HOW TO GET RED5 MEDIA SERVER DEPLOYED IN JBOSS AS 7

WHAT IS RED5?

From their web page (http://www.red5.org):

RED5 MEDIA SERVER 1.0 DELIVERS A POWERFUL VIDEO STREAMING AND MULTI-USER SOLUTION TO THE ©ADOBE ©FLASH PLAYER AND OTHER EXCITING CLIENT TECHNOLOGIES. BASED ON JAVA AND SOME OF THE MOST POWERFUL OPEN SOURCE FRAMEWORKS, RED5 STANDS AS A SOLID SOLUTION FOR BUSINESS OF ALL SIZES INCLUDING THE ENTERPRISE.

RED5 INCLUDES SUPPORT FOR THE LATEST MULTI-USER API'S INCLUDING NETCONNECTION, NETSTREAM AND SHAREDOBJECT'S WHILE PROVIDING A POWERFUL RTMP / SERVLET IMPLEMENTATION. IN ADDITION TO SUPPORT FOR THE RTMP PROTOCOL, THE APPLICATION SERVER HAS AN EMBEDDED TOMCAT SERVLET CONTAINER FOR JEE WEB APPLICATIONS. APPLICATION DEVELOPMENT DRAWS ADDITIONAL BENEFITS FROM THE SPRING FRAMEWORK AND SCOPE BASED EVENT DRIVEN SERVICES.

BY USING THE OPEN SOURCE RED5 MEDIA SERVER, YOU ARE DEVELOPING WITH A TRULY OPEN AND EXTENSIBLE PLATFORM THAT CAN BE USED IN VIDEO CONFERENCES, MULTI-USER GAMING AND ENTERPRISE APPLICATION SOFTWARE.

The Red5 Media Server uses the Spring Framework as their application server and, until the release under JBoss AS 7 was quite problematic due to dependencies conflicts. JBoss AS 7 presents us with a new modular and OSGi compliant architecture that facilitates this integration.

CREATING THE INITIAL RED5 EAR

PREPARE THE STAGE

1. JBoss AS 7 or higher (http://www.jboss.org/jbossas/downloads) – Only tested with Web Profile Only package
2. Red5 1.0 RC1 (WAR) - http://trac.red5.org/downloads/1_0/red5-war-1.0-RC1.zip
3. Unpack JBoss AS 7 package
4. In JBOSS_HOME/standalone/deployments create the following directory structure:
   - red5.ear

        - META-INF
        - Red5.war

CREATE THE RED5.JAR ARCHIVE

1. Unpack the downloaded red5-war-1.0-RC1.zip into a working directory
2. In it you'll find a ROOT.war, unpack it's contents into the directory called red5ear/red5.war
3. Move the red5.ear/red5.war/WEB-INF/lib into red5.ear
4. Package the directory red5.ear/red5.war/WEB-INF/classes/org into red5.jar
5. Move the created red5.jar into red5.ear/lib
6. Remove the red5.ear/red5.war/WEB-INF/classes/org directory

CREATE THE RED5-CONTEXT.JAR ARCHIVE

7. Pack the following files in a file named red5-context.jar:
    - red5.ear/red5.war/WEB-INF/classes/beanRefContext.xml
    - red5.ear/red5.war/WEB-INF/classes/defaultContext.xml
    - red5.ear/red5.war/WEB-INF/classes/red5-common.xml
    - red5.ear/red5.war/WEB-INF/classes/red5-core.xml
8. Move the created red5-context.jar to red5.ear/lib
9. Remove the following files:
    - red5.ear/red5.war/WEB-INF/classes/beanRefContext.xml
    - red5.ear/red5.war/WEB-INF/classes/defaultContext.xml
    - red5.ear/red5.war/WEB-INF/classes/red5-common.xml
    - red5.ear/red5.war/WEB-INF/classes/red5-core.xml

GETTING AND DEPLOYING THE INSTALLER APPLICATION

10. Checkout the installer directory from the SVN repository:
    - svn export http://red5.googlecode.com/svn/java/server/trunk/webapps/installer
11. Move the checked-out installer directory into red5.ear/red5.war
12. Rename red5.ear/red5.war/installer/WEB-INF/red5-web.xml to installer-web.xml and move it to red5.ear/red5.war/WEB-INF/classes
13. Remove the directory red5.ear/red5.war/installer/WEB-INF directory

GETTING AND DEPLOYING THE VOD WEB APPLICATION

14. Checkout the vod directory from the SVN repository:
    - svn export http://red5.googlecode.com/svn/java/server/trunk/webapps/vod
15. Move and rename the vod directory to red5.ear/vod.war
16. Create directory red5.ear/vod.war/WEB-INF/classes
17. Rename red5.ear/vod.war/WEB-INF/red5-web.xml to vod-web.xml and move it to red5.ear/vod.war/WEB-INF/classes
18. Remove red5.ear/vod.war/WEB-INF/red5-web.properties

CREATE EAR DESCRIPTORS

19. Create xml file red5.ear/META-INF/application.xml with the following contents:

```
<?xml version="1.0" encoding="UTF-8"?>
<application xmlns="http://java.sun.com/xml/ns/javaee"
        xmlns:xsi="http://www.w3.org/2001/XMLSchema-instance"
        xsi:schemaLocation="http://java.sun.com/xml/ns/javaee
http://java.sun.com/xml/ns/javaee/application_5.xsd"
        version="5">

 <display-name>red5</display-name>

 <module>
  <web>
    <web-uri>red5.war</web-uri>
    <context-root>/red5</context-root>
   </web>
 </module>

 <module>
  <web>
    <web-uri>vod.war</web-uri>
    <context-root>/red5/vod</context-root>
   </web>
 </module>

</application>
```

20. Create xml file red5.ear/META-INF/MANIFEST.MF with the following contents:

```
Dependencies: org.apache.commons.logging export, org.slf4j export
```

21. Ensure that the inserted line ends with a new line. This ensures that the last line in the manifest file will not be skipped.
22. As an alternative to the MANIFEST.MF file you can also use the following contents in a file named file red5.ear/META-INF/jboss-deployment-structure.xml

```
<?xml version="1.0" encoding="UTF-8"?>

<jboss-deployment-structure>

  <deployment>

    <dependencies>

      <module name="org.apache.commons.logging"/>

       <module name="org.slf4j"/>

    </dependencies>

  </deployment>

</jboss-deployment-structure>
```

FIX WEB.XML FILES

Open red5.ear/red5.war/WEB-INF/web.xml and set the webAppRootKey to:

```
        <context-param>
                <param-name>webAppRootKey</param-name>
                <param-value>/red5</param-value>
        </context-param>
```

Open red5.ear/vod.war/WEB-INF/web.xml and add the following context params:

```
        <context-param>
                <param-name>globalScope</param-name>
                <param-value>default</param-value>
        </context-param>

        <context-param>
                <param-name>parentContextKey</param-name>
                <param-value>default.context</param-value>
        </context-param>

        <context-param>
                <param-name>contextConfigLocation</param-name>
                <param-value>WEB-INF/classes/*-web.xml</param-value>
        </context-param>

        <listener>
                 <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
        </listener>
```

set the webAppRootKey to:

```
        <context-param>
                <param-name>webAppRootKey</param-name>
                <param-value>/red5/vod</param-value>
        </context-param>
```

FIX *-WEB.XML FILES

Open red5.ear/red5.war/WEB-INF/classes/root-web.xml and change contextPath property to:

```
<property name="contextPath" value="/" />
```

Open red5.ear/red5.war/WEB-INF/classes/installer-web.xml remove the following:

```
        <bean id="placeholderConfig"
class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
          <property name="location" value="/WEB-INF/red5-web.properties" />
        </bean>
```

and update the following tags to:

```
        <bean id="installer-web.context" class="org.red5.server.Context" autowire="byType" />

        <bean id="installer-web.scope" class="org.red5.server.WebScope"
                 init-method="register">
                <property name="server" ref="red5.server" />
                <property name="parent" ref="global.scope" />
                <property name="context" ref="installer-web.context" />
                <property name="handler" ref="installer-web.handler" />
                <property name="contextPath" value="/installer" />
                <property name="virtualHosts" value="*" />
        </bean>

        <bean id="installer-web.handler" class="org.red5.server.adapter.ApplicationAdapter" />

        <bean id="installer.service" class="org.red5.server.service.Installer">
                <property name="applicationRepositoryUrl"
                          value="http://red5.googlecode.com/svn/snapshots/" />
        </bean>
```

Open red5.ear/vod.war/WEB-INF/classes/vod-web.xml remove the following:

```
        <bean id="placeholderConfig"
class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
          <property name="location" value="/WEB-INF/red5-web.properties" />
        </bean>
```

and update the following tags to:

```
        <bean id="vod-web.context" class="org.red5.server.Context" autowire="byType" />

        <bean id="web.scope" class="org.red5.server.WebScope" init-method="register">
                <property name="server" ref="red5.server" />
                <property name="parent" ref="global.scope" />
                <property name="context" ref="vod-web.context" />
                <property name="handler" ref="vod-web.handler" />
                <property name="contextPath" value="/vod" />
                <property name="virtualHosts" value="*" />
        </bean>

        <bean id="vod-web.handler" class="org.red5.server.adapter.ApplicationAdapter" />
```

DEPLOY & TEST IT

- Start JBoss AS 7 by running the following command in a terminal: JBOSS_HOME/bin/standalone(.bat/.sh)
- Deploy the red5.ear by creating an empty text file named red5.ear.dodeploy in: JBOSS_HOME/standalone/deployments
- If everything is well, you should now be able to point your browser to: http://localhost:8080/red5
- And confirm that:
    1. The VOD stream (Red5 intro video) is working
    2. The install link starts the installer application and shows a list of application names to be installed (don't install anything because it will not work).

ADD RED5 INSTALLER WAR PACKAGES

The following demo web applications can be downloaded by going into the Red5 installer and selecting the designated application. Once selected, click on the **Install** button and this will download the respective war file. These file will be available in JBOSS_HOME/standalone/deployments/red5.ear once downloaded.

SHARED OBJECT WEB APP

- Install the **SOSample** web application
- Rename the file to something else
- Create directory SOSample.war
- Extract contents of downloaded war into the SOSample.war directory
- Rename red5.ear/SOSample.war/WEB-INF/red5-web.xml to sosample-web.xml
- Move it to red5.ear/SOSample.war/WEB-INF/classes
- Delete red5.ear/SOSample.war/WEB-INF/red5-web.properties

FIX RED5.EAR/SOSAMPLE.WAR/WEB-INF/WEB.XML

Open **red5.ear/SOSample.war/WEB-INF/web.xml** for editing and add the following:

```
        <context-param>
                <param-name>globalScope</param-name>
                <param-value>default</param-value>
        </context-param>

        <context-param>
                <param-name>parentContextKey</param-name>
                <param-value>default.context</param-value>
        </context-param>

        <context-param>
                <param-name>contextConfigLocation</param-name>
                <param-value>WEB-INF/classes/*-web.xml</param-value>
        </context-param>

        <listener>
                 <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
        </listener>
```

set the webAppRootKey to:

```
        <context-param>
                <param-name>webAppRootKey</param-name>
                <param-value>/red5/SOSample</param-value>
        </context-param>
```

FIX RED5.EAR/SOSAMPLE.WAR/WEB-INF/CLASSES/SOSAMPLE-WEB.XML

Open red5.ear/SOSample.war/WEB-INF/classes/sosample-web.xml remove the following:

```
        <bean id="placeholderConfig"
class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
          <property name="location" value="/WEB-INF/red5-web.properties" />
        </bean>
```

and update the following tags to:

```
<bean id="vod-web.context" class="org.red5.server.Context" autowire="byType" />

<bean id="web.scope" class="org.red5.server.WebScope" init-method="register">
        <property name="server" ref="red5.server" />
        <property name="parent" ref="global.scope" />
        <property name="context" ref="vod-web.context" />
        <property name="handler" ref="vod-web.handler" />
        <property name="contextPath" value="/vod" />
        <property name="virtualHosts" value="*" />
</bean>

<bean id="vod-web.handler" class="org.red5.server.adapter.ApplicationAdapter" />
```

UPDATE APPLICATION.XML

Add the following tag to red5.ear/META-INF/application.xml:

```
<module>
  <web>
    <web-uri>SOSample.war</web-uri>
    <context-root>/red5/SOSample</context-root>
 </web>
</module>
```

TEST IT

- Start JBoss AS 7
- Open two browser instances side-by-side and pointing to the following URL: http://localhost:8080/red5/demos/BallControl.html
- In both pages, substitute the RTMP end-point to be: rtmp://localhost/red5/SOSample
- In both pages, click on the blue plug icon to connect to the server
- In any of the pages drag the Red5 logo and you should see it being also moved on the other page

- Now set the two browser instances to http://localhost:8080/red5/demos/SimpleChat.html
- Connect and on the bottom text input, write something and press return

ECHO WEB APP

- Install the **echo** web application
- Rename the file to something else
- Create directory echo.war

- Extract contents of downloaded war into the echo.war directory
- Rename red5.ear/echo.war/WEB-INF/red5-web.xml to echo-web.xml
- Move it to red5.ear/echo.war/WEB-INF/classes
- Delete red5.ear/echo.war/WEB-INF/red5-web.properties

FIX RED5.EAR/ECHO.WAR/WEB-INF/WEB.XML

Open **red5.ear/echo.war/WEB-INF/web.xml** for editing and add the following:

```
<context-param>
        <param-name>globalScope</param-name>
        <param-value>default</param-value>
</context-param>

<context-param>
        <param-name>parentContextKey</param-name>
        <param-value>default.context</param-value>
</context-param>

<context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>WEB-INF/classes/*-web.xml</param-value>
</context-param>

<listener>
         <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
```

Remove the following entries:

```
<filter>
  <filter-name>LoggerContextFilter</filter-name>
  <filter-class>org.red5.logging.LoggerContextFilter</filter-class>
</filter>

<listener>
  <listener-class>org.red5.logging.ContextLoggingListener</listener-class>
</listener>

<filter-mapping>
  <filter-name>LoggerContextFilter</filter-name>
  <url-pattern>/*</url-pattern>
</filter-mapping>
```

FIX RED5.EAR/ECHO.WAR/WEB-INF/CLASSES/ECHO-WEB.XML

Open red5.ear/echo.war/WEB-INF/classes/echo-web.xml remove the following:

```
<bean id="placeholderConfig"
```

```
class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
          <property name="location" value="/WEB-INF/red5-web.properties" />
        </bean>
```

and update the following tags to:

```
  <bean id="echo-web.context" class="org.red5.server.Context"
    autowire="byType" />

  <bean id="echo-web.scope" class="org.red5.server.WebScope"
     init-method="register">
    <property name="server" ref="red5.server" />
    <property name="parent" ref="global.scope" />
    <property name="context" ref="echo-web.context" />
    <property name="handler" ref="echo-web.handler" />
    <property name="contextPath" value="/echo" />
    <property name="virtualHosts" value="*" />
  </bean>

  <bean id="echo-web.handler" class="org.red5.demos.echo.Application" />

  <!-- enable support for mx:RemoteObject requests -->
  <bean id="flexMessaging.service" class="org.red5.server.net.remoting.FlexMessagingService">
    <property name="serviceInvoker" ref="global.serviceInvoker" />
    <!-- add an entry for all possible "destination" attributes of your
       mx:RemoteObject sources -->
    <property name="endpoints">
      <map>
        <entry key="Red5Echo">
          <!-- object that handles requests sent to this endpoint -->
          <ref bean="echo-web.handler" />
        </entry>
      </map>
    </property>
  </bean>
```

UPDATE APPLICATION.XML

Add the following tag to red5.ear/META-INF/application.xml:

```
 <module>
   <web>
     <web-uri>echo.war</web-uri>
     <context-root>/echo</context-root>
  </web>
 </module>
```

TEST IT

- Start JBoss AS 7

- Open browser instance in http://localhost:8080/red5/demos/echo_test.html
- Change HTTP port to http://localhost:8080/echo/gateway
- Execute tests for RTMP and HTTP AMF0/3 – mx:RemoteObject not working yet

OFLA DEMO WEB APP

- Install the **oflaDemo** web application
- Rename the file to something else
- Create directory oflaDemo.war
- Extract contents of downloaded war into the oflaDemo.war directory
- Rename red5.ear/oflaDemo.war/WEB-INF/red5-web.xml to oflaDemo-web.xml

FIX RED5.EAR/OFLADEMO.WAR/WEB-INF/WEB.XML

Open **red5.ear/oflaDemo.war/WEB-INF/web.xml** for editing and add the following:

```
        <context-param>
                <param-name>globalScope</param-name>
                <param-value>default</param-value>
        </context-param>

        <context-param>
                <param-name>parentContextKey</param-name>
                <param-value>default.context</param-value>
        </context-param>

        <context-param>
                <param-name>contextConfigLocation</param-name>
                <param-value>WEB-INF/classes/*-web.xml</param-value>
        </context-param>

        <listener>
                 <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
        </listener>
```

Remove the following entries:

```
  <listener>
    <listener-class>org.red5.logging.ContextLoggingListener</listener-class>
  </listener>

  <filter>
    <filter-name>LoggerContextFilter</filter-name>
    <filter-class>org.red5.logging.LoggerContextFilter</filter-class>
  </filter>

  <filter-mapping>
    <filter-name>LoggerContextFilter</filter-name>
```

```
      <url-pattern>/*</url-pattern>
   </filter-mapping>
```

FIX RED5.EAR/OFLADEMO.WAR/WEB-INF/CLASSES/OFLADEMO-WEB.XML

Open red5.ear/oflaDemo.war/WEB-INF/classes/oflaDemo-web.xml remove the following:

```
        <bean id="placeholderConfig"
class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
            <property name="location" value="/WEB-INF/red5-web.properties" />
        </bean>
```

and update the following tags to:

```
        <bean id="oflaDemo-web.context" class="org.red5.server.Context"
                autowire="byType" />

        <bean id="oflaDemo-web.scope" class="org.red5.server.WebScope"
                 init-method="register">
                <property name="server" ref="red5.server" />
                <property name="parent" ref="global.scope" />
                <property name="context" ref="oflaDemo-web.context" />
                <property name="handler" ref="oflaDemo-web.handler" />
                <property name="contextPath" value="/oflaDemo" />
                <property name="virtualHosts" value="*" />
        </bean>

        <bean id="oflaDemo-web.handler" class="org.red5.demos.oflaDemo.Application" />
```

UPDATE APPLICATION.XML

Add the following tag to red5.ear/META-INF/application.xml:

```
 <module>
   <web>
     <web-uri>oflaDemo.war</web-uri>
     <context-root>/oflaDemo</context-root>
  </web>
 </module>
```

TEST IT

- Start JBoss AS 7
- Open browser instance in http://localhost:8080/red5/demos/ofla_demo.html
- Click on **[connect]**
- Select a video from the table and view it

- Open browser instance in http://localhost:8080/red5/demos/publisher.html
- Under setting click on Connect
- Under the video tab, select a video device (e.g. web cam) and click on Start
- Under the audio tab, select a audio device and click on Start
- Click on Publish to start streaming video
- Click on view tab and press on Play to subscribe to published streaming channel

- Set browser instance to http://localhost:8080/red5/demos/simpleBroadcaster.html
- Press connect blue button
- Set another browser instance to http://localhost:8080/red5/demos/simpleSubscriber.html
- You should see the broadcasted video
- Set yet another browser instance to http://localhost:8080/red5/demos/simpleSubscriber.html
- Click on connect icon
- Click on Start Recording and when happy, click on Stop Recording
- The recorded file will be under red5.ear/oflaDemo.war/streams

FITC DEMO WEB APP

- Install the **fitcDemo** web application
- Rename the file to something else
- Create directory fitcDemo.war
- Extract contents of downloaded war into the fitcDemo.war directory
- Rename red5.ear/fitcDemo.war/WEB-INF/red5-web.xml to fitcDemo-web.xml

FIX RED5.EAR/FITCDEMO.WAR/WEB-INF/WEB.XML

Open **red5.ear/fitcDemo.war/WEB-INF/web.xml** for editing and add the following:

```
<context-param>
        <param-name>globalScope</param-name>
        <param-value>default</param-value>
</context-param>

<context-param>
        <param-name>parentContextKey</param-name>
        <param-value>default.context</param-value>
</context-param>

<context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>WEB-INF/classes/*-web.xml</param-value>
</context-param>

<listener>
```

```
                <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
        </listener>
```

Remove the following entries:

```
  <listener>
    <listener-class>org.red5.logging.ContextLoggingListener</listener-class>
  </listener>

  <filter>
    <filter-name>LoggerContextFilter</filter-name>
    <filter-class>org.red5.logging.LoggerContextFilter</filter-class>
  </filter>

  <filter-mapping>
    <filter-name>LoggerContextFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>

  <security-constraint>
    <web-resource-collection>
      <web-resource-name>Forbidden</web-resource-name>
      <url-pattern>/streams/*</url-pattern>
    </web-resource-collection>
    <auth-constraint/>
  </security-constraint>
```

FIX RED5.EAR/FITCDEMO.WAR/WEB-INF/CLASSES/FITCDEMO-WEB.XML

Open red5.ear/fitcDemo.war/WEB-INF/classes/fitcDemo-web.xml remove the following:

```
        <bean id="placeholderConfig"
class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
            <property name="location" value="/WEB-INF/red5-web.properties" />
        </bean>
```

and update the following tags to:

```
        <bean id="fitcDemo-web.context" class="org.red5.server.Context"
                autowire="byType" />

        <bean id="fitcDemo-web.scope" class="org.red5.server.WebScope"
                 init-method="register">
                <property name="server" ref="red5.server" />
                <property name="parent" ref="global.scope" />
                <property name="context" ref="fitcDemo-web.context" />
                <property name="handler" ref="fitcDemo-web.handler" />
                <property name="contextPath" value="/fitcDemo" />
                <property name="virtualHosts" value="*" />
        </bean>

        <bean id="fitcDemo-web.handler" class="org.red5.demos.fitc.Application" />
```

UPDATE APPLICATION.XML

Add the following tag to red5.ear/META-INF/application.xml:

```
<module>
  <web>
    <web-uri>fitcDemo.war</web-uri>
    <context-root>/fitcDemo</context-root>
 </web>
</module>
```

TEST IT

- Start JBoss AS 7

- Open browser instance in http://localhost:8080/red5/demos/FITCPresentation.swf
- Move to the “Demo – Video Conference” slide
- Set the RTMP address to **rtmp://localhost/fitcDemo**
- Click on blue plugs to connect to server
- You should see your web camera stream once you accept access to it
- Repeat above steps for a second web cam view

AUDIO TRANSCODER

Before you start you need to get lib **xuggle-xuggler.jar** from Xuggler web site (http://xuggle.com/xuggler/red5) and copy it into **red5.ear/lib** directory.

- Install the **audiotranscoder** web application
- Rename the file to something else
- Create directory audiotranscoder.war
- Extract contents of downloaded war into the audiotranscoder.war directory
- Create red5.ear/audiotranscoder.war/WEB-INF/classes directory
- Rename red5.ear/audiotranscoder.war/WEB-INF/red5-web.xml to audiotranscoder-web.xml
- Move audiotranscoder-web.xml to red5.ear/audiotranscoder.war/WEB-INF/classes

FIX RED5.EAR/AUDIOTRANSCODER.WAR/WEB-INF/WEB.XML

Open **red5.ear/audiotranscoder.war/WEB-INF/web.xml** for editing and add the following:

```
        <context-param>
                <param-name>globalScope</param-name>
                <param-value>default</param-value>
        </context-param>
        <context-param>
                <param-name>parentContextKey</param-name>
                <param-value>default.context</param-value>
        </context-param>

        <context-param>
                <param-name>contextConfigLocation</param-name>
                <param-value>WEB-INF/classes/*-web.xml</param-value>
        </context-param>

        <listener>
                 <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
        </listener>
```

Remove the following entries:

```
  <listener>
    <listener-class>org.red5.logging.ContextLoggingListener</listener-class>
  </listener>

  <filter>
    <filter-name>LoggerContextFilter</filter-name>
    <filter-class>org.red5.logging.LoggerContextFilter</filter-class>
  </filter>

  <filter-mapping>
    <filter-name>LoggerContextFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>

  <security-constraint>
    <web-resource-collection>
      <web-resource-name>Forbidden</web-resource-name>
      <url-pattern>/streams/*</url-pattern>
    </web-resource-collection>
    <auth-constraint/>
  </security-constraint>
```

FIX RED5.EAR/AUDIOTRANSCODER.WAR/WEB-INF/CLASSES/AUDIOTRANSCODER-WEB.XML

Open red5.ear/audiotranscoder.war/WEB-INF/classes/audiotranscoder-web.xml remove the following:

```
        <bean id="placeholderConfig"
class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
          <property name="location" value="/WEB-INF/red5-web.properties" />
        </bean>
```

and update the following tags to:

```
 <bean id="audiotranscoder-web.context" class="org.red5.server.Context" autowire="byType"/>

 <bean id="audiotranscoder-web.scope" class="org.red5.server.WebScope" init-method="register">
```

```
  <property name="server" ref="red5.server"/>
  <property name="parent" ref="global.scope"/>
  <property name="context" ref="audiotranscoder-web.context"/>
  <property name="handler" ref="audiotranscoder-web.handler"/>
  <property name="contextPath" value="/audiotranscoder"/>
  <property name="virtualHosts" value="*"/>
</bean>

<bean id="audiotranscoder-web.handler"
  class="com.xuggle.red5.demo.AudioTranscoderDemoAdapter"
  init-method="init">
  <!-- Have the profiler spit out data every 15 seconds -->
  <property name="profilerFrequency" value="15"/>
</bean>
```

UPDATE APPLICATION.XML

Add the following tag to red5.ear/META-INF/application.xml:

```
<module>
    <web>
        <web-uri>audiotranscoder.war</web-uri>
        <context-root>/audiotranscoder</context-root>
  </web>
</module>
```

TEST IT

Also see onther info at http://localhost:8080/audiotranscoder/index.html.

- Start JBoss AS 7

- Open browser instance in http://localhost:8080/red5/demos/publisher.html

- On Server tab set Location to **rtmp://localhost/audiotranscoder**
- Set Buffer to 0
- Click on Connect to connect to rtmp server

- On Audio tab set Device to your audio input device
- Click on Start to init the input video device

- On Publish tab set the Name to **example**
- Click on Publish to start streaming

- On View tab set Name to **example**
- Click on Play to play the **example** stream and you should hear whatever is being streamed by your audio device.

Test Xuggler

- Click on Stop to end playing the stream
- On the view tab, rename the stream to **xuggle_example**
- Click on Play to play the on-the-fly modified stream and confirm that the audio is now being echoed.

VIDEO TRANSCODER

Before you start you need to get lib **xuggle-xuggler.jar** from Xuggler web site (http://xuggle.com/xuggler/red5) and copy it into **red5.ear/lib** directory.

- Install the **videotranscoder** web application
- Rename the file to something else
- Create directory videotranscoder.war
- Extract contents of downloaded war into the videotranscoder.war directory
- Create red5.ear/videotranscoder.war/WEB-INF/classes directory
- Rename red5.ear/videotranscoder.war/WEB-INF/red5-web.xml to videotranscoder-web.xml
- Move audiotranscoder-web.xml to red5.ear/videotranscoder.war/WEB-INF/classes

FIX RED5.EAR/VIDEOTRANSCODER.WAR/WEB-INF/WEB.XML

Open **red5.ear/videotranscoder.war/WEB-INF/web.xml** for editing and add the following:

```
<context-param>
        <param-name>globalScope</param-name>
        <param-value>default</param-value>
</context-param>

<context-param>
        <param-name>parentContextKey</param-name>
        <param-value>default.context</param-value>
</context-param>

<context-param>
        <param-name>contextConfigLocation</param-name>
        <param-value>WEB-INF/classes/*-web.xml</param-value>
</context-param>

<listener>
         <listener-class>org.springframework.web.context.ContextLoaderListener</listener-class>
</listener>
```

Remove the following entries:

```
<listener>
```

```
    <listener-class>org.red5.logging.ContextLoggingListener</listener-class>
  </listener>

  <filter>
    <filter-name>LoggerContextFilter</filter-name>
    <filter-class>org.red5.logging.LoggerContextFilter</filter-class>
  </filter>

  <filter-mapping>
    <filter-name>LoggerContextFilter</filter-name>
    <url-pattern>/*</url-pattern>
  </filter-mapping>

  <security-constraint>
    <web-resource-collection>
      <web-resource-name>Forbidden</web-resource-name>
      <url-pattern>/streams/*</url-pattern>
    </web-resource-collection>
    <auth-constraint/>
  </security-constraint>
```

FIX RED5.EAR/VIDEOTRANSCODER.WAR/WEB-INF/CLASSES/VIDEOTRANSCODER-WEB.XML

Open red5.ear/videotranscoder.war/WEB-INF/classes/videotranscoder-web.xml remove the following:

```
        <bean id="placeholderConfig"
class="org.springframework.beans.factory.config.PropertyPlaceholderConfigurer">
          <property name="location" value="/WEB-INF/red5-web.properties" />
        </bean>
```

and update the following tags to:

```
  <bean id="videotranscoder-web.context" class="org.red5.server.Context" autowire="byType"/>

 <bean id="web.scope" class="org.red5.server.WebScope" init-method="register">
  <property name="server" ref="red5.server"/>
  <property name="parent" ref="global.scope"/>
  <property name="context" ref="videotranscoder-web.context"/>
  <property name="handler" ref="videotranscoder-web.handler"/>
  <property name="contextPath" value="/videotranscoder"/>
  <property name="virtualHosts" value="*"/>
 </bean>

 <bean id="videotranscoder-web.handler"
  class="com.xuggle.red5.demo.VideoTranscoderDemoAdapter"
  init-method="init">
  <!-- Have the profiler spit out data every 15 seconds -->
  <property name="profilerFrequency" value="15"/>
 </bean>
```

UPDATE APPLICATION.XML

Add the following tag to red5.ear/META-INF/application.xml:

```
<module>
  <web>
    <web-uri>videotranscoder.war</web-uri>
    <context-root>/videotranscoder</context-root>
 </web>
</module>
```

TEST IT

Instructions are also available at http://localhost:8080/videotranscoder/index.html.

- Start JBoss AS 7

- Open browser instance in http://localhost:8080/red5/demos/publisher.html

- On Server tab set Location to **rtmp://localhost/videotranscoder**
- Set Buffer to 0
- Click on Connect to connect to rtmp server

- On Video tab set Device to USB Video Device (or other)
- Click on Start to init the input video device

- On Audio tab set Device to your audio input device
- Click on Start to init the input video device

- On Publish tab set the Name to **example**
- Click on Publish to start streaming

- On View tab set Name to **example**
- Click on Play to play the **example** stream and you should see whatever is being streamed by your video device.

Test Xuggler

- Click on Stop to end playing the stream
- On the view tab, rename the stream to **xuggle_example**
- Click on Play to play the on-the-fly modified stream and confirm that the video is now being mirrored into black-and-white.