Quick Start

NOTE: Linux users refer to this page: Build Environment - Linux

Preparing a Windows System for Build

1. Get CableLabs known working cygwin install: cygwin-minimal-20120803.zip
2. Install cygwin from local directory, using the CableLabs known working cygwin. See Build Environment - Cygwin MinGW.
3. Download and install appropriate JDK: http://www.oracle.com/technetwork/java/javase/downloads/index.html. See "Windows Build Tools" section of Build Environment - Cygwin MinGW.
4. Download and install appropriate Ant (version 1.8.1 or later is required): http://ant.apache.org/bindownload.cgi. See "Windows Build Tools" section of Build Environment - Cygwin MinGW.
5. Add the paths to the JDK and Ant to your path, e.g. export PATH=$PATH:/cygdrive/c/apache-ant-1.8.1/bin:/cygdrive/c/Java/jdk1.6.0_20/bin).
6. Checkout source from repository. In an appropriate sub-directory:

```
> svn checkout -N https://community.cablelabs.com/svn/OCAPRI
> cd OCAPRI
> ./checkout_dev_env.sh trunk
```

> Note: You may need to set environment variables SVN_USER and SVN_PASSWORD in order to avoid svn authentication challenges. If you do not have commit access for the repository, use SVN_USER=cruise, SVN_PASSWORD=cruise

> Note: It may be necessary to temporarily disable your virus protection software while downloading files. Remember to enable virus protection after svn checkout is finished.

7. Set environment variables. A trunk/setEnv file is generated that has the necessary environment settings. Either add these settings to your profile or source the setEnv file as follows:

```
> cd trunk
> source setEnv
```

> NOTES: setEnv does not set JAVA_HOME or ANT_HOME - you need to do that separately e.g., in ~/.profile export JAVA_HOME=c:/Java/jdk1.6.0_13 export ANT_HOME=c:/apache-ant-1.8.0 -You need an appropriate command line subversion client in your path and you may need to repeat the script several times due to issues with cygwin port of svn.

Building Everything for the First Time

1. Build the platform (the support code for the OCAP stack). Start a new terminal session to pick up the environment changes you just made

```
> cd $PLATFORMROOT
> make clean purge build
```

2. Build the OCAP stack (the Java VM and support libraries):

```
> cd $OCAPROOT
> ant clean purge build
```

From this point forward, `ant purge` should work. You can do a super-clean build using `ant clean purge build`.

See Building Overview for additional information.

Running the RI

Before running the RI you may need to make some configuration changes:

- It is very likely that you will need to disable any firewall on your development machine as the RI uses various ports and processes to handle streaming, etc.
- Depending on your particular video hardware/driver configurationit is likely that you will need to turn off hardware acceleration for your display.
    - (Control Panel->Display->Settings-> Advanced->Troubleshoot (Windows XP)

Run the RI:

1. Run the RI with the TuneTest xlet to verify that the RI is installed and functioning as expected.

```
> cd $PLATFORMROOT
> ./runRI.sh -tunetest -setup
```

Note: On a fresh checkout or binary install the initial tune test may fail. If this occurs Ctrl-C to stop the RI and repeat "./runRI.sh -tunetest". The tune should now work. For more information on TuneTest please see Running Tune Test.

Another Note: If you have not run vlc on your machine you may need to run it interactively in order to answer an interactive prompt. To do so:

```
> cd <wherever you installed it ocap-r>i/trunk/common/resources/Win32/VLC
> ./vlc.exe
```

vlc will start interactively (you may need to find the window). After you have closed the window vlc will start without the prompt.

So... now what?

To get **your** Xlet running with the RI, look at $OCAPROOT/bin/$OCAPTC/env/hostapp.properties.

There is also a wealth of QA Integration Xlets available (some may not work at the moment - YMMV). See Running QA Apps on the RI and Integration Xlet Testing for more information.

Other useful information:

RI PC Platform Logging (Quick Start)