

Reference Manual for the OCAP-RI

Edited by Chris Sweeney

Introduction

Welcome to the OpenCable Reference Implementation (aka OCAP-RI) Reference Manual. These pages are intended to provide a more coherent organization of current project information, and to "fill in the gaps" where applicable.

The OCAP-RI is one component of the OpenCable "triad" - the other two being specification(s) and the CTP (Certification Test Program) suites. As such, one of the major intended uses of the OCAP-RI is to help discover and resolve inconsistencies, deficiencies and ambiguities in specification and certification. Another intended use is to support platform and stack developers in understanding and implementing OpenCable compliant systems. Finally, the OCAP-RI in combination with the <tru2way> Workbench SDK (aka TWB - an Eclipse feature/plugins) is intended to provide application developers with an OCAP-specific IDE.

Licensing

The OCAP-RI is available under a triple license model - BSD 2-clause, GPLv2 for non-commercial use (Open Source) and a parallel commercial license. Please refer to https://java.net/projects/ocap-ri/pages/FAQ_Licensing for more detail.

Getting Started

- For Xlet developers.
 - Download and install an appropriate revision of the OCAP-RI. Instructions can be found [here](#).
 - Download and install Eclipse.
 - Install <tru2way> SDK - instructions are contained here: http://ri.opencable.com/sdk/updates/eclipse.3.6/stable/latest/TWB_ReleaseNotes.pdf.
 - Follow instructions in the TWB QuickStart Guide: <http://ri.opencable.com/sdk/updates/eclipse.3.6/stable/Rel6/TWBQuickStart.pdf>.
 - Details on Xlet architecture, design and deployment are beyond the scope of this manual. However, many resources are available on the internet.
- For Stack (Java) developers.
 - Both Stack and Platform developers will need to create/install/build a source development environment Instructions are here:[Quick Start](#)
 - Developers may choose to use Eclipse for development, debug. Instructions are here:[Configuring Eclipse for OCAP stack debugging](#)
- For Platform (C) developers.
 - Both Stack and Platform developers will need to create/install/build a source development environment Instructions are here:[Quick Start](#)
 - Developers may choose to use Eclipse for development, debug. Instructions are here: [Eclipse Platform Configuration](#)

Running/Testing the OCAP-RI

There are several mechanisms available for validating that a given OCAP-RI build is functional.

- For a quick test try running the TuneTest:

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

For more details about TuneTest see [Tune Test Instructions](#).

- If you have access to CableLabs CTP respository, a more comprehensive test suite is the "Smoke Test".
- CableLabs uses Cruise Control <http://cruisecontrol.sourceforge.net/> for continuous integration and for nightly builds. Additional information is at [CableLabs Cruise Control Configuration](#).
- runRI.sh has several options that provide test capabilities. Documentation: [Using runRI.sh](#)

Tips and Tricks

- [Configuring Java stack logging](#)
- [Profiling the RI's JVM using NetBeans 6.8](#)
- [Generating heap dumps on the RI's JVM using CVM Inspector](#)
- [Generating thread monitor stats on the RI's JVM using hprof](#)
- [Configuring RI to use libmpeg2 instead of ffmpeg for video decoding](#)
- [Running QA Apps on the RI](#)

Contributing to the OCAP-RI

[Contribution Tracking Log](#)

Architecture, Design, and Other Such Stuff

[RI PC Platform Overview](#)