

"How It Works" Design Documentation

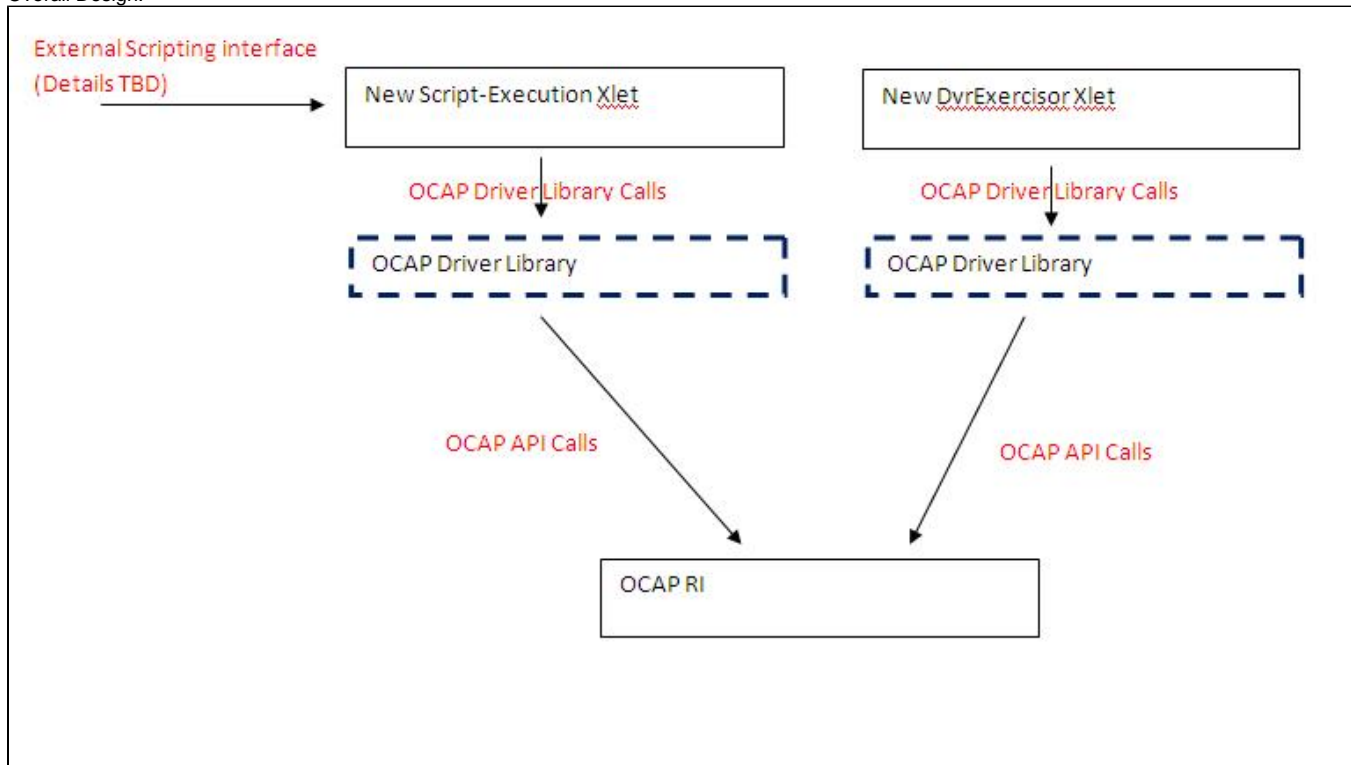
Project Goals:

- Phase 1 Goal: Rework DVRExerciser into a more maintainable form by
 - Splitting out scripting capability into a separate xlet
 - Aggregating OCAP calls into commonly used actions (e.g., Tune, Record, etc) to be used by both the new RiExerciser (replaces DvrExerciser) xlet and the new Scripting xlet.
 - Restructure new RiExerciser to have same functionality as DvrExerciser except utilizing aggregated function calls instead of calling OCAP APIs directly.
- Phase 2 Goal:
 - Increase scripting capability by adding functionality to perform a wider variety of tests via scripts.
 - What else?

Design Goals:

- Develop library of commonly used functions (e.g. Tune, Record, etc) that can be used by xlets or scripts. The idea is to make this library easier to use than the raw OCAP calls so that a script-writer in particular does not need detailed OCAP API knowledge to write a script. This library is called the OcapAppDriver.
- OcapAppDriver lib calls need to clearly assert whether they are synchronous or asynchronous. In general, synchronous calls will be favored over asynchronous.
- OcapAppDriver lib calls will pass/return only simple Java primitives (String, int, etc) in keeping with the philosophy that the API should require minimal OCAP API knowledge to use.
- Callbacks...
- The Scripting Xlet needs to expose a scripting capability that allows a reasonable level of scripting functionality (e.g. variables, if/else statements, for loops, etc).

Overall Design:



Driver Lib Detailed Design

For details concerning the OcapAppDriver methods, consult the OcapAppDriver Javadoc.

Scripting Xlet Detailed Design

The Script Xlet will provide a means of executing custom scripts either at Xlet start-up or via an external interface whose details still need to be worked out. The current scripting capability in the DvrExerciser is limited and mostly simulates key-presses on the DvrExerciser.

In order to provide a fully functional scripting interface, the Script Xlet will use a different approach. BeanShell will be used as a script interpreter. BeanShell is an open-source (LGPL) Java interpreter. It processes scripts written in Java syntax (without needing a class and main method) and can call external method (written in Java). Using BeanShell, we gain

- The common syntax constructs such as if-else, for loops, variables, etc.
- The ability to call OCAP Driver Lib methods without the need for any wrapper code.

- BeanShell's use of Java syntax allows a user with essentially no Java knowledge to write basic scripts, while at the same time allowing more advanced scripts to be written by a Java-savvy user.

More design details TBD...

RiExerciser Xlet Detailed Design

The RiExerciser will provide a "Guide" Type OCAP Xlet which can be utilized to support functional/application level testing. The RiExerciser will utilize the OcapAppDriver Library rather than calling OCAP APIs directly. This will ensure that the functionality used in RiExerciser will be incorporated as part of the framework and included as part of automated testing to minimize regressions from being introduced. Another goal of the refactoring of DvrExerciser to better structure the RiExerciser into a Model View Controller Architecture. In Phase 0 & 1, the existing UI format and layout will be used as is, but the underlying structure to support this will be refactored into the new framework, especially in terms of separation between Model View and Control. More information on the UI can be found at <https://community.cablelabs.com/wiki/pages/viewpage.action?pageId=39846066>