuration change.

#### *WMS configuration – content-disposition is set to “attachment”*

```
<file>
  <id>MP4-1</id>
  <label>MP4-1</label>
  <dateCreated>2014-02-25T14:56:37.000Z</dateCreated>
  <formatURI/>
  <fname>pres-1-00070411.mp4</fname>
  <mimeType>mp4</mimeType>
  <checksum type="SHA-256">f4c512f3af7866fa150bf1c6e32c7a57165e597ce32b90bb...</checksum>
  <ref>https://rucore-devel.libraries.rutgers.edu/api/get/rutgers-lib:201785/MP4/1</ref>
  <status>200</status>
  <size unit="bytes">41235976</size>
</file>
```

#### *WMS configuration – content disposition set to “inline” and media server is set to “Wowza”*

```
<file>
  <id>MP4-1</id>
  <label>MP4-1</label>
  <dateCreated>2014-02-25T14:56:37.000Z</dateCreated>
  <formatURI/>
  <fname>pres-1-00070411.mp4</fname>
  <mimeType>mp4</mimeType>
  <checksum type="SHA-256">f4c512f3af7866fa150bf1c6e32c7a57165e597ce32b90bb...</checksum>
  <server>stream1.libraries.rutgers.edu:1935/vod/_definst_/</server>
  <loc>isilon_rep_devel/2014/0225/14/56/rutgers-lib_201785+MP4-1+MP4-1.0</loc>
  <status>200</status>
  <size unit="bytes">41235976</size>
</file>
```

In the “attachment” example notice the `<ref>` element, this indicates direct access to the datastream, i.e. downloadable. After the configuration change the `<ref>` element is gone and any URI manipulation to it will not work. In place of the `<ref>` element a `<server>` and `<loc>` element are provided. These value are used the media player, JWPlayer, to form a connection with the Wowza media server for playback.

## **Impact Elsewhere**

---

**Darwin** - Since we are going to start supporting MOV datastreams using the Wowza media server we can discontinue providing video content via the existing Darwin Streaming.

**WMS** – Since we are decommissioning streaming via the Darwin Streaming Server the WMS will no longer need to copy MOV datastreams to an alternate location on ingest. Existing MOV datastreams in the repository will provide legacy support. New video objects that will be ingested will only need an MP4 video files. The FLV and MOV options should be removed and in their place a single MP4 option should exist.

Search retrieval – In the public search display the legacy FLV and/or MOV datastreams should be suppressed in favor of the MP4 datastreams. Legacy support will continue if a user has bookmarked a FLV or MOV datastream, but existing links should be removed when a MP4 is available. This will remove a good bit of confusion for the user as they will only have one choice instead of two when playing back a video.

Embedding the playback on the main record screen is now possible as well and should be explored. MP3 support and playback will remain unchanged unless we want to embed MP3 playback on the main record result screen.

Analytic Tool – The analytic tool uses FLV datastreams and will start using MP4 datastream when they are available in the resource. For legacy support the FLV datastreams will still be used in some cases. We should consider migrating published analytics from using the FLV datastream to the MP4 datastream at some point.

## Looking Ahead

---

After this initial release and when funds allow we should explore providing Digital Rights Management to the HLS protocol. At the same time for the RTMPE protocol we should start enforcing the SWF Hotlink protection that is native to the Wowza media server.

**Figure 1**

---

