

TITLE:

Administrator Manual – Synchronized Program Content Delivery**SOURCE:**

Michigan State University Capstone Team**AUTHORS:**

Kabe VanderBaan	Applied Research Center-Motorola Mobility
Krunal Shah	
Timothy Belcher	
Paul Detkowski	MSU Capstone Team
Tareq Musleh	

DATE:

November 23, 2011
Version 0.1**Revision History**

Version	Date	Change Description
.1	11/23/2011	Initial document
1.0	12/6/2011	Final document

1	Summary	4
2	Build Instructions	4
2.1	Generic Instructions	4
2.1.1	Configuration Files	4
2.1.2	SqlMapConfig Files	4
2.2	Complete TV Applications	4
2.2.1	Android	4
2.3	Build Instructions Problem Solving	7
2.3.1	SqlMapConfig problem	7
2.3.2	Config.properties problem	8
3	Server Configuration	8
3.1	JAVA	8
3.2	Ant	8
3.3	Sun Glassfish v2.1.1 Enterprise Server	8
3.4	Echoprint Server	9
3.4.1	Tokyo Cabinet	9
3.4.2	Tokyo Tyrant	9
3.4.3	Echoprint Server	9
3.4.4	Echoprint Codegen	9
3.5	PostgresQL	10
4	Synchronized Program Content Delivery Configuration	10
4.1	Electronic Programming Guide	10
4.1.1	Database Setup	10
4.1.2	PostgreSQL jar files	10
4.1.3	Glassfish Deployment	10
4.1.4	Glassfish Connection Pools	10
4.1.5	Configuration Test	11
4.2	VodServerDouble , STB, and the TV Portal	11
4.2.1	VodServerDouble service Setup	11
4.2.2	STB and TV Setup	16
4.2.3	Setup of Streaming Network	18
4.2.4	Start Streaming	20
4.3	Content Service	21
4.3.1	Database Setup	21
4.3.2	Application Setup	21
4.4	Identity Management Device Service	22
4.4.1	Database Setup	22
4.5	Sync Service	22
4.5.1	Database Setup	22
4.6	Proxy Service	23

4.7	Mobile Web Application -----	23
4.8	Complete TV Applications -----	23
4.8.1	Android-----	23
1.	“Refresh” button web application.-----	23
5	<i>DLNA Configuration</i> -----	24
5.1	XBMC -----	24
5.2	Control Point -----	24
6	<i>Service API's</i> -----	25
6.1	Electronic Programming Guide -----	25
6.2	VodServerDouble -----	26
6.3	Content Service -----	27
6.4	Proxy Service -----	27
6.5	Identity Management Service -----	28
6.6	Sync Service -----	29
7	<i>Reference Manual</i> -----	29
7.1	Managing Environment Variables -----	29
7.2	Windows Command Prompt -----	30
7.3	Glassfish Deployment -----	30
7.3.1	Autodeploy-----	30
7.3.2	Web Deployment-----	30
7.4	Editing Files inside a Web Archive (WAR) -----	31

1 Summary

This document is intended to facilitate the building, configuration and deployment of the Synchronized Program Content Delivery framework. The following sections explain the build process as well as the configuration needed to run/deploy the Synchronized Program Content Delivery framework.

2 Build Instructions

2.1 Generic Instructions

All programs that are not listed below can easily be built using maven. Navigate into the projects root directory where there is a pom.xml file. From the command line you can simply run the command “mvn clean package” and it will clean previous builds and re-package the war files.

2.1.1 Configuration Files

For programs that require configuration files, the file will be named config.properties. It is located in src/main/resource/config/config.properties. This holds the base uri's of the different services the program interacts with. They should be changed before building. The resources directory is the default directory maven uses for static files that are to be placed in the same place as our .java files. Maven does and will miss non-.java files while packaging a war. See [Config problems](#).

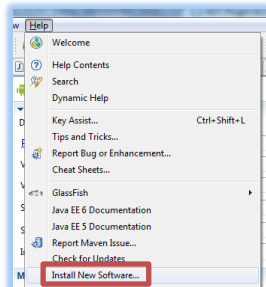
2.1.2 SqlMapConfig Files

Some services such as the User Service and the Content Service require database configuration files used by the java library MyBatis. These are .xml files that come already configured, but minor modification may be necessary. SqlMapConfig holds things like postgres configuration settings and database connection strings that should be changed according to your database. See [SqlMapConfig problems](#).

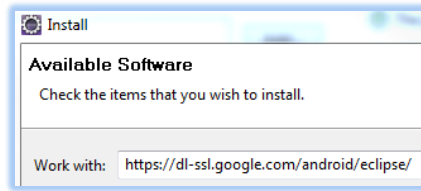
2.2 Complete TV Applications

2.2.1 Android

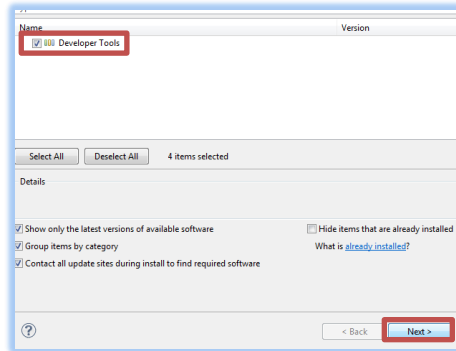
1. Download and install the Android SDK from link below:
http://dl.google.com/android/installer_r10-windows.exe
2. Start Eclipse and select “Install New Software...” under the “Help” menu.



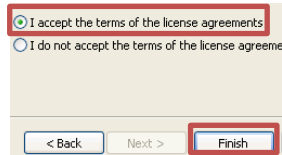
3. Type in <https://dl-ssl.google.com/android/eclipse/> in “Work with:” address bar.



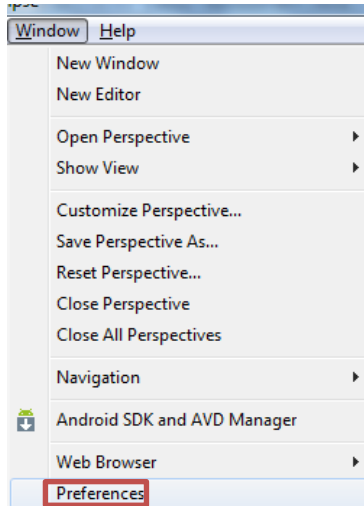
4. Select “Developer Tools” and click “Next >”.



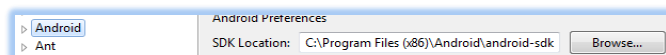
5. Click “Next >” on Install page.
6. Read the license agreements and select “I accept the terms of the license agreements” and click “Finish”.



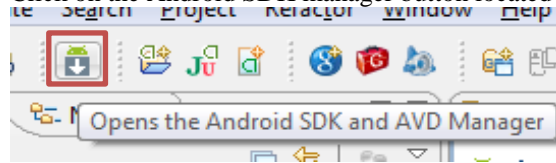
7. Once the Installation finishes, a restart of Eclipse is required.
8. Select “Preferences” under the “Window” menu.



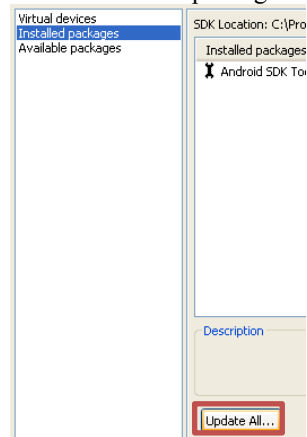
9. Select “Android” in left panel, and type in the correct “SDK Location”, then click OK



10. Click on the Android SDK manager button located on the toolbar.



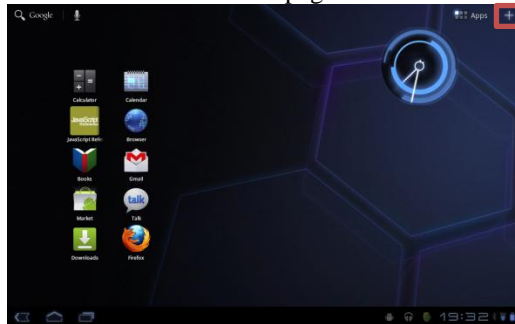
11. Select “Installed packages” in left panel and click on “Update All...”.



12. Click on “Install” to install all packages needed (especially Google Android API 11, SDK Platform Android 3.0 API 11).

13. Turn on the Xoom.

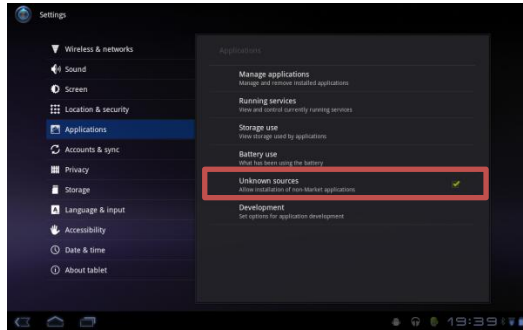
14. On the Xoom’s homepage. Touch “+” in the top right corner.



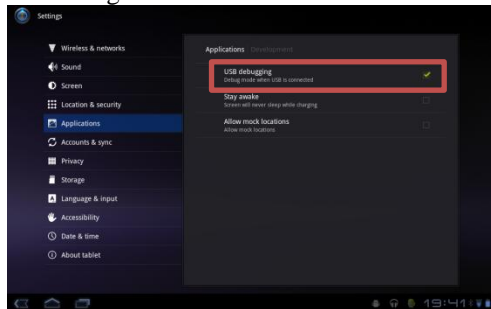
15. Find “Settings” in “App shortcuts” tab and touch it to add this shortcut to homepage.



16. Touch “Settings” and select “Applications”. Check “Unknown sources” to allow installation of non-Market applications.



17. Touch “Development” below “Unknown sources” and check “USB debugging” to enable debug mode on the Xoom.



18. If you’re using Window XP and your Microsoft Media Player is not up-to-date, you may need to install the latest Microsoft Media Player for MTP drivers. If you connect Xoom to your development environment and all the drivers are installed. You can skip this step.
19. Go to the link below to register, download, and install the MOTODEV Xoom usb drivers. http://developer.motorola.com/docstools/USB_Drivers/Handset_USB_Driver/
20. Connect Xoom to the develop environment.
21. Start Eclipse and import the “AndroidCompleteTV” project into workspace.
22. Run “AndroidCompleteTV” project as Android Application. Eclipse will build, install, and run Android Complete TV on Xoom automatically.

2.3 Build Instructions Problem Solving

This section is written for common problems that are run into while building programs. Maven is a powerful tool, but often times can require tweaking.

2.3.1 SqlMapConfig problem

If you run into errors regarding the SqlMapConfig files. You may manually have to copy the xml files into the war. The files should be placed in the same place that classes are located inside a war. Take the files from src/main/resources/conf/MyBatis/ to WEB-INF/classes/conf/MyBatis/* .xml in the war file. If you are unaware of how to edit things within a war file, refer to [Editing War files](#).

2.3.2 Config.properties problem

If it appears as if your configuration files are missing they probably are. You should manually try to copy the files into the correct location and see if the problem is fixed. Copy files from src/main/resources/config/ folder to WEB-INF/classes/config/ folder within the war. If you are unaware of how to edit things within a war file, refer to [Editing War files](#).

3 Server Configuration

The server is where you will be hosting all the software and external programs used by our application. The first step you will have to do is get the Eclipse IDE. You can get Eclipse at <http://www.eclipse.org/>.

3.1 JAVA

Java can be downloaded at : <http://java.sun.com/javase/downloads/widget/jdk6.jsp>. Be sure to set the JAVA_HOME environment variable (ex.

JAVA_HOME=C:\Path\to\Java\jdk1.6.0_20) Check the version numbers, 1.6.0_20 may very well change.

3.2 Ant

Ant is a build tool for java applications. Ant can be downloaded at : <http://ant.apache.org/> Set the ANT_HOME environment variable **ANT_HOME=D:\Path\to\ant-1.8.1** and add that variable to your PATH environment variable. If you are unfamiliar with environment variables please see [Managing Environment Variables](#) .

3.3 Sun Glassfish v2.1.1 Enterprise Server

Glassfish v2.1.1 is the web server that host our applications. Glassfish v2.1.1 depends on Java JDK and Ant. If those are not installed yet, please refer to [Java](#) and [Ant](#). Glassfish v2.1.1 can be installed at <http://glassfish.java.net/downloads/v2.1.1-final.html> . After downloading glassfish and placing the jar in the desired directory you must run a few command line programs. If you are unfamiliar with the windows command prompt see [Windows Command Prompt](#) . Navigate to the same directory that you placed the glassfish jar file in.

1. `java -Xmx256m -jar filename.jar`
2. `cd glassfish`
3. `lib\ant\bin\ant -f setup.xml`

To test the installation start a domain by :

1. open the command prompt if you have not already done so
2. `cd C:\path\to\glassfish\bin`
3. `asadmin start-domain`
4. Open any web browser and navigate to <http://localhost:8080/>
5. At the top of the page You should see the following (or something very close) :

```
Sun GlassFish Enterprise Server v2.1.1 ((v2.1 Patch06)(9.1_02 Patch12)) (build b31g-fcs)
```

Your GlassFish Enterprise Server is now running

Figure 1 : Glassfish Server Welcome

3.4 Echoprint Server

Note: Everything regarding the Echoprint server must be installed on a unix machine (Such as Ubuntu or Mac OS X)

3.4.1 Tokyo Cabinet

1. Download Tokyo Cabinet from <http://fallabs.com/tokyocabinet/tokyocabinet-1.4.47.tar.gz>
2. Save it to your system
3. Open this page to view the installation instructions: <http://fallabs.com/tokyocabinet/spex-en.html#installation>
4. Extract the file to the directory you want it in (/Projects/Apps/tokyocabinet)
5. Follow the installation instructions
 - a. `cd /Projects/Apps/tokyocabinet`
 - b. `sudo ./configure`
 - c. `sudo make`
 - d. `sudo make check`
 - e. `sudo make install`

3.4.2 Tokyo Tyrant

1. Download Tokyo Tyrant from <http://fallabs.com/tokyotyrant/tokyotyrant-1.1.41.tar.gz>
2. Save it to your system
3. Open this page to view the installation instructions: <http://fallabs.com/tokyotyrant/spex.html#installation>
4. Extract the file to the directory you want it in (/Projects/Apps/tokyotyrant)
5. Follow the installation instructions
 - a. `cd /Projects/Apps/tokyotyrant`
 - b. `sudo ./configure`
 - c. `sudo make`
 - d. `sudo make install`
 - e. Now run the server
 - i. `ttserver`
 - f. To check the installation, open another terminal and execute the following
 - i. `cd /Projects/Apps/tokyotyrant`
 - ii. `sudo make check`

3.4.3 Echoprint Server

1. Download the Echoprint Server from <https://github.com/echonest/echoprint-server>
2. Save it to your system
3. Extract the file to the directory you want it in (/Projects/Apps/echonest-echoprint-server)
4. Follow the instructions for "How to run the server"
 - a. `cd /Projects/Apps/echonest-echoprint-server/solr/solr/`
 - b. `java -Dsolr.solr.home=/Projects/Apps/echonest-echoprint-server/solr/solr/solr/ -Djava.awt.headless=true -jar start.jar`

3.4.4 Echoprint Codegen

1. Download the Echoprint codegen from <https://github.com/echonest/echoprint-codegen>
2. Save it to your system
3. Extract the file to the directory you want it in (/Projects/Apps/echonest-echoprint-codegen)
4. Follow the instructions under "Requirements" for installing ffmpeg and taglib if you don't already have them installed
 - a. We are using a Mac, so we ran: `brew install ffmpeg boost taglib`
5. Now make the codegen
 - a. `cd /Projects/Apps/echonest-echoprint-codegen/src/`
 - b. `sudo make`
 - c. `sudo make installPostgresSQL Configuration`

3.5 PostgreSQL

PostgreSQL is the Database Management System that our application uses to store data. PostgreSQL can be downloaded by <http://www.enterprisedb.com/products-services-training/pgdownload#windows> . After running the windows installer and following the on screen instructions., open up pgAdmin to verify that the install was successful.

4 Synchronized Program Content Delivery Configuration

4.1 Electronic Programming Guide

4.1.1 Database Setup

Create TOD database from postgresQL Admin console. From Tools on the main menu of the console, select restore, and select the *TOD.backup* file to create all tables and load data into them. Check the TOD database todasset table to see if data has been successful loaded.

4.1.2 PostgreSQL jar files

Copy postgresql-8.4-701.jdbc4.jar and Configuration-LAB-CoreEdge.jar to your glassfish domains/domain1/lib. Change the configuration files contained in Configuration-LAB-CoreEdge.jar according to your local settings. For example, inside the .jar, you need to change the following line in com_mot_tod_webservices.xml:

```
<p:property id="base_uri" tip="The URI prefix for the CSF's REST web services; the part before
'/tod/v1.'">http://10.2.207.143:8888/WebServices</p:property>
to your localhost assuming your http port is 8888:
<p:property id="base_uri" tip="The URI prefix for the CSF's REST web services; the part
before'/tod/v1.'">http://localhost:8888/WebServices</p:property>
```

4.1.3 Glassfish Deployment

Start your glassfish server and go to http://localhost:4848. From Applications/Enterprise Applications, deploy the CSF-trunk_core_*.EAR file.

4.1.4 Glassfish Connection Pools

On your glassfish admin page, go to Resources/JDBC. Under Connection Pools, create a PostgreSQL pool. Select Postgresql in the Database Vendor box. Glassfish will then automatically select following JDBC driver:

org.postgresql.ds.PGConnectionPoolDataSource

In the Additional Properties table, you need to specify which database the connection pool is connecting to. Following is the example of basic information you need to enter in order to configure the connection pool for Postgresql.

Additional Properties

Name	Value
User	<your user name>
DatabaseName	TOD
Password	<your password>
ServerName	<database server's IP address, localhost for local>
PortNumber	<port number> (default is 5432)

Go into the JDBC Resources page in the glassfish admin console. Create two new JDBC resources, each pointing to the pool that you created in the previous step. Name one "com.mot.tod.db.readonly", and the other "com.mot.tod.db.readwrite".

4.1.5 Configuration Test

Assuming you set up your HTTP port on your glassfish server to be 8080, go to your browser and navigate to <http://localhost:8080/WebServices/tod/v1/admin> and use TOD as the username and RESTful as the passwd to access the admin page and make sure you can see the admin page.

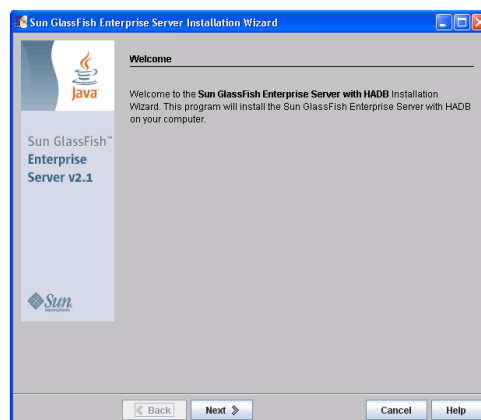
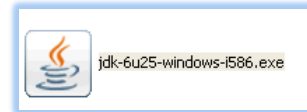
4.2 VodServerDouble , STB, and the TV Portal

The VodServerDouble is a restful web service that allows specific media file(s) stream to the local network by multicasting. The STB will receive the media stream in domain network and display the content of media file on TV. This part will request experience on setting up hardware and software environment. Please concern system administrator for more information.

4.2.1 VodServerDouble service Setup

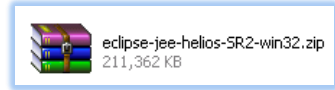
The VodServerDouble will decode and stream the media file(s) to all computers on the local network. The machine running the service has some requirements to allow it to decode and stream media files. First, this machine should at least have enough hardware resources and stable software environment to decode more than one media file simultaneously (Service Performance). Second, this machine will send a large amount of packets through its local network(Network Performance).

1. There are a few compatibility issues when running the service on Windows 7. These instructions will be for a new Windows XP 32 bit machine.
2. Download and install the Java SDK and JRE from link below:
 JDK: <http://download.oracle.com/otn-pub/java/jdk/6u25-b06/jdk-6u25-windows-i586.exe>
 JER: <http://download.oracle.com/otn-pub/java/jdk/6u25-b06/jre-6u25-windows-i586.exe>
3. Download and install VLC from link below:
<http://sourceforge.net/projects/vlc/files/1.1.9/win32/vlc-1.1.9-win32.exe/download>
4. Download and install Sun Glassfish 2.1.1 from link below:
[http://cds.sun.com/is-bin/INTERSHOP.enfinity/WFS/CDS-CDS_SMI-Site/en_US/-/USD/VerifyItem-Start/sges_ee-2_1_1-windows-](http://cds.sun.com/is-bin/INTERSHOP.enfinity/WFS/CDS-CDS_SMI-Site/en_US/-/USD/VerifyItem-Start/sges_ee-2_1_1-windows-ml.exe?BundledLineItemUUID=Ei2J_hCyCdEAAAEvELYR4bY&OrderID=ihCJ_hCyp6AAAAEv8rUIR4bY&ProductID=O61IBe.p09sAAAEk3fsuA36m&FileName=sges_ee-2_1_1-windows-ml.exe)

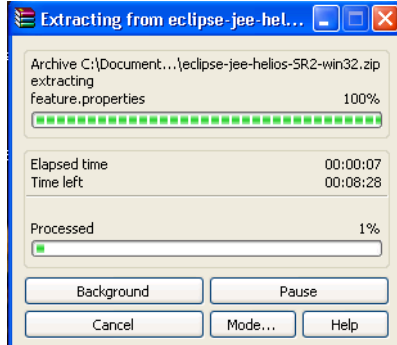


[ml.exe?BundledLineItemUUID=Ei2J_hCyCdEAAAEvELYR4bY&OrderID=ihCJ_hCyp6AAAAEv8rUIR4bY&ProductID=O61IBe.p09sAAAEk3fsuA36m&FileName=sges_ee-2_1_1-windows-ml.exe](http://cds.sun.com/is-bin/INTERSHOP.enfinity/WFS/CDS-CDS_SMI-Site/en_US/-/USD/VerifyItem-Start/sges_ee-2_1_1-windows-ml.exe?BundledLineItemUUID=Ei2J_hCyCdEAAAEvELYR4bY&OrderID=ihCJ_hCyp6AAAAEv8rUIR4bY&ProductID=O61IBe.p09sAAAEk3fsuA36m&FileName=sges_ee-2_1_1-windows-ml.exe)

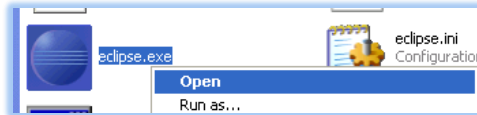
- Download Eclipse IDE for Java EE Developers from link below:
<http://mirrors.med.harvard.edu/eclipse//technology/epp/downloads/release/helios/SR2/eclipse-jee-helios-SR2-win32.zip>



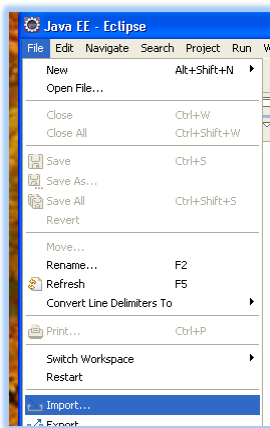
- Unzip Eclipse to a workspace path.



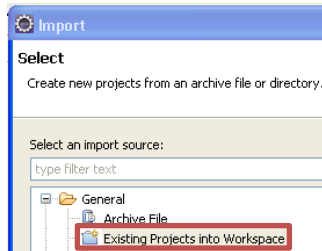
- After unzipping eclipse, right click on “eclipse.exe” and select “open”.



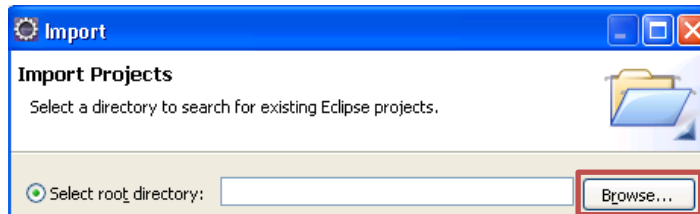
- Select “File” in menu and select “Import”.



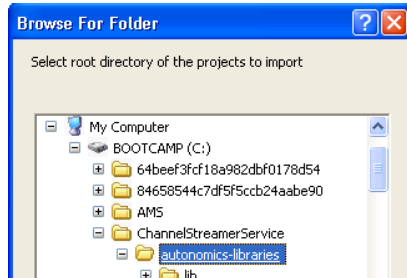
- In the “General” folder select “Existing Projects into Workspace”. And then click “Next”.



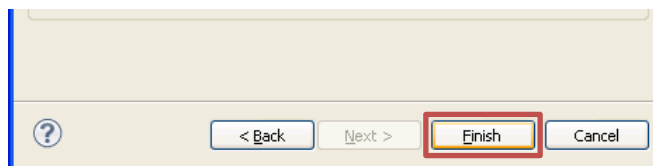
- Click on “Browse...”



- Find and select “autonomics-libraries” directory and click on “OK”.

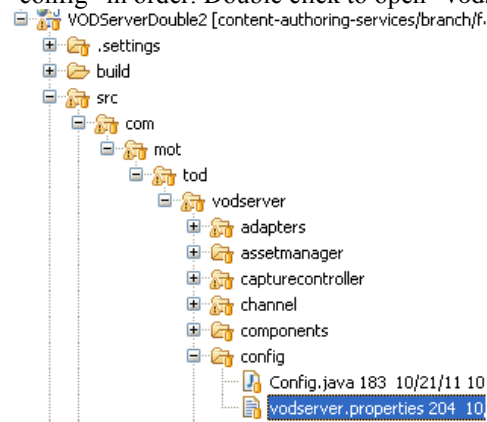


- Click on “Finish” to import the project.



- Repeat step 11 to 15 to import the “VodServerDouble2” Project into workspace. Now you will have two projects in the “Project Explorer”.

- Unfold “VodServerDouble2” project. And Unfold “src”, “com”, “mot”, “tod”, and “config” in order. Double click to open “vodserver.properties”.

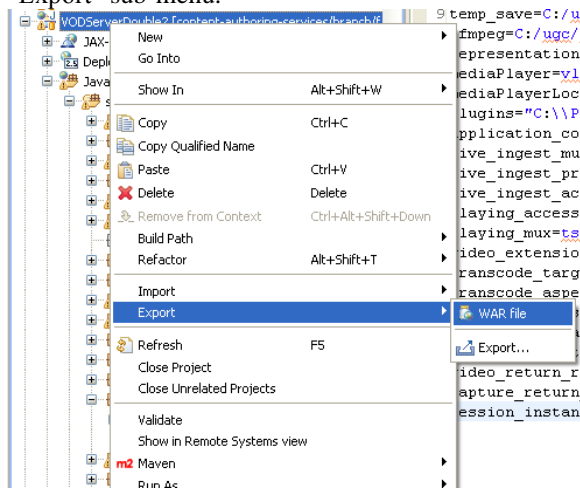


- Change the mediaPlayerLoc to where your VLC is installed and plugins to where your VLC plugins folder is located. For example, if you install VLC at “C:/Projects/Apps/VideoLAN/VLC/you will need to modify to “mediaPlayerLoc = C:/Projects/Apps/VideoLAN/VLC/vlc.exe” and modify “plugins= C:/Projects/Apps/VideoLAN/VLC/plugins”.
- ```

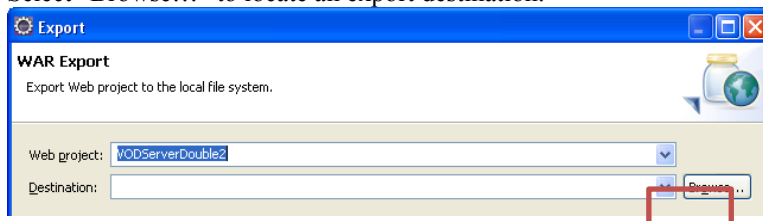
mediaPlayerLoc=C:/Projects/Apps/VideoLAN/VLC/vlc.exe
plugins="C:\\Projects\\Apps\\VideoLAN\\VLC\\plugins"

```

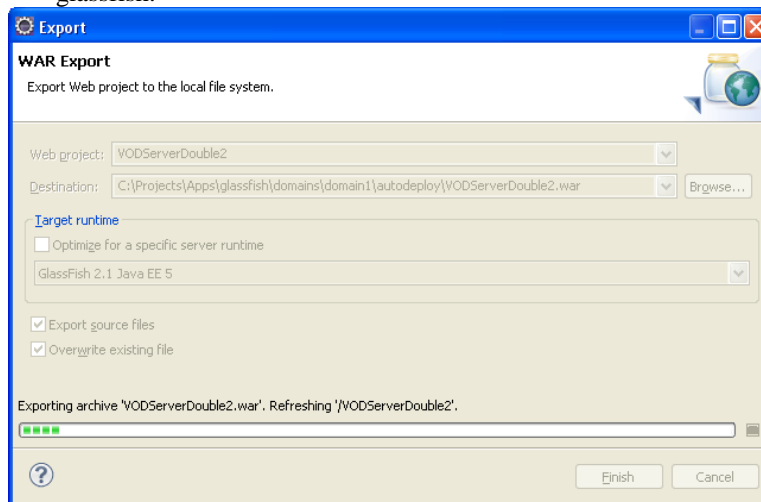
16. Right click on “VodServerDouble2” in Project Explorer and select “WAR file” under the “Export” sub-menu.



17. Select “Browse...” to locate an export destination.



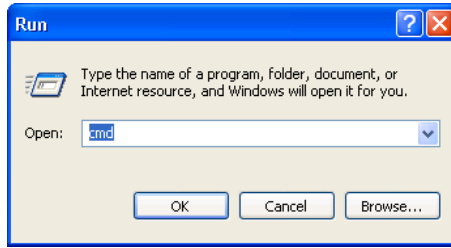
18. Select a location and click “Finish” to generate a WAR file that can be deployed to glassfish.



19. Click on “Run” under Windows “Start” menu.



20. In “Open:”, type “cmd” and click “OK” to run “Command Prompt” in Windows.



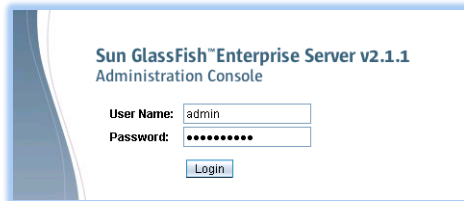
- Use “cd” command to enter the “bin” directory under the “GlassFish”.  
For example: “cd C:\Projects\Apps\glassfish\bin”.

```
C:\Documents and Settings\Team Motorola>cd C:\Projects\Apps\glassfish\bin
C:\Projects\Apps\glassfish\bin>
```

- Type “asadmin start-domain” to start all domains in GlassFish.

```
C:\Projects\Apps\glassfish\bin>asadmin start-domain
Starting Domain domain1, please wait.
Default Log location is C:\Projects\Apps\glassfish\domains\domain1\logs\server.log.
Redirecting output to C:\Projects\Apps\glassfish\domains\domain1\logs\server.log
Domain domain1 is ready to receive client requests. Additional services are being
started in background.
Domain (domain1) is running [Sun GlassFish Enterprise Server v2.1.1 <<v2.1 Patch
06>>(9.1_02 Patch12)> (build b31g-fcs)1 with its configuration and logs at: [C:\P
rojects\Apps\glassfish\domains\].
Admin Console is available at [http://localhost:4848].
Use the same port [4848] for "asadmin" commands.
User web applications are available at these URLs:
[http://localhost:8080 https://localhost:8181].
Following web-contexts are available:
[/web1 /_wstx-services].
Standard JMX Clients (like JConsole) can connect to JMXServiceURL:
[service:jmx:rmi:///jndi/rmi://motorola-media:8686/jmxrmi] for domain management
purposes.
Domain listens on at least following ports for connections:
[8080 8181 4848 3700 3820 3920 8686].
Domain does not support application server clusters and other standalone instanc
es.
C:\Projects\Apps\glassfish\bin>
```

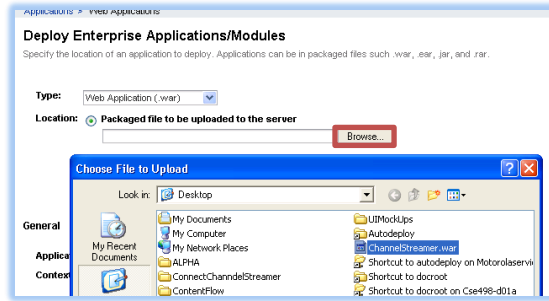
- Open an internet browser and type link <http://localhost:4848> to address bar.
- Use “admin” as user name and “adminadmin” as password to get access to the Server.



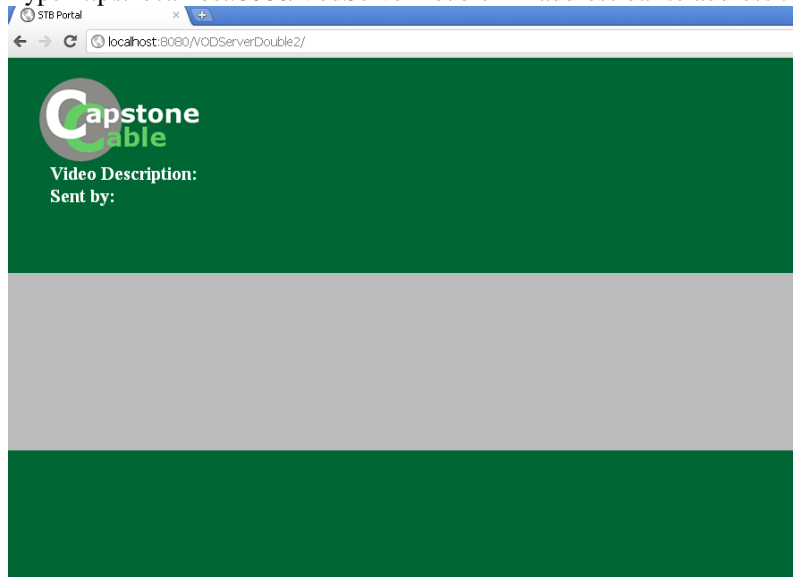
- Select “Web Applications” in left panel and click “Deploy...”



26. Click on “Browse...” and choose the exported WAR file for the VodServerDouble2.



27. Click “OK” in the bottom-right corner to finish the deployment.
28. Select “Web Applications” in left panel and click “Deploy...” again. Click on browse, and this time export the supplied WAR file TVPortal. Click “OK” in the bottom-right corner to finish the deployment.
29. Type <http://localhost:8080/VodServerDouble2/> in address bar to address the test page.



30. Type <http://localhost:8080/TVPortal/> in address bar to address the test page.
31. Select “Web Applications” in left panel and click “Deploy...” again. Click on browse, and this time export the supplied WAR file Interceptor. Click “OK” in the bottom-right corner to finish the deployment.

#### 4.2.2 STB and TV Setup





1. Connect HDMI cable on the STB.



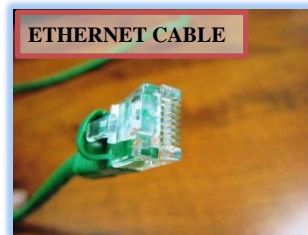
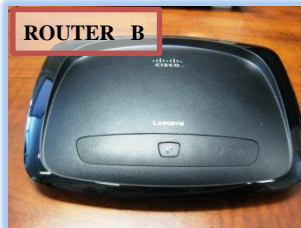
2. Connect HDMI cable to the TV.



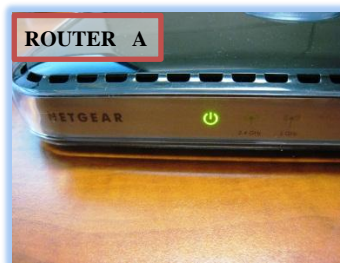
3. Connect both the STB and TV to their respective power cords.



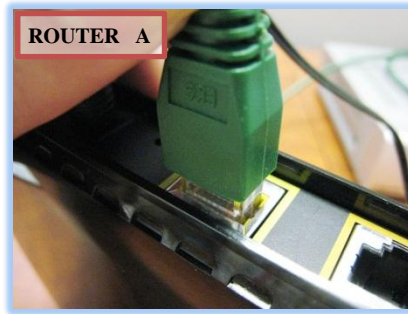
### 4.2.3 Setup of Streaming Network



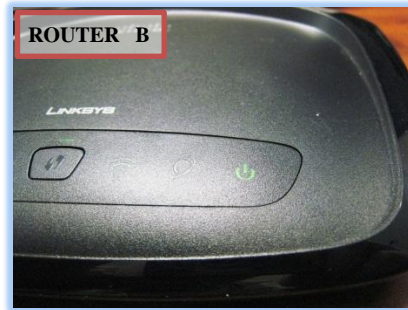
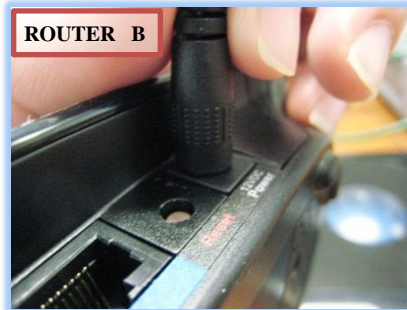
1. We will need at least five CAT-5E Ethernet cable and two routers to connect the VodServerDouble and STB in this architecture. To make things simpler, mark each router with a number or a letter to avoid the confusion. In this case, Netgear Router will be Router A and Linksys router will be Router B.
2. Connect the AC-DC Adapter to Router A.



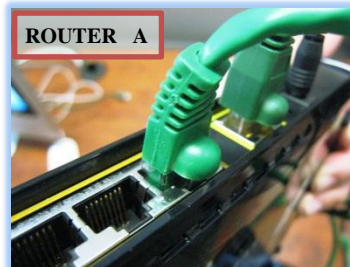
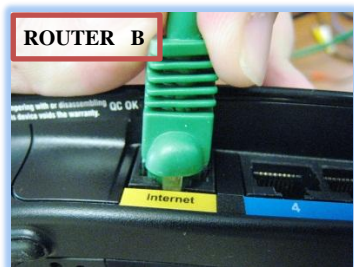
3. Connect Router A to the local network. In this case, connect Internet port of Router A to the switch by Ethernet cable.



4. Connect the AC-DC Adapter to Router B.



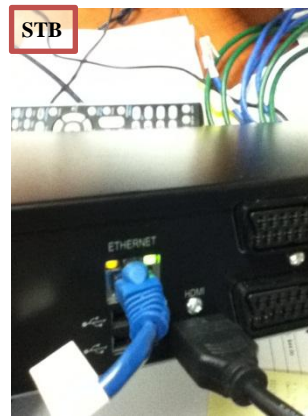
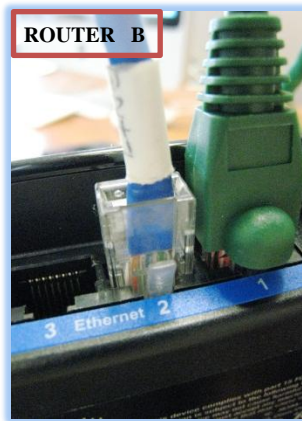
5. Connect Internet port of Router B to Ethernet port of Router A by Ethernet cable.



6. Connect VodServerDouble Service to Ethernet port on Router B.



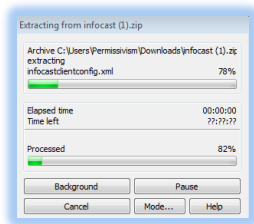
7. Connect STB to Ethernet port on Router B.



8. Set VodServerDouble Service as DMZ service at Router B.

#### 4.2.4 Start Streaming

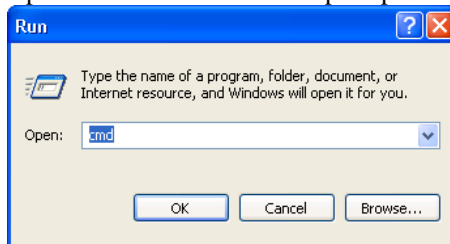
1. Unzip “infocast.zip” to disk on the machine that holds the VodServerDouble.



2. Go to where you unzipped infocast, and open up the “portalurls.xml” file. Modify the first PortalURL to point to the web server that has your TVPortal deployed (use the actual IP address, not localhost).

```
<?xml version="1.0" ?>
<!DOCTYPE PortalURLs SYSTEM "portalurls.dtd">
<PortalURLs>
 <PortalURL>http://192.168.1.9:8080/TVPortal/</PortalURL>
 <PortalURL>file:///usr/local/portals</PortalURL>
</PortalURLs>
```

3. Open the windows command prompt.



4. Use “cd” command to address the unzipped file folder.

```
c:\>cd c:\infocast
c:\infocast>
```

5. Type in “java -jar InfocastServer.jar” to update the channel configuration table to STB.

```
c:\infocast> java -jar InfocastServer.jar
Infocast Server v1.15
(c) 2000-2004 Kreatel Communications AB

2011-04-26 04:58:20 - Reading configuration

2011-04-26 04:58:21 - (224.2.2.13:22222): Transmitting packets at 512 kbit/s
2011-04-26 04:58:23 - (224.2.2.11:22222): Transmitting packets at 2048 kbit/s
2011-04-26 04:58:23 - (224.2.2.2:22222): Transmitting packets at 8 kbit/s
```

6. Turn On STB and TV.
7. When the STB finished loading, press menu on the remote.
8. Navigate to the Kreatel icon, then down, and press OK.
9. The TVPortal should now be visible on the TV screen. See the TV Portal user manual for instructions on how to use the TV Portal.

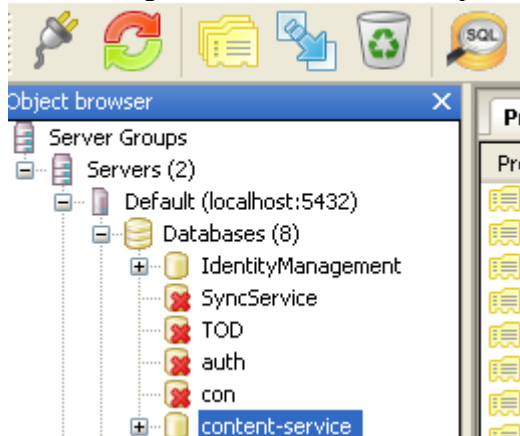
## 4.3 Content Service

The content service is a restful web service that allows all data about secondary content, types, and associations to be stored in a Postgres database. The content service is built on top of the Restlet Framework.

### 4.3.1 Database Setup

Open up the pgAdmin program and connect to your postgres database server (default is localhost:5432).

1. Right click on Databases
2. click New Database ...
3. Create a database ( recommend using name "content-service" for easiest setup )
4. Select the newly created database
5. Navigate to the toolbar at the top and open up the SQL Query browser



6. Click File->Open, select the ContentService.sql (included with this document)
7. Execute the sql script

### 4.3.2 Application Setup

Take the content.war file that is included with this file and deploy it to the glassfish server. For instructions on deploying to a glassfish server refer to [Glassfish Deployment](#). If you changed the name of the database you will need to edit the content.war\WEB-INF\classes\conf\MyBatis\SqlMapConfig.xml to have the proper connection strings (ex. jdbc:postgresql://localhost:5432/content-service ). To edit files within a war file please refer to [Editing Files inside a Web Archive](#).



## 4.4 Identity Management Device Service

The identity management service is a web service that allows all data about users and their devices to be stored in a Postgres database. The identity management service is built on top of the Restlet Framework.

### 4.4.1 Database Setup

Open up the pgAdmin program and connect to your postgres database server (default is localhost:5432).

2. Right click on Databases
3. click New Database ...
4. Create a database ( recommend using name “IdentityManagement” for easiest setup )
5. Select the newly created database

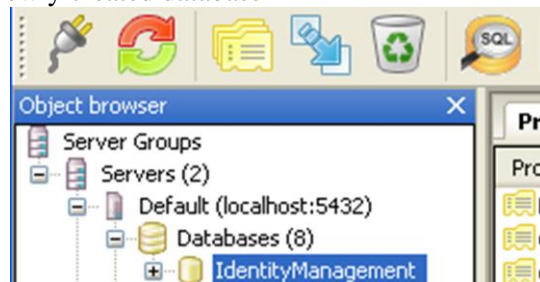


Figure 2 : pgAdmin SQL Browser

6. Navigate to the toolbar at the top and open up the SQL Query browser
7. Click File->Open and select the IDMS.sql (included with this document)
8. Execute the sql script

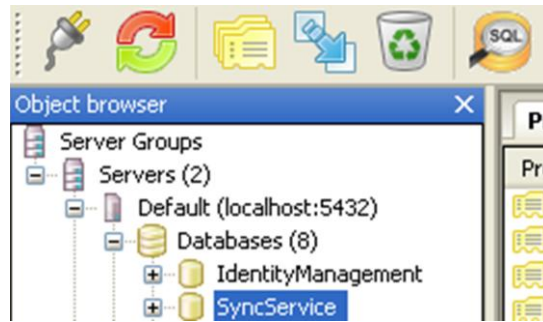
## 4.5 Sync Service

The sync service is a web service that allows all data about sync events and devices to be stored in a Postgres database. The identity management service is built on top of the Restlet Framework.

### 4.5.1 Database Setup

Open up the pgAdmin program and connect to your postgres database server (default is localhost:5432).

9. Right click on Databases
10. Click New Database ...
11. Create a database ( recommend using name “SyncService” for easiest setup )
12. Select the newly created database



**Figure 3 : pgAdmin SQL Browser**

13. Navigate to the toolbar at the top and open up the SQL Query browser
14. Click File->Open and select the SyncService.sql (included with this document)
15. Execute the sql script

## 4.6 Proxy Service

The Proxy Service is built on top of the Restlet Framework. It is the client portal to access the EPG, Content Service and VodServerDouble. You must fix the configuration files used by the proxy. To do this you must change the file proxy.war\WEB-INF\classes\conf\conf.properties so the Base URI's for the Content Service and the EPG point to the correct hose. To edit files within a war file please refer to [Editing Files inside a Web Archive](#). After editing the files you must deploy the war to your glassfish server. For instructions on deploying to a glassfish server refer to [Glassfish Deployment](#).

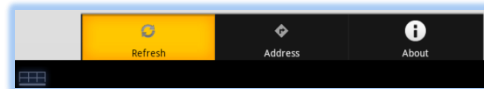
## 4.7 Mobile Web Application

### 4.8 Complete TV Applications

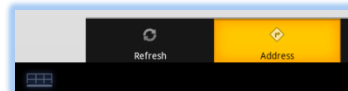
The Complete TV applications are the wrapper applications so that the Mobile Web Application can be put into the Android market and iTunes store.

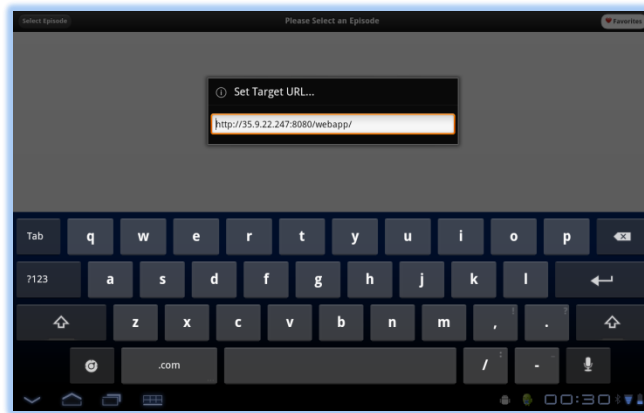
#### 4.8.1 Android

1. "Refresh" button web application.

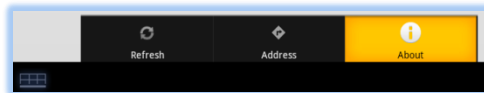


1. "Address" button can change web application address.





2. “About” button contain the Author and Special Thanks.



## 5 DLNA Configuration

### 5.1 XBMC

Download and install [XBMC](#) for whichever operating system you are running. After it's installed, do the following:

- Go to Home -> System -> Network -> Services
- Enable “Share video and music libraries through UPnP”
- Enable “Allow Control of XBMC via UPnP”
- Enable “Allow Control of XBMC via HTTP”
- Change the port to 8056
- Enable “Allow programs on this system to control XBMC”
- Enable “Allow programs on other systems to control XBMC”
- Go to Home -> Videos -> Add Source. If Add Source doesn't appear, make sure that you aren't in Library Mode.
- Browse to the folder containing your .mpg files, press OK
- Change the name for the media source to “Movies”, and press OK
- Press OK again
- Click on “Movies”; you should be able to see the video files in the folder you selected
- Right click the video you want to add to your library, and click “ Manually add to library”
- Change the title to the following format:  
SeriesName\_SeriesNumber\_ProgramName\_AbstractAssetID.mpg
- Select a genre or add a new one, press ok
- Repeat the last 4 steps for all videos you want in your library

### 5.2 Control Point

Control Point is an Android application developed by Motorola. After checking it out from SVN, do the following:



- Open /src/res/values/strings.xml
- Change the “defaultSyncUrl” to whatever URL you use to post sync events to
- Change “defaultSleepPeriod” to the interval (in milliseconds) for posting sync events
- Connect your Android device to the computer via USB, making sure that USB debugging is enabled
- Once this is done, go to Run -> Run As -> Android Application, and select the device you have plugged in.
- You can now use the app from your Android device to connect to XBMC. You may need to make sure that port 8056, which was set in XBMC, is forwarded from the router that machine is connected to. For this, go to <http://www.portforwarding.com/> and find your router, then follow the appropriate instructions.

## 6 Service API's

### 6.1 Electronic Programming Guide

The URI's for accessing the EPG data objects are listed below. The BaseURI for GET is [http://\\$SERVER:\\$PORT/WebServices/tod/v1/](http://$SERVER:$PORT/WebServices/tod/v1/) “, e.g. if you deploy it on localhost with 8080 port, this would be “<http://localhost:8080/WebServices/tod/v1/> “. Motorola is hosting this service now, and it was located at “<http://144.188.54.214:8080/WebServices/tod/v1/>”

- **GET /lineups**  
Retrieves a list of lineups
- **GET /lineup/{lineup\_id}**  
Retrieve the lineup of specified {lineup\_id}
- **GET /lineup/{lineup\_id}/stations**  
Retrieve the stations on the specified lineup of {lineup\_id}
- **GET/lineup/{lineup\_id}/station/{station\_id}**  
Retrieve the station of specified {station\_id} on the lineup of {lineup\_id}
- **GET/lineup/{lineup\_id}/station/{station\_id}/asset**  
Retrieve a list of Assets on the station of specified {station\_id} on the lineup of {lineup\_id}
- **GET/lineup/{lineup\_id}/station/{station\_id}/asset/starttime/{starttime}/endtime/{endtime}**  
Retrieve a list of Assets on the station of specified {station\_id} on the lineup of {lineup\_id} whose starttime and endtime falls into the interval of [{starttime},{endtime}].
- **GET/lineup/{lineup\_id}/epg**  
Retrieve the list of EPGstations of specified lineup of {lineup\_id}. An EPGStation contains a list of EPGAssets. For more information, please refer to the javadoc of com.mot.tod.dto.v1.EPG, com.mot.tod.dto.v1.EPGStation and com.mot.tod.dto.v1.EPGAsset.
- **GET/lineup/{lineup\_id}/epg/starttime/{starttime}/endtime/{endtime}**  
Retrieve the list of EPGStations of specified lineup of {lineup\_id}, which contains EPGAssets whose starttime and endtime fall into the interval of [{starttime},{endtime}]
- **GET /lineup/{lineup\_id}/station/{station\_id}/epg**

Retrieve the list of EPGAssets on the specified station of {station\_id} on the specified lineup of {lineup\_id}.

- **GET /lineup/{lineup\_id}/station/{station\_id}/epg/starttime/{starttime}/endtime/{endtime}**  
Retrieve the list of EPGAssets on the specified station of {station\_id} on the specified lineup of {lineup\_id} whose starttime and endtime fall into the interval of [{starttime},{endtime}].
- **GET asset/{asset\_id}**  
Retrieve the asset of specified {asset\_id}
- **GET epg/asset/{asset\_id}**  
Retrieve the EPG asset of specified {asset\_id}. An EPG asset contains extra information in addition to Asset such as Capture, Visibility, and Classes (TOD, LiveTV, VOD). For more information, please refer to the javadoc of com.mot.tod.dto.v1.EPGAsset.
- **GET /lineup/{lineup\_id}/epg**  
Retrieve a list of all EPGStations, each of which contains a list of EPGAssets, all on the specified lineup of {lineup\_id}
- **GET/lineup/{lineup\_id}/station/{station\_id}/epg**  
Retrieve a list of EPGAssets on the station of specified {station\_id} on the lineup of {lineup\_id}

## 6.2 VodServerDouble

The URI's for controlling the VodServerDouble are listed below. The Base URI is `http://$SERVER:$PORT/VodServerDouble2/`, e.g. if you deploy it on localhost with 8080 port, this would be <http://localhost:8080/VodServerDouble2/>.

- **POST /api/session**  
Create a session with json similar to the following:  
`{"Session": {"ip": "225.1.1.119", "port": "6001", "videoId": "123"}}`
- **POST /api/channel/ip={ip\_address}/port={ip\_port}**  
Starts streaming video id's specified in the request to {ip\_address}:{ip\_port}  
Format: "[videoId1, videoId2, videoId3]"
- **DELETE /api/channel/ip={ip\_address}/port={ip\_port}**  
Stops streaming video id's specified in the request at {ip\_address}:{ip\_port}
- **PUT /api/stream/play/{session\_id}**  
Play the video created in the session call, where the session is specified by {session\_id}
- **PUT /api/stream/pause/{session\_id}**  
Stop the video created in the session call, where the session is specified by {session\_id}
- **PUT /api/stream/stop/{session\_id}**  
Pause the video created in the session call, where the session is specified by {session\_id}

### 6.3 Content Service

- **/secondarycontent/**
  - Create or retrieve all secondary content
- **/secondarycontent/{id}/**
  - Retrieve, edit, or delete secondary content by id
- **/type/**
  - Create or retrieve all content types
- **/type/{id}/**
  - Retrieve, edit, delete or create content types by id
- **/association/**
  - Create or retrieve all secondary content associations
- **/association/{type}/**
  - Retrieve secondary content associations of a particular type
- **/association/{type}/{id}/**
  - Retrieve, edit, delete or create a secondary content association by type and id
- **/stationassoc/{id}/**
  - Retrieve, edit, delete or create a station association by id
- **/stationassoc/**
  - Create or retrieve all station associations
- **/assetassoc/{id}/**
  - Retrieve, edit, delete or create an asset association by id
- **/assetassoc/**
  - Create or retrieve all asset associations
- **/abstractassoc/{id}/**
  - Retrieve, edit, delete or create an abstract association by id
- **/abstractassoc/**
  - Create or retrieve all abstract associations

### 6.4 Proxy Service

All GET api's support JSONP, to receive JSONP you should add the following query term to your request. `?callback=[some javascript function]`.

- **/content/{id}**
  - Retrieve a piece of secondary content according to its id.
  - Variables
    - id – Secondary Content Id – Integer
  - Example :
    - <http://localhost:8080/proxy/content/101>
- **/content/types**
  - Retrieve all the different types of content
  - Example :
    - <http://localhost:8080/proxy/content/types>
- **/associations**
  - Retrieve the associations for a specific TV Program
  - Query Terms :
    - asset – Asset Id – Integer
  - Example :
    - <http://localhost:8080/proxy/associations?asset=59838>
- **/abstractasset/assets**

- Return program metadata for a particular TV show.
- Example :
  - <http://localhost:8080/proxy/abstractassets/asset>
- **/sync/event/user/{userId}**
  - Getting sync events by user.
- **/user/login**
  - Have a user login
  - Variables
    - username – UserName – String
    - password – Password - String
  - Example :
    - <http://localhost:8080/proxy/user/login?username=boydale1&password=apple123>

## 6.5 Identity Management Service

- **GET /user/{userId}/devices**  
Get all the devices registered to a user
- **GET /user/{userId}/device/{deviceId}**  
Get the association between a {userId} and {deviceId}
- **PUT /user/{userId}/device/{deviceId}**  
Update the association between a {userId} and {deviceId}  
{“deviceId”:”123”,”userId”:”12”,”alias”:”Bedroom STB”}
- **DELETE /user/{userId}/device/{deviceId}**  
Delete the association between a {userId} and {deviceId}
- **GET /user/{userId}/devices**  
Get all associations for a {userId}
- **POST /user/{userId}/devices**  
Add an association for a {userId}  
{“deviceId”:”123”,”userId”:”12”,”alias”:”Bedroom STB”}
- **DELETE /user/{userId}/devices**  
Delete all associations for a {userId}
- **POST /user**  
Create a new user  
{“username”:”test”, “password”:”foo”, “firstName”:”Tim”, “lastName”:”Belcher”, “createdAt”:”(timestamp)”, “emailAddress”:”[test@test.com](mailto:test@test.com)”}
- **GET /user/{userId}**  
Get a user’s information based on the {userId}
- **DELETE /user/{userId}**  
Delete a user
- **PUT /user/{userId}**  
Update a user’s information

```
{“username”:"test", “password”:"foo", “firstName”:"Tim", “lastName”:"Belcher",
“createdAt”:"(timestamp)", “emailAddress”:"test@test.com"}
```

- **POST /login**  
Post the username and password to allow the user to login  
Ex: {“username”:"username", “password”:"password"}
- **POST /device**  
Post the device’s information to create a device  
{“type”:"STB”, “provider”:"Comcast”, “manufacturer”:"Motorola”, “physicalId”:"12:  
23:45:67:ab”, “model”:"Kreatel-5163"}
- **PUT /device/{deviceId}**  
Update information for the device  
{“type”:"STB”, “provider”:"Comcast”, “manufacturer”:"Motorola”, “physicalId”:"12:  
23:45:67:ab”, “model”:"Kreatel-5163"}
- **GET /device/{deviceId}**  
Get information for the device based on the {deviceId}
- **GET /device/id/{physicalId}**  
Get information for the device based on the physicalId in the database
- **DELETE /device/{deviceId}**  
Remove a device based on the {deviceId}

## 6.6 Sync Service

- **GET /device/{deviceId}**  
Get the most recent sync event for the device with id={deviceId}
- **POST /syncevent**  
Post a sync event to the sync service directly  
{“deviceId”:"12:34:56:68:ab”, “abstractAssetId”, “12345”, “abstractAssetOffset”, “  
45543”, “syncEventType”:"PLAY”, “timestamp”:"(timestamp)", “userId”:"123"}
- **GET /user/{userId}**  
Get the most recent sync event for each device registered to user with  
id={userId}
- **POST /user/{userId}/device/{deviceId}**  
Post {deviceId} and {userId} to the sync\_request table for logging purposes. No  
json is required in the request.

## 7 Reference Manual

This manual provides you with useful instructions on tasks that need to be done multiple times throughout the configuration process.

### 7.1 Managing Environment Variables

1. Right-click Computer, then click Properties.

2. On the left side click Advanced system settings.
3. In the advanced tab select environment variables.

## 7.2 Windows Command Prompt

1. Click on the Windows start menu
2. Type cmd.exe in the search box
3. Click and open the cmd.exe program

## 7.3 Glassfish Deployment

You will need to add a few extra libraries in {glassfish\_home}/lib

- asm-3.1.jar
- activemq-all-5.5.1.jar
- camel-core-2.8.1.jar
- camel-jms-2.8.0.jar
- cglib-2.2.2.jar
- commons-logging-1.1.1.jar
- log4j-1.2.14.jar
- postgresql-9.0.801.jdbc4.jar
- slf4j-log4j12-1.5.11.jar
- spring-asm-3.0.1.RELEASE.jar
- spring-beans-3.0.1.RELEASE.jar
- spring-context-3.0.1.RELEASE.jar
- spring-core-3.0.1.RELEASE.jar
- spring-expression-3.0.1.RELEASE.jar
- spring-jms-3.0.1.RELEASE.jar
- spring-tx-3.0.1.RELEASE.jar

You may also need to remove jackson-asl.jar

Below are instructions on deploying a war file to a glassfish server. There are two different options which are both explained below.

### 7.3.1 Autodeploy

Simply copy the war file into your domains autodeploy directory. (ex. C:\path\to\glassfish\domains\domain1\autodeploy).

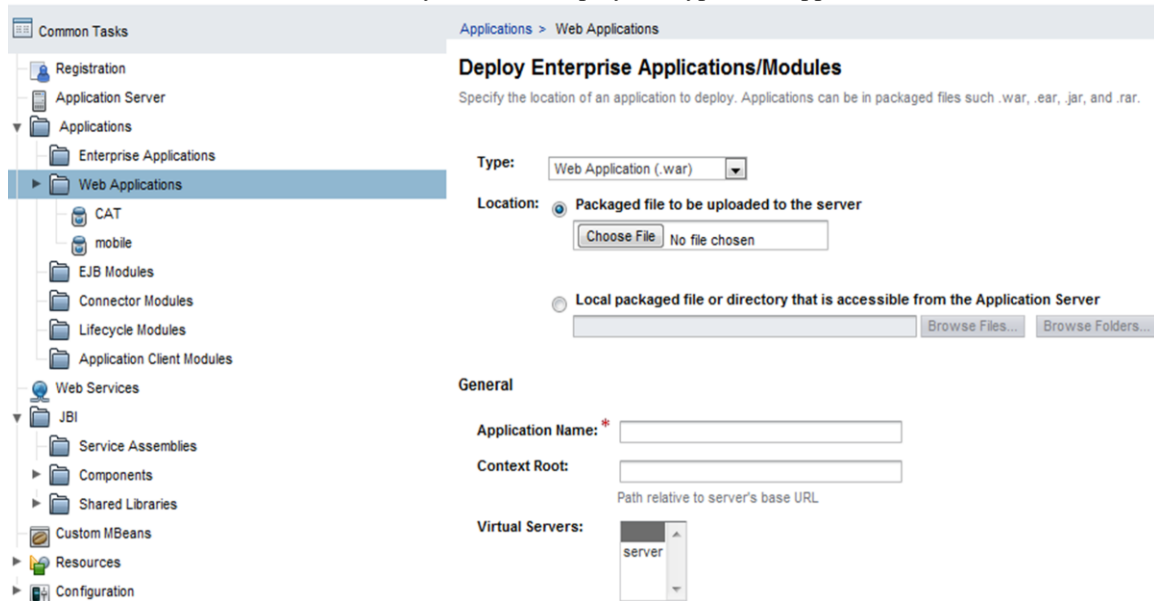
Another easy way to do this when using maven is to edit your pom.xml file. Add the following to your plugins:

```
<plugin>
 <groupId>org.apache.maven.plugins</groupId>
 <artifactId>maven-war-plugin</artifactId>
 <configuration>
 <warName>CompleteTVWebApplication</warName>
 <failOnMissingWebXml>>false</failOnMissingWebXml>
 <outputDirectory>autodeploy/dir</outputDirectory>
 </configuration>
</plugin>
```

### 7.3.2 Web Deployment

1. Navigate to the admin interface for your glassfish installation by going to <http://localhost:4848/login.jsf> .
2. Default Username is admin and default password is adminadmin.
3. Click Applications->Web Applications

4. Click deploy
5. Choose the war file you wish to deploy and type in an application name



**Figure 4 : Glassfish Web Deployment**

## 7.4 Editing Files inside a Web Archive (WAR)

1. Change the file extension of the WAR file to a ZIP extension.
2. Extract all the files
3. Edit files
4. ZIP up the file
5. Change the file extension of the ZIP file to a WAR extension