Create pseudo-live streams using files available to the application with a Wowza Streaming Engine server listener

Originally Published on 10/03/2010 | Updated on 05/17/2019 3:01 pm PDT

The following **StreamPublisher** example shows how to create a server listener with the Wowza Streaming Engine™ Java API that uses the **Stream** class functionality to create pseudo-live streams using files available to the application publishing the streams.

1. Compile the following server listener in the **Wowza IDE**:

```java
package com.wowza.forum;
import com.wowza.wms.server.*;
import com.wowza.wms.vhost.*;
import com.wowza.wms.stream.publish.*;
public class StreamPublisher implements IServerNotify {
    public void onServerCreate(IServer server) {
    }
    public void onServerInit(IServer server) {
        IVHost vhost = VHostSingleton.getInstance(VHost.VHOST_DEFAULT);
        Stream stream1 = Stream.createInstance(vhost, "live", "Stream1");
        stream1.play("mp4:sample.mp4", 5, 5, true);
        stream1.play("mp4:sample.mp4", 50, 5, false);
        stream1.play("mp4:sample.mp4", 150, 5, false);
        Stream stream2 = Stream.createInstance(vhost, "live", "Stream2");
        stream2.play("mp4:sample.mp4", 0, -1, true);
    }
    public void onServerShutdownComplete(IServer server) {
    }
    public void onServerShutdownStart(IServer server) {
    }
}
```

2. Add the server listener to `/conf/Server.xml` `/ServerListeners`:
3. Create a Wowza application named **live** and change the Application.xml /StreamType to **live**.

4. To play, open
   
   [install-dir]/examples/LiveVideoStreaming/FlashRTMPPlayer/player.html
   
   **Server:** rtmp://[wowza-ip-address]/live
   
   **Stream:** Stream1 (or Stream2)

The **Stream** class functionality provides the ability to add files or streams that will be played sequentially or looped based on configuration of each entry.

In the above example, the play function takes the following arguments:

- **play** (string filename, int startTime, int length, Boolean reset)
- **filename** - name of media item
- **start** - where to start playing the item. (-2 implies play a live stream)
- **length** - how much of the item to play (-1 implies play the entire file or live stream)
- **reset** - if **true**, will begin a new playlist, otherwise items are appended

If adding a live stream then the next item will not be played without interaction with the API.