Insert a pre-roll or mid-roll for video on demand playback in Flash RTMP client with the Wowza Streaming Engine Java API

Originally Published on 10/03/2010 | Updated on 11/01/2019 9:44 am PDT

The following code example shows how to create a video-on-demand, server-side playlist with the Wowza Streaming Engine™ Java API. In this example, the requested video is played in two parts with a midroll in between. You could play any number of streams whole or in part as one stream. You can use the stream name requested by the client along with other videos, as in this example, or substitute anything you want.

This code example requires Flash client. There is not an equivalent for non-Flash clients at present. For a client-side work-around for iOS see Do pre-roll and playlist for iOS, but it doesn’t offer seamless playback. A similar workaround is possible in a Silverlight client. There is no RTSP workaround.
Add this module to the /conf/[app-name]/Application.xml Module list. Add it to the bottom, make it the last Module.
Add these properties to Application.xml /Properties.

To send the calculated duration from playlist above to the Flash client, create this MediaReaderMP4InjectMetadata Class, then modify /conf/MediaReaders.xml as shown.
To hook up this class edit [install-dir]/conf/MediaReaders.xml and change the BassClass for

```java
public class MediaReaderMP4InjectMetadata extends MediaReaderH264 {
  public List getMetadata() {
    List ret = super.getMetadata();
    WMSLogger getLogger = WMSLoggerFactory.getLogger(null);
    // Here the calculated duration stored by above is extracted.
    // Note that the default duration is used if there is no duration property in the stream, so simple video streaming will work as usual
    double overrideDuration = this.stream.getProperties().getPropertyDouble("overrideDuration", (double) this.duration / 1000); // Value in seconds.
    getLogger.info("overrideDuration: " + duration);
    while (true) {
      if (ret == null) break;
      if (ret.size() <= 0) break;
      ByteBuffer packet = ret.get(0);
      AMFDataList myMetadata = new AMFDataList(packet);
      AMFDataMixedArray dataObj = (AMFDataMixedArray) myMetadata.get(1);
      dataObj.put("duration", new AMFDataItem(overrideDuration));  // duration in seconds
      byte[] data = myMetadata.serialize();
      ByteBuffer newPacket = ByteBuffer.wrap(data);
      ret.set(0, newPacket);
      break;
    }
    return ret;
  }
}
```
the mp4 file type from

to

Note: The above change is only possible with Wowza Media Server™ 3 or earlier. For Wowza Streaming Engine, use the following VHost listener.
To enable the VHost listener, edit `[install-dir]/conf/Server.xml` and add the following to the VHostListeners section.

This only works for H.264 video.