

OCAP 入門: 第 1 回 OCAP RI を実行する

概念とアプリケーション

Jonathan Knudsen

2010年 4月 05日
(初版 2010年 3月 02日)

OCAP RI (OpenCable Application Platform Reference Implementation) は、セットトップボックスやテレビ、その他の機器で実行する双方向アプリケーションのための包括的プラットフォームの実装であり、無料で入手することができます。この記事では [3 回連載の第 1 回](#) として、OCAP の概要と RI (Reference Implementation: リファレンス実装) の入手方法および実行方法について説明します。また、OCAP RI がどのようにアプリケーションを発見し、どのようにアプリケーションを実行するのかについても学びます。

[このシリーズの他の記事を見る](#)

2010年 3月 18日: 著者からの要請で hostapps.properties をすべて hostapp.properties に変更しました。

OCAP (OpenCable Application Platform) はセットトップボックスやテレビを対象とした、ケーブルテレビ業界のプラットフォームであり、ハードウェアに依存しない標準的なベースを双方向アプリケーションに提供します。OCAP を利用して開発されたアプリケーションは、ケーブルテレビの利用者にさまざまな機能を提供することができます。OCAP は消費者向けに、tru2way という名前を持っています。tru2way の使用ライセンスを持つ企業には、家電市場における名だたる企業が数多く名を連ねています ([「参考文献」](#) を参照)。

OCAP 規格は CableLabs という業界コンソーシアムによって作成されました。この OCAP のドキュメントは CableLabs の Web サイトに公開されており、また関連する何十もの規格も公開されています ([「参考文献」](#) を参照)。OCAP 1.1.2 文書は 667 ページもあり、優れた規格の例に漏れず、他の多くの文書をベースに作成されています。このプラットフォームは何と言っても複雑であるため、これから OCAP の開発を始めようとする人にとっては、どこから手を付ければよいのか簡単にはわかりません。この 3 回連載の記事の目的は、この OCAP プラットフォーム用のアプリケーション開発に関心を持つ Java™ 開発者が効率的に作業を開始できるように、OCAP の RI (Reference Implementation: リファレンス実装) を紹介することです。今回の第 1 回では、OCAP について詳しく学ぶとともに、RI をダウンロードしてインストールし、OCAP RI の上でいくつかのアプリケーションを実行します。 [第 2 回](#) では、コマンドラインまたは Eclipse IDE を利用して、独自の OCAP アプリケーションを作成する方法について説明します。最後に第 3 回では、OCAP の内

部構造について説明し、皆さん自身で OCAP RI をビルドするための方法を説明します。また第 3 回では OCAP RI のソース・コードについても概要を説明します。

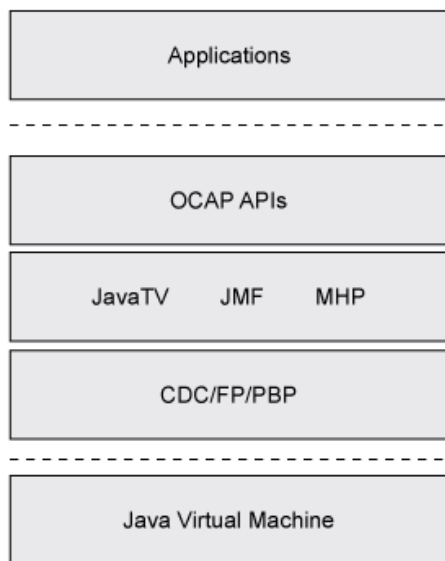
注: 現在、OCAP RI は Microsoft® Windows® 用のみが配布、サポートされています。しかし第 3 回で説明するとおり、Linux® で RI をビルドすることも可能です。

OCAP スタック

アプリケーションの観点から見る場合には、OCAP を API の集合と見なすと便利です。OCAP には、画面上に描画するためのグラフィックス API、テレビのチャンネルを検出、表示するためのチューナー API、その他数多くの API が含まれています。しかし、OCAP を API のスタックと見なしてしまうのは、便利ではあるものの簡略化しすぎています。OCAP はそうした API 以外にも、さまざまな振る舞いについて規定しています (例えば 1 台の機器がチューナーを最低いくつ持つ必要があるのか、あるいは OCAP 機器はどれだけのディスク容量を備えている必要があるのかなど)。

API に関して言えば、OCAP は標準的な Java ME 技術の上に構築されています。図 1 は OCAP のスタックを示しています。

図 1. OCAP のスタック



ブルーレイに関して

Java 技術をベースとする別のスタックが既にテレビの世界の主流として採用されています。それは、すべてのブルーレイ・プレーヤーで使われている、BD-J というソフトウェア・スタックで OCAP スタックの仲間です。OCAP と BD-J はどちらも、CDC/FP/PBP をベースにしており Java TV などの API を持っています。ただし全体として見ると、この 2 つのスタックは同じではありません。そのためほとんどのアプリケーションは、変更を加えない限り、両方のスタックで実行することはできません。

OCAP スタックには長い歴史があり、それは多くの異なる部分で構成されていることを意味します。OCAP スタックには DAVIC や HAVi などの名前が含まれ、また図 1 に示すようなコンポーネントも含まれています。

- Java TV は Sun Microsystems によって開発された標準です。

- JMF は Java Media Framework であり、動画やその他のメディアを描画するための API です。
- MHP (DVB Multimedia Home Platform) も、テレビに関する別の標準です。
- CDC は Java ME の Connected Device Configuration です。CDC は Java 技術スタックのベースとなっています。デスクトップ・コンピューター用の Java 技術が大規模であると考え、携帯電話用の Java 技術を小規模であると考えれば、CDC は中規模です。CDC が対象としているのは、テレビのセットトップボックスや車載用コンピューター、そしてその他の中規模のコンピューターです。
- FP (Foundation Profile) と PBP (Personal Basis Profile) では、CDC にさらなる API が追加されます (ユーザー・インターフェースやグラフィックスのための API など)。

CDC アプリケーションと標準的なデスクトップ Java アプリケーションとの重要な違いは、アプリケーションのモデルにあります。CDC アプリケーションは `xlet` です。Xlet はアプレットや (エンタープライズ Java 技術での) サブレット、または (モバイル Java 技術での) MIDlet のように、ライフサイクルが管理されています。

OCAP を実装する

厳密に言えば、OCAP は単なる規格にすぎないため、OCAP の実装は機器メーカー次第です。この実装プロセスを容易にするために、RI が用意されています。

OCAP RI は OCAP スタックの実装例であり、さまざまな機器に適用可能な実動スタックの実装として、OCAP スタックを実装する際の出発点にすることができます。このため、機器メーカーは実装作業が容易になります。また OCAP RI は実動 OCAP スタックとしてデスクトップ・コンピューター上で実行することも可能です。デスクトップ・コンピューターは、アプリケーション・プログラマーが実装したアプリケーションを実際の機器にデプロイする前に実行、デバッグするのに非常に適しているため、デスクトップ・コンピューター上でアプリケーションを実行することでアプリケーション・プログラマーの作業が楽になります。

OCAP RI は多種多様な人達に利用されるため、いくつかの形式で提供されています。

- 単純なインストーラーを利用する場合、Windows コンピューターにバイナリーがインストールされます。OCAP スタックが実行されると、擬似的なテレビ画面、フロントパネルの表示、リモコン、そしてコンソール出力が表示されます。このインストーラーは最小限のセットアップで OCAP スタックを実行したい場合に便利です。この記事では、このインストーラーを使用します。
- Eclipse プラグインを利用する場合、Eclipse IDE を使用して OCAP アプリケーションを作成することや、それらのアプリケーションを OCAP RI 上でテストすることができます。この Eclipse プラグインについては、この連載の [第 2 回](#) で学びます。
- OCAP RI を自分でビルドしたい人には、完全なソース・コードが用意されています。実際の機器に OCAP を実装しようとしている人達や、大規模なソース・コードを読むのが楽しい人達には、この完全なソース・コードが適しています。この連載の第 3 回では OCAP RI をビルドする方法について説明します。

OCAP の入手方法と実行方法

まず、Windows 用の [インストーラーをダウンロード](#) します。この記事の執筆時点での最新バージョンは 1.1.3RelA です。このリンクでうまくいかない場合には、[OCAP RI](#) のページに行き、OCAP RI のリリースへのリンクをたどってください。

このインストーラーは約 155MB もある巨大なファイルです。ダウンロードしたら、このインストーラーを実行し、表示に従ってインストールを進めます。数多くの選択をする必要があるわけではありませんが、どうすべきか迷った場合には、OCAP RI プロジェクトのウィキのページを調べてみてください(「[参考文献](#)」を参照)。

インストーラーの実行が完了すると、デスクトップ上にアイコンが表示されます。このアイコンをダブルクリックすると、コマンドライン・ウィンドウがポップアップ表示され、そのすぐ後に OCAP RI のウィンドウが表示されます (図 2)。

図 2. 何も手を加えない状態の OCAP RI



このウィンドウは CirclesXlet というアプリケーションを実行するように設定されています。ここではこの CirclesXlet を、ボール・アプリケーションと呼ぶことにします。まずこのアプリケーションで遊んでみます。リモコンのキーをクリックすると、ボールの数とサイズを変更することができます。

OCAP RI を無効にする方法

OCAP RI を終了するためには、コマンドライン・ウィンドウを見つけて選択し、Ctrl-C を押します。すると、ウィンドウには「Terminate batch job (Y/N)? (バッチ・ジョブを終了しますか)」という確認が表示されます。「n」を入力して Enter を押します。すると OCAP RI のウィンドウとコマンドライン・ウィンドウの両方が消えます。

「n」を入力する理由は、OCAP RI が VLC (「[参考文献](#)」を参照) を使用しているためです。VLC はインターネット上に存在する珠玉のツールの 1 つであり、非常に優れたオープンソース・アプリケーションです。VLC を使うことで、動画を再生したり、トランスコードしたり、ストリーミング動画を提供したりすることができます。OCAP RI は送られてくる動画をシミュレートするため

に VLC を使用しています。ユーザーが OCAP RI を終了すると、スクリプト (バッチ・ファイル) も VLC を終了しようとします。プロンプトで「n」を入力すると、スクリプトがジョブを実行して VLC を終了します。

再度コマンドラインから実行する

デスクトップ上のアイコンから OCAP RI を起動すると便利ですが、コマンドラインから OCAP RI を起動することもできます。コマンドラインから OCAP RI を起動すると、いろいろな点で便利です。

Windows のコマンドライン (「スタート」 > 「すべてのプログラム」 > 「アクセサリ」 > 「コマンド プロンプト」) で、カレント・ディレクトリーを OCAP RI をインストールした ocap-ri サブディレクトリーに切り換えます。例えば、デフォルトのインストールの場所を受け入れた場合には、ディレクトリーを下記のように変更します。

```
C:\>cd CableLabs\tru2way\ocap-ri
```

今度はコマンド・プロンプトから「runRI.bat」と入力して Enter を押し、runRI.bat スクリプトを実行します。これはもちろん、デスクトップ上のアイコンをダブルクリックした時に実行されるスクリプトと同じです。ボール・アプリケーションで遊ぶのを終えたら、先ほどと同じ方法で OCAP RI を終了します。つまりコマンドライン・ウィンドウで Ctrl-C を押し、表示される確認のプロンプトに対して「n」を入力します。

OCAP RI 上でアプリケーションを扱う

ある時点になると、皆さんは単に CirclesXlet で遊ぶ以上のことをしたくなるはずです。それにはまず、OCAP RI がどのアプリケーションを実行するのかをどのように判断しているのかを理解する必要があります。それが理解できたら、コンソール出力をどのように制御するのかを少し理解する必要があります。

どのアプリケーションを実行するのか？

OCAP RI は主に 2 つのファイルを使ってアプリケーションを見つけます。その 1 つは、重要な構成ファイルである mpeenv.ini です。このファイルは、OCAP RI がインストールされているディレクトリーの ocap\bin\CableLabs\simulator\Win32\debug\env\ にあります。このファイルをテキスト・エディターで開き、内容を調べてみます。このファイルでは数々の設定がなされています。

ここで注目すべき設定は JVM のクラスパスです。JVM のクラスパスはファイルの先頭の近くがあり、VMOPT.0 という名前で指定されています。VMOPT.0 に指定されているのは、実質的にはコマンドライン・パラメーターです。VMOPT.0 の定義は下記のとおりです。

```
VMOPT.0=-Djava.class.path=/syscwd/sys/ocap-classes.jar;  
/syscwd/sys/ocap-rez.jar;  
/syscwd/sys/support.jar;  
/syscwd/qa/ocap-test.jar;  
/syscwd/qa/support-test.jar;  
/syscwd;  
/syscwd/usr;  
/syscwd/apps/eas
```

VMOPT.0 の定義は長い 1 行で記述されていますが、ここではわかりやすくするために何行にも分けて記述してあります。OCAP RI の世界では、/syscwd はカレント作業ディレクトリーを意味し、この場合には mpeenv.ini がある場所と同じです。

OCAP RI はクラスパスの各エントリーを調べ、hostapp.properties ファイルを探します。OCAP RI はこのファイルに記述されているアプリケーションを起動します。このことがわかると、CirclesXlet がどのように実行されるのかを理解することができます。

クラスパスの最初の 5 つのエントリーは JAR ファイルであり、これらのファイルに対応する hostapp.properties はありません。クラスパス上で次にあるのは /syscwd です。ocap\bin\CableLabs\simulator\Win32\debug\env\ (実はこれが OCAP の世界の /syscwd です) を調べてみると、hostapp.properties ファイルが見つかります。hostapp.properties をテキスト・エディターで開くと、このファイルはいくつかのフィールドを定義しており、またアプリケーション・クラスの名前を指定しています。この場合、そのクラスの名前は com.cablelabs.circlesxlet.CirclesXlet です。これこそが OCAP RI が実行するアプリケーションです。

テレビを見る

これらがどのように組み合わされるのかを理解するためには、CirclesXlet ではなく TuneTest を実行するように hostapp.properties を変更します。TuneTest は OCAP によるテレビのチューニング用 API を使用する比較的単純なデモです。

まず、CirclesXlet を指定している application_name と initial_class_name の行の前にハッシュ記号 (#) を付けることで、この 2 行をコメントアウトします。次に、コメントアウトしてあった TuneTest の 4 つのエントリーのコメントを削除します。この変更を加えると、hostapp.properties の最後の部分は下記ようになるはずです。

```
app.0.application_identifier=0x000000015205
app.0.application_control_code=AUTOSTART
app.0.visibility=VISIBLE
app.0.priority=220
app.0.base_directory=/syscwd/qa/xlet
#app.0.application_name=Circles
#app.0.initial_class_name=com.cablelabs.circlesxlet.CirclesXlet
app.0.application_name=TuneTest
app.0.initial_class_name=org.cablelabs.xlet.TuneTest.TuneTestXlet
app.0.args.0=config_file=config_tunetest.properties
app.0.args.1=java_tv_channel_map=true
```

これで OCAP RI を再度起動する準備がほぼ整いましたが、起動する前にあと 2 つしなければならないことがあります。1 つは TuneTest 用の構成ファイルを用意すること、そしてもう 1 つは動画ストリーミング・ソフトウェアを調整することです。

hostapp.properties の app.0.args.0 というエントリーに注目してください。このエントリーは TuneTest が見つけようとする構成ファイルの名前を指定しています。このファイルを適切な場所に配置する必要があります。

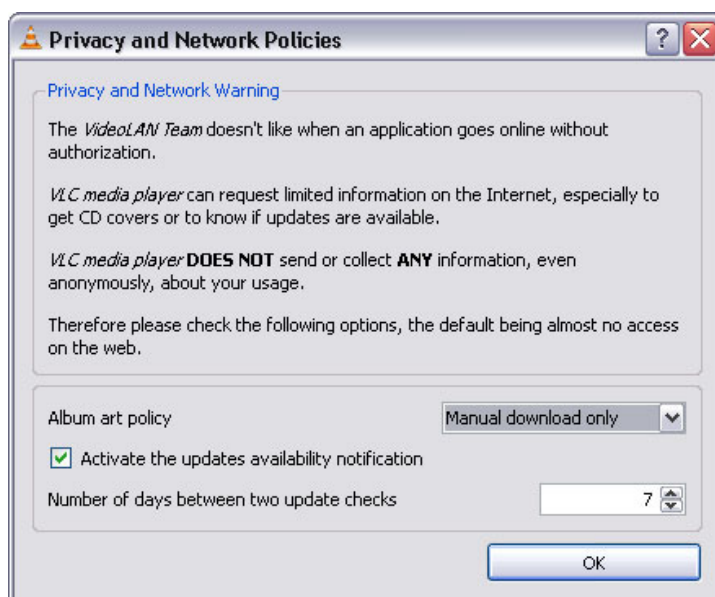
構成ファイルは OCAP RI をインストールしたディレクトリーの extras ディレクトリーにあります。デフォルトのインストール・ディレクトリーを使用した場合には、このディレクトリーは

c:\CableLabs\tru2way\ocap-ri\extras です。config_tunetest.properties をコピーし、ocap\bin\CableLabs\simulator\Win32\debug\env\qa\xlet に貼り付けます。このディレクトリーは /syscwd/qa/xlet と同じですが、hostapp.properties でベース・ディレクトリーとして指定されています。

TuneTest は受信した動画を表示します。そのため、OCAP RI は VLC を使います。VLC は野球の新しいグローブを使うときと同様に、使用前に少し調整が必要です。初めて VLC を実行すると、プライバシー設定を含むダイアログが表示されます。OCAP RI が VLC を実行しようとするとき、VLC は (実際には表示されませんが) このダイアログで止まってしまい、まだ VLC を使って動画を表示することはできません。

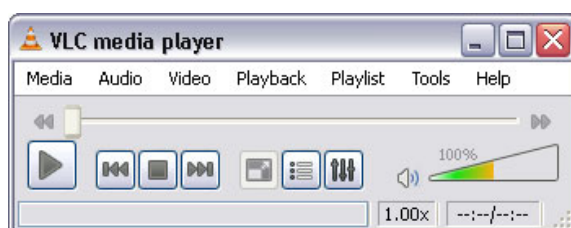
VLC を調整するためには、コマンドライン・ウィンドウを開きます。OCAP RI をインストールしたディレクトリーの ricommon\resources\Win32\VLC までナビゲートします。そして「vlc」と入力して Enter を押します。すると即座にコマンド・プロンプトに戻り、数秒すると VLC のプライバシー設定のダイアログが表示されます (図 3)。

図 3. VLC のプライバシー設定のダイアログ



この画面で必要なことは、「OK」をクリックすることだけです。するとダイアログが消え、VLC のメイン・ウィンドウが表示されます (図 4)。

図 4. VLC のメイン・ウィンドウ



この段階で VLC を終了することも可能です。VLC を終了した場合、再度 VLC を実行すると、即座にメイン・ウィンドウが表示されます。これで、OCAP RI から VLC を自動的に実行することができます。

これで、すべてのものが準備できました。runRI.bat コマンドを使って再度 OCAP RI を起動します。おめでとうございます。皆さんが見ているのは擬似的なテレビ画面です。図 5 はこの様子を示したものです。

図 5. ビールとポップコーンを片手にテレビの画面を見る



ここで再生される動画は、滑らかな表示には程遠く、チャンネルを切り換えるのにも 20 秒近くかかります (ハードウェアが私のものよりも高速な処理を行えるのであれば、もっと良い結果が得られるかもしれません)。とは言え、TuneTest は OCAP を使ってテレビらしいことをするための好例とすることができます。

内部を調べる

コマンドライン・ウィンドウで OCAP RI を実行すると、驚くほど大量の出力が生成されることに気付いた人がいるかもしれません。この出力を調べるとデバッグに役立つ場合があります。

OCAP RI を実行すると、OCAP RI は出力のログを自動的に RILog.txt ファイルに保存します。このファイルをテキスト・エディターで開き、その内容を調べてみてください。何か特定のものを探している場合には、コンソールで上下にスクロールするよりもテキスト・エディターを使って見つけた方が容易なはずです。

mpeenv.ini の設定を変更すると、ログの出力を制御することができます。エントリーの先頭に「LOG.MPE」を付けると、特定のタイプのログが出力されるように制御することができます。LOG.MPE.DEFAULT を「ALL」に設定すると、すべてのメッセージが出力されま

す。LOG.MPE.DEFAULT を「NONE」に設定した後、選択的に他のタイプのロギングを有効にすると、特定のログの出力のみを得ることができます。

また、「ALL」は「ALL DEBUG」と同じではないことに注意してください。「ALL DEBUG」を使った方が、より多くのメッセージをコンソールに出力することができます。

第 1 回のまとめ

今回の記事によって、OCAP RI の何が素晴らしいのかがわかり始めてきたことと思います。次世代のテレビのためのアプリケーションを作成しようとする開発者にとって、OCAP RI は強力なツールです。この記事で見てきたように、OCAP RI は Windows コンピューターのみを使って OCAP アプリケーションを実行できる便利なツールです。OCAP RI を使用すれば、特別なハードウェアや実際のテレビ放送環境を利用しなくても、ユーザー・インターフェースを作成したり、チューニングなどの高度な機能をテストしたりすることができます。

この連載の[第 2 回](#)をご期待ください。第 2 回では、OCAP RI や他の OCAP 機器で実行可能なアプリケーションを作成する方法について学びます。そして最後に第 3 回では、OCAP RI そのものの内部構造について学びます。

関連トピック

- [OCAP](#) 規格は CableLabs からオンラインで入手することができます。
- [tru2way](#) は 一般消費者向けの OCAP のブランド名です。
- CableLabs には [OCAP-RI 専用のウィキ](#)が用意されています。
- OCAP RI は [VLC](#) を使用しています。VLC は、送られてくる映像をシミュレートする、クロスプラットフォームでオープンソースのマルチメディア・フレームワークであり、プレーヤーであり、サーバーでもあります。
- 「[Creating Interactive TV Applications With the Tru2way Platform and OCAP](#)」(Bruce Hopkins 著、Sun Developer Network、2009年4月) は、双方向テレビの概念と、OCAP/tru2way の開発を開始するための方法を解説した記事です (ただし RI は使用していません)。
- 「[Blu-ray Disc Application Development with Java ME, Part 1: Creating Your First Application](#)」(Bruce Hopkins 著、Sun Developer Network、2008年9月) には、ブルーレイ・アプリケーションの開発を始めるためのヒントが含まれています。
- 「[Cable Technology - Why Tru2way Will Explode In The U.S. In 2009](#)」(Jeff Bonin 著、TMCnet、2009年5月) は、なぜ OCAP によって新しい製品やサービスへの需要が喚起されているのかについて、ケーブルテレビ業界内部の目で解説しています。
- [developerWorks の Java technology](#) には、Java プログラミングのあらゆる側面を網羅した記事が豊富に用意されています。
- [OCAP RI](#) プロジェクトは [java.net](#) でホストされており、このプロジェクトからバイナリー・インストーラーや Eclipse プラグインをダウンロードすることができ、またソース・コードの入手方法を知ることができます。

© Copyright IBM Corporation 2010

(www.ibm.com/legal/copytrade.shtml)

商標

(www.ibm.com/developerworks/jp/ibm/trademarks/)