



BonzaWare.Com

URL Handler Module Installation Guide

BonzaWare.Com

Version 1.1

Copyright Notice

© Copyright 2012

The information which is contained in this document is copyright BonzaWare.Com.
Prospero and Pickle Ltd (trading as BonzaWare.Com)

Commercial In Confidence

Document Control

Change History

Date	Author	Version	Change Reference
18 th May 2012	BonzaWare	1.0	First Release
21 st May 2012	BonzaWare	1.1	Included additional configuration example to aid simpler deployments.

Table of Contents

1.	Overview	4
2.	Obtaining the BonzaWare URLHandler Module	4
3.	Download Steps	4
4.	Installation Steps	5
5.	Server Configuration	5
6.	URLHandler Properties	6
1.	Connect Handler	6
2.	Play Handler	9
3.	Disconnect Handler	12
4.	Pause Handler	14
5.	Seek Handler	15
6.	Stop Handler	16
7.	Example Configuration	19
7.	Additional Application.xml properties	19
8.	Back End XML Files	20
7.1.1.	Connect XML file	20
7.1.2.	Play XML file	20
7.1.3.	Pause XML file	20
7.1.4.	Stop XML file	20
7.1.5.	Seek XML file	20
9.	Example Application.xml file	21

1. Overview

This document outlines how to install and configure the BonzaWare.Com URLHandler module for Wowza 3 or above.

2. Obtaining the BonzaWare URLHandler Module

Signup for an account with BonzaWare at <https://www.bonzaware.com/signup/>

3. Download Steps

Download the URLHandler module from your account. You can login to your BonzaWare.Com account at

<http://www.BonzaWare.Com/login/>

Select the Purchase option and then the Extras section located at the bottom right of the page.

Select the URLHandler module page and choose the appropriate format to download, ZIP or TAR.

4. Installation Steps

To install the URLHandler module you need to copy the BonzaWare-URLHandler.jar file provided in the ZIP or TAR file to the lib/ directory of your Wowza Installation.

5. Server Configuration

In order to activate the URLHandler module add the following as the last Module in the Modules section in the Application.xml

```
<Module>
  <Name>BonzaWareURLHandler</Name>
  <Description>BonzaWareURLHandler</Description>
  <Class>com.bonzaaware.free.url.URLNotifications</Class>
</Module>
```

For the purposes of this document we will assume an application called urlhandler has been created.

6. URLHandler Properties

1. Connect Handler

Parameter Name	Description
URLConnectHandler (optional) If omitted all connect requests are allowed automatically	This property defines a URL that information will be sent to when a client make a connect request. An example form would be <code>http://192.168.1.1/connect/index.html</code> It should NOT contain any parameters as ? will be post fixed and all attributes will be separated by &.
ConnectPARAM1 (optional)	This property defines the attribute name that will be used to send PARAM1 via the URL form. An example would be to set ConnectPARAM1 to ID, this would make the URL sent to the URLConnectHandler become <code>http://192.168.1.1./connect/index.html?ID=<PARAM1 value></code>
ConnectPARAM2 (optional)	This property defines the attribute name that will be used to send PARAM2 via the URL form. An example would be to set ConnectPARAM2 to ID, this would make the URL sent to the URLConnectHandler become <code>http://192.168.1.1./connect/index.html?ID=<PARAM2 value></code>
ConnectPARAM3 (optional)	This property defines the attribute name that will be used to send PARAM3 via the URL form. An example would be to set ConnectPARAM3 to ID, this would make the URL sent to the URLConnectHandler become <code>http://192.168.1.1./connect/index.html?ID=<PARAM3 value></code>
ConnectPARAM4 (optional)	This property defines the attribute name that will be used to send PARAM4 via the URL form. An example would be to set ConnectPARAM4 to ID, this would make the URL sent to the URLConnectHandler become <code>http://192.168.1.1./connect/index.html?ID=<PARAM4 value></code>
ConnectPARAM5 (optional)	This property defines the attribute name that will be used to send PARAM5 via the URL form. An example would be to set ConnectPARAM5 to ID, this would make the URL sent to the URLConnectHandler become <code>http://192.168.1.1./connect/index.html?ID=<PARAM5 value></code>
ConnectPARAM6 (optional)	This property defines the attribute name that will be used to send PARAM6 via the URL form. An example would be to set ConnectPARAM6 to ID, this would make the URL sent to the URLConnectHandler become <code>http://192.168.1.1./connect/index.html?ID=<PARAM6 value></code>
ConnectPARAM7 (optional)	This property defines the attribute name that will be used to send PARAM7 via the URL form. An example would be to set ConnectPARAM7 to ID, this would make the URL sent to the URLConnectHandler become <code>http://192.168.1.1./connect/index.html?ID=<PARAM7 value></code>

ConnectPARAM8 (optional)	<p>This property defines the attribute name that will be used to send PARAM8 via the URL form. An example would be to set ConnectPARAM8 to ID, this would make the URL sent to the URLConnectHandler become</p> <p><code>http://192.168.1.1./connect/index.html?ID=<PARAM8 value></code></p>
ConnectSuccess (Required)	<p>This property should contain the value that will be sent back to Wowza on a successful connect. All responses to the HTTP request are XML and defined later in the document.</p>
ConnectIP (Optional)	<p>This property should contain 'On' to enable the IP of the connecting client to be sent as a parameter in the HTTP requests. The parameter sent is IP.</p>
ConnectProtocol (Optional)	<p>This property should contain 'On' to enable the protocol of the connecting client to be sent as a parameter in the HTTP requests. The parameter sent is Protocol.</p>
ConnectReferrer (Optional)	<p>This property should contain 'On' to enable the referrer of the connecting client to be sent as a parameter in the HTTP requests. The parameter sent is Referrer.</p>
ConnectFlashVer (Optional)	<p>This property should contain 'On' to enable the flash player version of the connecting client to be sent as a parameter in the HTTP requests. The parameter sent is FlashVer.</p>
ConnectFlashCap (Optional)	<p>This property should contain 'On' to enable the flash player capabilities of the connecting client to be sent as a parameter in the HTTP requests. The parameter sent is FlashCap.</p>
ConnectClientID (Optional)	<p>This property should contain 'On' to enable the connection identification of the connecting client to be sent as a parameter in the HTTP requests. The parameter sent is ClientID.</p>
ConnectEpoch (Optional)	<p>This property should contain 'On' to enable the current epoch of the Wowza server to be sent as a parameter in the HTTP requests. The parameter sent is Epoch.</p>
ConnectURLParams (Optional)	<p>This property defines the URL parameters sent in the RTMP string that will be proxied in the connect HTTP request.</p> <p>For example the RTMP could be</p> <p>Server URL : <code>rtmp://192.168.1.1/testapplication/?hello=there</code></p> <p>If this parameter is set to hello then the ConnectHTTP request will look like</p> <p><code>http://192.168.1.1/connect/index.html?hello=there</code></p> <p>Multiple parameters are supported although unless configured correctly on the client only one is sent from the client.</p>

ConnectPrivName1	This property defines an attribute name to be sent to Wowza and uses the value in the ConnectPrivValue1 property. This is a fixed property name and value.
ConnectPrivValue1	<p>The property defines the value sent with the attribute named in ConnectPrivName1.</p> <p>An example would be</p> <p>ConnectPrivName1 set to ServerID</p> <p>ConnectPrivValue1 set to 100</p> <p>The connect URL would contain</p> <p>http://192.168.1.1/connect/index.html?ServerID=100</p>
ConnectPrivName2	This property defines an attribute name to be sent to Wowza and uses the value in the ConnectPrivValue2 property. This is a fixed property name and value.
ConnectPrivValue2	<p>The property defines the value sent with the attribute named in ConnectPrivName2.</p> <p>An example would be</p> <p>ConnectPrivName2 set to CompanyName</p> <p>ConnectPrivValue2 set to BonzaWare</p> <p>The connect URL would contain</p> <p>http://192.168.1.1/connect/index.html?ServerID=100&CompanyName=BonzaWare</p>
ConnectPrivName3	This property defines an attribute name to be sent to Wowza and uses the value in the ConnectPrivValue3 property. This is a fixed property name and value.
ConnectPrivValue3	<p>The property defines the value sent with the attribute named in ConnectPrivName3.</p> <p>An example would be</p> <p>ConnectPrivName3 set to Product</p> <p>ConnectPrivValue3 set to Free</p> <p>The connect URL would contain</p> <p>http://192.168.1.1/connect/index.html?ServerID=100&CompanyName=BonzaWare&Product=Free</p>
ConnectPrivName4	See example for ConnectPrivName1
ConnectPrivValue4	See Example for ConnectPrivValue1
ConnectPrivName5	See example for ConnectPrivName1
ConnectPrivValue5	See Example for ConnectPrivValue1

2. Play Handler

Parameter Name	Description
URLPlayHandler (optional) If omitted all play requests are allowed automatically	This property defines the URL that information will be sent to when a client makes a play request. An example form would be <code>http://192.168.1.1/play/index.html</code> It should NOT contain any parameters as ? will be post fixed and all attributes will be separated by &.
PlayPARAM1 (optional)	This property defines the attribute name that will be used to send PARAM1 via the URL form. An example would be to set PlayPARAM1 to ID, this would make the URL sent to the URLPlayHandler become <code>http://192.168.1.1/play/index.html?ID=<PARAM1 value></code>
PlayPARAM2 (optional)	This property defines the attribute name that will be used to send PARAM2 via the URL form. An example would be to set PlayPARAM2 to ID, this would make the URL sent to the URLPlayHandler become <code>http://192.168.1.1/play/index.html?ID=<PARAM2 value></code>
PlayPARAM3 (optional)	This property defines the attribute name that will be used to send PARAM3 via the URL form. An example would be to set PlayPARAM3 to ID, this would make the URL sent to the URLPlayHandler become <code>http://192.168.1.1/play/index.html?ID=<PARAM3 value></code>
PlayPARAM4 (optional)	This property defines the attribute name that will be used to send PARAM4 via the URL form. An example would be to set PlayPARAM4 to ID, this would make the URL sent to the URLPlayHandler become <code>http://192.168.1.1/play/index.html?ID=<PARAM4 value></code>
PlayPARAM5 (optional)	This property defines the attribute name that will be used to send PARAM5 via the URL form. An example would be to set PlayPARAM5 to ID, this would make the URL sent to the URLPlayHandler become <code>http://192.168.1.1./playt/?ID=<PARAM5 value></code>
PlayPARAM6 (optional)	This property defines the attribute name that will be used to send PARAM6 via the URL form. An example would be to set PlayPARAM6 to ID, this would make the URL sent to the URLPlayHandler become <code>http://192.168.1.1/play/index.html?ID=<PARAM6 value></code>
PlayPARAM7 (optional)	This property defines the attribute name that will be used to send PARAM7 via the URL form. An example would be to set PlayPARAM7 to ID, this would make the URL sent to the URLPlayHandler become <code>http://192.168.1.1/play/index.html?ID=<PARAM7 value></code>

PlayPARAM8 (optional)	This property defines the attribute name that will be used to send PARAM8 via the URL form. An example would be to set PlayPARAM8 to ID, this would make the URL sent to the URLPlayHandler become <code>http://192.168.1.1/play/index.html?ID=<PARAM8 value></code>
PlaySuccess (Required)	This property should contain the value that will be sent back to Wowza on a successful connect. All responses to the HTTP request are XML and defined later in the document.
PlayIP (Optional)	This property should contain 'On' to enable the IP of the connecting client to be sent as a parameter in the HTTP requests. The parameter sent is IP.
PlayProtocol (Optional)	This property should contain 'On' to enable the protocol of the connecting client to be sent as a parameter in the HTTP requests. The parameter sent is Protocol.
PlayReferrer (Optional)	This property should contain 'On' to enable the referrer of the connecting client to be sent as a parameter in the HTTP requests. The parameter sent is Referrer.
PlayFlashVer (Optional)	This property should contain 'On' to enable the flash player version of the connecting client to be sent as a parameter in the HTTP requests. The parameter sent is FlashVer.
PlayFlashCap (Optional)	This property should contain 'On' to enable the flash player capabilities of the connecting client to be sent as a parameter in the HTTP requests. The parameter sent is FlashCap.
PlayClientID (Optional)	This property should contain 'On' to enable the connection identification of the connecting client to be sent as a parameter in the HTTP requests. The parameter sent is ClientID.
PlayEpoch (Optional)	This property should contain 'On' to enable the current epoch of the Wowza server to be sent as a parameter in the HTTP requests. The parameter sent is Epoch.
PlayURLParams (Optional)	<p>This property defines the URL parameters sent in the RTMP string that will be proxied in the connect HTTP request.</p> <p>For example the RTMP could be</p> <p>Server URL : <code>rtmp://192.168.1.1/testapplication/?hello=there</code></p> <p>If this parameter is set to hello then the play HTTP request will look like</p> <p><code>http://192.168.1.1/play/index.html?hello=there</code></p> <p>Multiple parameters are supported although unless configured correctly on the client only one is sent from the client.</p>
PlayStreamName	This property should contain 'On' to enable the stream name of the playing client to be sent as a parameter in the HTTP requests. The parameter sent is StreamName.

PlayPrivName1	This property defines an attribute name to be sent to Wowza and uses the value in the PlayPrivValue1 property. This is a fixed property name and value.
PlayPrivValue1	<p>The property defines the value sent with the attribute named in PlayPrivName1.</p> <p>An example would be</p> <p>PlayPrivName1 set to ServerID</p> <p>PlayPrivValue1 set to 100</p> <p>The play URL would contain</p> <p>http://192.168.1.1/play/index.html?ServerID=100</p>
PlayPrivName2	This property defines an attribute name to be sent to Wowza and uses the value in the PlayPrivValue2 property. This is a fixed property name and value.
PlayPrivValue2	<p>The property defines the value sent with the attribute named in PlayPrivName2.</p> <p>An example would be</p> <p>PlayPrivName2 set to CompanyName</p> <p>PlayPrivValue2 set to BonzaWare</p> <p>The play URL would contain</p> <p>http://192.168.1.1/play/index.html?ServerID=100&CompanyName=BonzaWare</p>
PlayPrivName3	This property defines an attribute name to be sent to Wowza and uses the value in the PlayPrivValue3 property. This is a fixed property name and value.
PlayPrivValue3	<p>The property defines the value sent with the attribute named in PlayPrivName3.</p> <p>An example would be</p> <p>PlayPrivName3 set to Product</p> <p>PlayPrivValue3 set to Free</p> <p>The play URL would contain</p> <p>http://192.168.1.1/play/index.html?ServerID=100&CompanyName=BonzaWare&Product=Free</p>
PlayPrivName4	See example for PlayPrivName1
PlayPrivValue4	See Example for PlayPrivValue1
PlayPrivName5	See example for PlayPrivName1
PlayPrivValue5	See Example for PlayPrivValue1

3. Disconnect Handler

Parameter Name	Description
URLDisconnectHandler (optional) If omitted all play requests are allowed automatically	This property defines the URL that information will be sent to when a client makes a disconnect request. An example form would be <code>http://192.168.1.1/disconnect/index.html</code> It should NOT contain any parameters as ? will be post fixed and all attributes will be separated by &.
DisconSuccess (Optional + Not Required)	This is not required but can be set to anything. It is included for consistency.
DisconIP (Optional)	This property should contain 'On' to enable the IP of the disconnecting client to be sent as a parameter in the HTTP requests. The parameter sent is IP.
DisconProtocol (Optional)	This property should contain 'On' to enable the protocol of the disconnecting client to be sent as a parameter in the HTTP requests. The parameter sent is Protocol.
DisconReferrer (Optional)	This property should contain 'On' to enable the referrer of the disconnecting client to be sent as a parameter in the HTTP requests. The parameter sent is Referrer.
DisconFlashVer (Optional)	This property should contain 'On' to enable the flash player version of the disconnecting client to be sent as a parameter in the HTTP requests. The parameter sent is FlashVer.
DisconFlashCap (Optional)	This property should contain 'On' to enable the flash player capabilities of the disconnecting client to be sent as a parameter in the HTTP requests. The parameter sent is FlashCap.
DisconClientID (Optional)	This property should contain 'On' to enable the connection identification of the disconnecting client to be sent as a parameter in the HTTP requests. The parameter sent is ClientID.
DisconEpoch (Optional)	This property should contain 'On' to enable the current epoch of the Wowza server to be sent as a parameter in the HTTP requests. The parameter sent is Epoch.
DisconURLParams (Optional)	This property defines the URL parameters sent in the RTMP string that will be proxied in the disconnect HTTP request. For example the RTMP could be Server URL : <code>rtmp://192.168.1.1/testapplication/?hello=there</code> If this parameter is set to hello then the Disconnect HTTP request will look like <code>http://192.168.1.1/disconnect/index.html?hello=there</code> Multiple parameters are supported although unless configured correctly on the client only one is sent from the client.
DisconStreamName	This property should contain 'On' to enable the stream name of the disconnecting client to be sent as a parameter in the HTTP requests. The parameter sent is StreamName.

DisconOutBytes	This property should contain 'On' to enable the outbound byte count of the disconnecting client to be sent as a parameter in the HTTP requests. The parameter sent is OutBytes.
DisconInBytes	This property should contain 'On' to enable the inbound byte count of the disconnecting client to be sent as a parameter in the HTTP requests. The parameter sent is InBytes.
DisconOutAvgKbps	This property should contain 'On' to enable the average outbound bitrate of the disconnecting client to be sent as a parameter in the HTTP requests. The parameter sent is OutAvgKbps.
DisconInAvgKbps	This property should contain 'On' to enable the average inbound bitrate of the disconnecting client to be sent as a parameter in the HTTP requests. The parameter sent is InAvgKbps.
DisconPrivName1	This property defines an attribute name to be sent to Wowza and uses the value in the DisconPrivValue1 property. This is a fixed property name and value.
DisconPrivValue1	<p>The property defines the value sent with the attribute named in DisconPrivName1.</p> <p>An example would be</p> <p>DisconPrivName1 set to ServerID</p> <p>DisconPrivValue1 set to 100</p> <p>The disconnect URL would contain</p> <p>http://192.168.1.1/disconnect/index.html?ServerID=100</p>
DisconPrivName2	This property defines an attribute name to be sent to Wowza and uses the value in the DisconPrivValue2 property. This is a fixed property name and value.
DisconPrivValue2	<p>The property defines the value sent with the attribute named in DisconPrivName2.</p> <p>An example would be</p> <p>DisconPrivName2 set to CompanyName</p> <p>DisconPrivValue2 set to BonzaWare</p> <p>The disconnect URL would contain</p> <p>http://192.168.1.1/disconnect/index.html?ServerID=100&CompanyName=BonzaWare</p>
DisconPrivName3	This property defines an attribute name to be sent to Wowza and uses the value in the DisconPrivValue3 property. This is a fixed property name and value.
DisconPrivValue3	<p>The property defines the value sent with the attribute named in DisconPrivName3.</p> <p>An example would be</p> <p>DisconPrivName3 set to Product</p> <p>DisconPrivValue3 set to Free</p> <p>The disconnect URL would contain</p> <p>http://192.168.1.1/disconnect/index.html?ServerID=100&CompanyName=BonzaWare&Product=Free</p>
DisconPrivName4	See example for DisconPrivName1
DisconPrivValue4	See Example for DisconPrivValue1
DisconPrivName5	See example for DisconPrivName1
DisconPrivValue5	See Example for DisconPrivValue1

4. Pause Handler

Parameter Name	Description
URLPauseHandler (optional) If omitted all play requests are allowed automatically	This property defines the URL that information will be sent to when a client makes a pause request. An example form would be <code>http://192.168.1.1/pause/index.html</code> It should NOT contain any parameters as ? will be post fixed and all attributes will be separated by &.
PauseIP (Optional)	This property should contain 'On' to enable the IP of the pausing client to be sent as a parameter in the HTTP requests. The parameter sent is IP.
PauseProtocol (Optional)	This property should contain 'On' to enable the protocol of the pausing client to be sent as a parameter in the HTTP requests. The parameter sent is Protocol.
PauseReferrer (Optional)	This property should contain 'On' to enable the referrer of the pausing client to be sent as a parameter in the HTTP requests. The parameter sent is Referrer.
PauseFlashVer (Optional)	This property should contain 'On' to enable the flash player version of the pausing client to be sent as a parameter in the HTTP requests. The parameter sent is FlashVer.
PauseFlashCap (Optional)	This property should contain 'On' to enable the flash player capabilities of the pausing client to be sent as a parameter in the HTTP requests. The parameter sent is FlashCap.
PauseClientID (Optional)	This property should contain 'On' to enable the connection identification of the pausing client to be sent as a parameter in the HTTP requests. The parameter sent is ClientID.
PauseEpoch (Optional)	This property should contain 'On' to enable the current epoch of the Wowza server to be sent as a parameter in the HTTP requests. The parameter sent is Epoch.
PauseURLParams (Optional)	This property defines the URL parameters sent in the RTMP string that will be proxied in the connect HTTP request. For example the RTMP could be Server URL : <code>rtmp://192.168.1.1/testapplication/?hello=there</code> If this parameter is set to hello then the Pause HTTP request will look like <code>http://192.168.1.1/pause/index.html?hello=there</code> Multiple parameters are supported although unless configured correctly on the client only one is sent from the client.

5. Seek Handler

Parameter Name	Description
URLSeekHandler (optional) If omitted no seek requests are made	This property defines the URL that information will be sent to when a client makes a seek request. An example form would be http://192.168.1.1/seek/index.html It should NOT contain any parameters as ? will be post fixed and all attributes will be separated by &.
SeekIP (Optional)	This property should contain 'On' to enable the IP of the seeking client to be sent as a parameter in the HTTP requests. The parameter sent is IP.
SeekProtocol (Optional)	This property should contain 'On' to enable the protocol of the seeking client to be sent as a parameter in the HTTP requests. The parameter sent is Protocol.
SeekReferrer (Optional)	This property should contain 'On' to enable the referrer of the seeking client to be sent as a parameter in the HTTP requests. The parameter sent is Referrer.
SeekFlashVer (Optional)	This property should contain 'On' to enable the flash player version of the seeking client to be sent as a parameter in the HTTP requests. The parameter sent is FlashVer.
SeekFlashCap (Optional)	This property should contain 'On' to enable the flash player capabilities of the seeking client to be sent as a parameter in the HTTP requests. The parameter sent is FlashCap.
SeekClientID (Optional)	This property should contain 'On' to enable the connection identification of the seeking client to be sent as a parameter in the HTTP requests. The parameter sent is ClientID.
SeekEpoch (Optional)	This property should contain 'On' to enable the current epoch of the Wowza server to be sent as a parameter in the HTTP requests. The parameter sent is Epoch.
SeekURLParams (Optional)	This property defines the URL parameters sent in the RTMP string that will be proxied in the seek HTTP request. For example the RTMP could be Server URL : rtmp://192.168.1.1/testapplication/?hello=there If this parameter is set to hello then the seek HTTP request will look like http://192.168.1.1/seek/index.html?hello=there Multiple parameters are supported although unless configured correctly on the client only one is sent from the client.

6. Stop Handler

Parameter Name	Description
URLStopHandler (optional) If omitted no stop requests are made	This property defines the URL that information will be sent to when a client makes a stop request. An example form would be <code>http://192.168.1.1/stop/index.html</code> It should NOT contain any parameters as ? will be post fixed and all attributes will be separated by &.
StopIP (Optional)	This property should contain 'On' to enable the IP of the stopping client to be sent as a parameter in the HTTP requests. The parameter sent is IP.
StopProtocol (Optional)	This property should contain 'On' to enable the protocol of the stopping client to be sent as a parameter in the HTTP requests. The parameter sent is Protocol.
StopReferrer (Optional)	This property should contain 'On' to enable the referrer of the stopping client to be sent as a parameter in the HTTP requests. The parameter sent is Referrer.
StopFlashVer (Optional)	This property should contain 'On' to enable the flash player version of the stopping client to be sent as a parameter in the HTTP requests. The parameter sent is FlashVer.
StopFlashCap (Optional)	This property should contain 'On' to enable the flash player capabilities of the stopping client to be sent as a parameter in the HTTP requests. The parameter sent is FlashCap.
StopClientID (Optional)	This property should contain 'On' to enable the connection identification of the stopping client to be sent as a parameter in the HTTP requests. The parameter sent is ClientID.
StopEpoch (Optional)	This property should contain 'On' to enable the current epoch of the Wowza server to be sent as a parameter in the HTTP requests. The parameter sent is Epoch.
StopURLParams (Optional)	This property defines the URL parameters sent in the RTMP string that will be proxied in the stop HTTP request. For example the RTMP could be Server URL : <code>rtmp://192.168.1.1/testapplication/?hello=there</code> If this parameter is set to hello then the stop HTTP request will look like <code>http://192.168.1.1/stop/index.html?hello=there</code> Multiple parameters are supported although unless configured correctly on the client only one is sent from the client.
StopStreamName	This property should contain 'On' to enable the stream name of the stop client to be sent as a parameter in the HTTP requests. The parameter sent is StreamName.

StopOutBytes	This property should contain 'On' to enable the outbound byte count of the disconnecting client to be sent as a parameter in the HTTP requests. The parameter sent is OutBytes.
StopInBytes	This property should contain 'On' to enable the inbound byte count of the disconnecting client to be sent as a parameter in the HTTP requests. The parameter sent is InBytes.
StopOutAvgKbps	This property should contain 'On' to enable the average outbound bitrate of the disconnecting client to be sent as a parameter in the HTTP requests. The parameter sent is OutAvgKbps.
StopInAvgKbps	This property should contain 'On' to enable the average inbound bitrate of the disconnecting client to be sent as a parameter in the HTTP requests. The parameter sent is InAvgKbps.
StopPrivName1	This property defines an attribute name to be sent to Wowza and uses the value in the StopPrivName1 property. This is a fixed property name and value.
StopPrivValue1	<p>The property defines the value sent with the attribute named in StopPrivName1.</p> <p>An example would be</p> <p>StopPrivName1 set to ServerID</p> <p>StopPrivValue1 set to 100</p> <p>The stop URL would contain</p> <p>http://192.168.1.1/stop/index.html?ServerID=100</p>
StopPrivName2	This property defines an attribute name to be sent to Wowza and uses the value in the StopPrivValue2 property. This is a fixed property name and value.
StopPrivValue2	<p>The property defines the value sent with the attribute named in StopPrivName2.</p> <p>An example would be</p> <p>StopPrivName2 set to CompanyName</p> <p>StopPrivValue2 set to BonzaWare</p> <p>The stop URL would contain</p> <p>http://192.168.1.1/stop/index.html?ServerID=100&CompanyName=BonzaWare</p>
StopPrivName3	This property defines an attribute name to be sent to Wowza and uses the value in the DisconPrivValue3 property. This is a fixed property name and value.
StopPrivValue3	<p>The property defines the value sent with the attribute named in PlayPrivName3.</p> <p>An example would be</p> <p>StopPrivName3 set to Product</p> <p>StopPrivValue3 set to Free</p> <p>The stop URL would contain</p> <p>http://192.168.1.1/stop/index.html?ServerID=100&CompanyName=BonzaWare&Product=Free</p>
StopPrivName4	See example for StopPrivName1

StopPrivValue4	See Example for StopPrivValue1
StopPrivName5	See example for StopPrivName1
StopPrivValue5	See Example for StopPrivValue1

7. Example Configuration

For the purposes of this document we will assume there is a HTTP server available at the IP address 192.168.1.1. The following properties should be added to the bottom of your Application.xml for the application you wish to enable URL Handling for

7. Additional Application.xml properties

```
<Property>
  <Name>URLConnectHandler</Name>
  <Value>http://192.168.1.1/connect/index.html</Value>
</Property>
<Property>
  <Name>ConnectSuccess</Name>
  <Value>1</Value>
</Property>
<Property>
  <Name>URLPlayHandler</Name>
  <Value>http://192.168.1.1/play/index.html</Value>
</Property>
<Property>
  <Name>PlaySuccess</Name>
  <Value>1</Value>
</Property>
<Property>
  <Name>URLDisconnectHandler</Name>
  <Value>http://192.168.1.1/disconnect/index.html</Value>
</Property>
```

8. Back End XML Files

7.1.1. Connect XML file

The contents of your index.html file contents at <http://192.168.1.1/connect/>

```
<?xml version="1.0"?>
<result>
    <success>1</success>
</result>
```

7.1.2. Play XML file

The contents of your index.html file contents at <http://192.168.1.1/play/>

```
<?xml version="1.0"?>
<result>
    <success>1</success>
</result>
```

7.1.3. Pause XML file

The contents of your index.html file contents at <http://192.168.1.1/pause/>

```
<?xml version="1.0"?>
<result>
    <success>1</success>
</result>
```

7.1.4. Stop XML file

The contents of your index.html file contents at <http://192.168.1.1/stop/>

```
<?xml version="1.0"?>
<result>
    <success>1</success>
</result>
```

7.1.5. Seek XML file

The contents of your index.html file contents at <http://192.168.1.1/seek/>

```
<?xml version="1.0"?>
<result>
    <success>1</success>
</result>
```

9. Example Application.xml file

The following example Application.xml is for video on demand (VoD) and will work with a default installation of Wowza 3.1. You should be able to change the URLs pointed to and test using the sample.mp4 provided with Wowza Media Server.

```
<Root>
  <Application>
    <!-- Uncomment to set application level timeout values
    <ApplicationTimeout>60000</ApplicationTimeout>
    <PingTimeout>12000</PingTimeout>
    <ValidationFrequency>8000</ValidationFrequency>
    <MaximumPendingWriteBytes>0</MaximumPendingWriteBytes>
    <MaximumSetBufferTime>60000</MaximumSetBufferTime>
    <MaximumStorageDirDepth>25</MaximumStorageDirDepth>
    -->
    <Connections>
      <AutoAccept>true</AutoAccept>
      <AllowDomains></AllowDomains>
    </Connections>
    <!--
      StorageDir path variables

      ${com.wowza.wms.AppHome} - Application home directory
      ${com.wowza.wms.ConfigHome} - Configuration home directory
      ${com.wowza.wms.context.VHost} - Virtual host name
      ${com.wowza.wms.context.VHostConfigHome} - Virtual host config directory
      ${com.wowza.wms.context.Application} - Application name
      ${com.wowza.wms.context.ApplicationInstance} - Application instance name

    -->
    <Streams>
      <StreamType>file</StreamType>
      <StorageDir>${com.wowza.wms.context.VHostConfigHome}/content</StorageDir>
      <KeyDir>${com.wowza.wms.context.VHostConfigHome}/keys</KeyDir>
      <!-- LiveStreamPacketizers (separate with commas): cupertinostreamingpacketizer, smoothstreamingpacketizer,
      sanjosestreamingpacketizer, cupertinostreamingrepeater, smoothstreamingrepeater, sanjosestreamingrepeater, dvrstreamingpacketizer, dvrstreamingrepeater -->
      <LiveStreamPacketizers></LiveStreamPacketizers>
      <!-- Properties defined here will override any properties defined in conf/Streams.xml for any streams types loaded by this
      application -->
      <Properties>
      </Properties>
    </Streams>
    <Transcoder>
      <!-- To turn on transcoder set to: transcoder -->
```

```

    <LiveStreamTranscoder></LiveStreamTranscoder>
    <!-- [templatename].xml or ${SourceStreamName}.xml -->
    <Templates>${SourceStreamName}.xml,transrate.xml</Templates>
    <ProfileDir>${com.wowza.wms.context.VHostConfigHome}/transcoder/profiles</ProfileDir>
    <TemplateDir>${com.wowza.wms.context.VHostConfigHome}/transcoder/templates</TemplateDir>
    <Properties>
    </Properties>
</Transcoder>

<DVR>
    <!-- As a single server or as an origin, use dvrstreamingpacketizer in LiveStreamPacketizers above -->
    <!-- Or, in an origin-edge configuration, edges use dvrstreamingrepeater in LiveStreamPacketizers above -->
    <!-- As an origin, also add dvrchunkstreaming to HTTPStreamers below -->

    <!-- To turn on DVR recording set Recorders to dvrrecorder. This works with dvrstreamingpacketizer -->
    <Recorders></Recorders>

    <!-- As a single server or as an origin, set the Store to dvrfilestorage-->
    <!-- edges should have this empty -->
    <Store></Store>

    <!-- Window Duration is length of live DVR window in seconds. 0 means the window is never trimmed. -->
    <WindowDuration>0</WindowDuration>

    <!-- Storage Directory is top level location where dvr is stored. e.g. c:/temp/dvr -->
    <StorageDir>${com.wowza.wms.context.VHostConfigHome}/dvr</StorageDir>

    <!-- valid ArchiveStrategy values are append, version, delete -->
    <ArchiveStrategy>append</ArchiveStrategy>

    <!-- If this is a dvrstreamingrepeater, define ChunkOriginURL to point back to origin -->
    <!-- And define Application/Repeater/OriginURL to point back to the origin -->
    <Repeater>
        <ChunkOriginURL></ChunkOriginURL>
    </Repeater>

    <!-- Properties for DVR -->
    <Properties>
    </Properties>
</DVR>

>
    <!-- HTTPStreamers (separate with commas): cupertinostreaming, smoothstreaming, sanjosestreaming, dvrchunkstreaming -->
    <HTTPStreamers>cupertinostreaming,smoothstreaming,sanjosestreaming</HTTPStreamers>
    <SharedObjects>

```

```

    <StorageDir></StorageDir>
</SharedObjects>
<Client>
  <IdleFrequency>-1</IdleFrequency>
  <Access>
    <StreamReadAccess>*</StreamReadAccess>
    <StreamWriteAccess>*</StreamWriteAccess>
    <StreamAudioSampleAccess></StreamAudioSampleAccess>
    <StreamVideoSampleAccess></StreamVideoSampleAccess>
    <SharedObjectReadAccess>*</SharedObjectReadAccess>
    <SharedObjectWriteAccess>*</SharedObjectWriteAccess>
  </Access>
</Client>
<RTP>
  <!-- RTP/Authentication/[type]Methods defined in Authentication.xml. Default setup includes; none, basic, digest -->
  <Authentication>
    <PublishMethod>digest</PublishMethod>
    <PlayMethod>none</PlayMethod>
  </Authentication>
  <!-- RTP/AVSyncMethod. Valid values are: senderreport, systemclock, rtpimecode -->
  <AVSyncMethod>senderreport</AVSyncMethod>
  <MaxRTCPWaitTime>12000</MaxRTCPWaitTime>
  <IdleFrequency>75</IdleFrequency>
  <RTSPSessionTimeout>90000</RTSPSessionTimeout>
  <RTSPMaximumPendingWriteBytes>0</RTSPMaximumPendingWriteBytes>
  <RTSPBindIpAddress></RTSPBindIpAddress>
  <RTSPConnectionIpAddress>0.0.0.0</RTSPConnectionIpAddress>
  <RTSPOriginIpAddress>127.0.0.1</RTSPOriginIpAddress>
  <IncomingDatagramPortRanges>*</IncomingDatagramPortRanges>
  <!-- Properties defined here will override any properties defined in conf/RTP.xml for any depacketizers loaded by this
application -->
  <Properties>
  </Properties>
</RTP>
<MediaCaster>
  <RTP>
    <RTSP>
      <!-- udp, interleave -->
      <RTPTransportMode>interleave</RTPTransportMode>
    </RTSP>
  </RTP>
  <!-- Properties defined here will override any properties defined in conf/MediaCasters.xml for any MediaCasters loaded
by this applications -->
  <Properties>
  </Properties>
</MediaCaster>

```

```

    <MediaReader>
      <!-- Properties defined here will override any properties defined in conf/MediaReaders.xml for any MediaReaders loaded
by this applications -->
      <Properties>
      </Properties>
    </MediaReader>
    <MediaWriter>
      <!-- Properties defined here will override any properties defined in conf/MediaWriter.xml for any MediaWriter loaded by
this applications -->
      <Properties>
      </Properties>
    </MediaWriter>
    <LiveStreamPacketizer>
      <!-- Properties defined here will override any properties defined in conf/LiveStreamPacketizers.xml for any
LiveStreamPacketizers loaded by this applications -->
      <Properties>
      </Properties>
    </LiveStreamPacketizer>
    <HTTPStreamer>
      <!-- Properties defined here will override any properties defined in conf/HTTPStreamers.xml for any HTTPStreamer
loaded by this applications -->
      <Properties>
    <Property>
      <Name>cupertinoRelativePlaylistPlaylists</Name>
      <Value>false</Value>
      <Type>Boolean</Type>
    </Property>
      </Properties>
    </HTTPStreamer>
    <Repeater>
      <OriginURL></OriginURL>
      <QueryString><![CDATA[]]></QueryString>
    </Repeater>
    <Modules>
      <Module>
        <Name>base</Name>
        <Description>Base</Description>
        <Class>com.wowza.wms.module.ModuleCore</Class>
      </Module>
      <Module>
        <Name>properties</Name>
        <Description>Properties</Description>
        <Class>com.wowza.wms.module.ModuleProperties</Class>
      </Module>
      <Module>
        <Name>logging</Name>
        <Description>Client Logging</Description>

```



```
<Class>com.wowza.wms.module.ModuleClientLogging</Class>
</Module>
<Module>
  <Name>flvplayback</Name>
  <Description>FLVPlayback</Description>
  <Class>com.wowza.wms.module.ModuleFLVPlayback</Class>
</Module>
<Module>
  <Name>BonzaWareURLHandler</Name>
  <Description> BonzaWareURLHandler </Description>
  <Class>com.bonzaaware.free.url.URLNotifications</Class>
</Module>
</Modules>
<!-- Properties defined here will be added to the IApplication.getProperties() and IApplicationInstance.getProperties()
collections -->
<Properties>
  <Property>
    <Name>URLConnectHandler</Name>
    <Value>http://192.168.1.1/play/index.html</Value>
  </Property>
  <Property>
    <Name>ConnectSuccess</Name>
    <Value>1</Value>
  </Property>
  <Property>
    <Name>URLPlayHandler</Name>
    <Value>http://192.168.1.1/play/index.html</Value>
  </Property>
  <Property>
    <Name>PlaySuccess</Name>
    <Value>1</Value>
  </Property>
  <Property>
    <Name>URLDisconnectHandler</Name>
    <Value>http://192.168.1.1/connect/index.html</Value>
  </Property>
</Properties>
</Application>
</Root>
```