



Android でのMagic RIA

モジュールカスタマイズ編



OUTPERFORM THE FUTURE™



目的と前提条件

- 目的: Magic xpa 2.2 を使った、Android RIA クライアントモジュールを、カスタマイズする方法を説明します。
- 環境的な前提条件としては:
 - Windows PC上に、Magic xpa 2.2 Enterprise Studio (あるいは体験版) をインストールしてあること。
 - Android デバイスを所有していて、テストに使えること。
※ Android SDKのエミュレータでも可能です。
 - Magic xpa Studio のあるPCへ、Androidからネットワーク接続できること。
- 読者のスキルの前提条件としては:
 - Magic xpa 2.2 (あるいは uniPaaS V1Plus) で、Windows RIA システムを作成した経験があること。
 - Androidアプリの開発経験は前提としません。(経験あればもっと良い)



カスタムビルドモジュールはなぜ必要ですか？

- 製品に添付されているMyApp.apk は、開発時の確認用の暫定モジュールです。実運用には使いません。
- 実運用時には、アプリケーション(御社)独自のアイコン、ロゴ、サーバ情報などをカスタマイズしたクライアントモジュールを作成して、エンドユーザ様に納入してください。



準備

カスタマイズを始めるにあたり、開発用 Windows PCに、次のような準備をしておきます。

- Java SE SDK (RTEではない) をインストール
- Android SDK をインストール
- Android SDK Manager で必要ファイルを追加ダウンロード & インストール

参考: Eclipse は必要ありません。



Java SE SDK

Oracle Technology Network

Sign In/Register for Account Help Select Country/Region Communities I am a... I want to... Search

PRODUCTS AND SERVICES SOLUTIONS DOWNLOADS STORE SUPPORT TRAINING PARTNERS ABOUT Oracle Technology Network

Oracle Technology Network > Java > Java SE > Downloads

Overview Downloads Documentation Community Technologies Training

Java SE Downloads

Latest Release Next Release (Early Access) Embedded Use Previous Releases

Java Platform (JDK) 7u13 JavaFX 2.2.5 JDK 7 + NetBeans

Here are the Java SE downloads in detail:

Java Platform, Standard Edition	
Java SE 7u13 This release includes important security fixes. Oracle strongly recommends that all Java SE 7 users upgrade to this release. Learn more	JDK DOWNLOAD
"What Java Do I Need?" You must have a copy of the JRE (Java Runtime Environment)	JRE DOWNLOAD
JDK 7 Docs Installation	JRE 7 Docs Installation

Java SDKs and Tools

- Java SE
- Java EE and Glassfish
- Java ME
- JavaFX
- Java Card
- NetBeans IDE

Java Resources

- New to Java?
- APIs
- Code Samples & Apps
- Developer Training
- Documentation
- Java.com
- Java.net
- Student Developers
- Tutorials

Java magazine Get it now for FREE! Subscribe Today

13:50 2013/02/18

Java SE Development Kit 7u13

You must accept the [Oracle Binary Code License Agreement for Java SE](#) to download this software.

Thank you for accepting the Oracle Binary Code License Agreement for Java SE; you may now download this software.

Product / File Description	File Size	Download
Linux x86	106.64 MB	 jdk-7u13-linux-i586.rpm
Linux x86	92.97 MB	 jdk-7u13-linux-i586.tar.gz
Linux x64	104.77 MB	 jdk-7u13-linux-x64.rpm
Linux x64	91.69 MB	 jdk-7u13-linux-x64.tar.gz
Mac OS X x64	143.71 MB	 jdk-7u13-macosx-x64.dmg
Solaris x86 (SVR4 package)	135.55 MB	 jdk-7u13-solaris-i586.tar.Z
Solaris x86	91.95 MB	 jdk-7u13-solaris-i586.tar.gz
Solaris x64 (SVR4 package)	22.54 MB	 jdk-7u13-solaris-x64.tar.Z
Solaris x64	14.96 MB	 jdk-7u13-solaris-x64.tar.gz
Solaris SPARC (SVR4 package)	135.83 MB	 jdk-7u13-solaris-sparc.tar.Z
Solaris SPARC	95.28 MB	 jdk-7u13-solaris-sparc.tar.gz
Solaris SPARC 64-bit (SVR4 package)	22.89 MB	 jdk-7u13-solaris-sparcv9.tar.Z
Solaris SPARC 64-bit	17.58 MB	 jdk-7u13-solaris-sparcv9.tar.gz
Windows x86	88.74 MB	 jdk-7u13-windows-i586.exe
Windows x64	90.41 MB	 jdk-7u13-windows-x64.exe



Java環境変数設定

Java SDKをインストールしただけだと、Java関係の環境変数がセットされていません。以下の環境変数を設定してください。

- JAVA_HOME
 - C:¥Program Files¥Java¥jdk1.7.0_13¥
- PATH
 - C:¥Program Files¥Java¥jdk1.7.0_13¥bin¥ 追加

※ JDKのバージョンにより違いがあります。



確認:

コマンドプロンプトで、`javac -version` と入れて、インストールした Java のバージョンが表示されればOKです。(javac です。java ではありません)

A screenshot of a Windows command prompt window. The title bar is blue and contains the text '管理者: C:\Windows\system32\cmd.exe'. The command prompt itself has a black background with white text. The first line shows the command 'C:\Users\magic>javac -version'. The second line shows the output 'javac 1.7.0_13'. The third line shows the prompt 'C:\Users\magic>' again, indicating the command has finished executing.

```
管理者: C:\Windows\system32\cmd.exe  
C:\Users\magic>javac -version  
javac 1.7.0_13  
C:\Users\magic>
```




Android SDK インストール

- Android SDKのバージョンにより、インストールのしかたが違いかもしれません。
- 現時点 (2013年2月) での最新は、
 1. Android のサイトから、ADT Bundle (必要な物がひと通り入っているZIPファイル)をダウンロードして、解凍する。
 2. Android SDK Manager から、必要なものを更にダウンロード&インストールする。



Android のダウンロードサイトを開く

・ <http://developer.android.com/sdk/index.html>

The screenshot shows the Android Developer website at <http://developer.android.com/sdk/index.html>. The browser's address bar displays the URL. The website has a navigation bar with tabs: Developers, Design, Develop (highlighted in orange), and Distribute. Below this is a secondary navigation bar with links: Training, API Guides, Reference, Tools (highlighted in orange), and Google Services. The main content area is titled "Get the Android SDK". It includes a "Developer Tools" sidebar with a "Download" link highlighted. The main text explains that the Android SDK provides API libraries and developer tools for building, testing, and debugging apps. It recommends downloading the ADT Bundle for new developers. A large blue button with a red border is labeled "Download the SDK" and "ADT Bundle for Windows". To the right of the text is a 3D graphic of the Android robot made of blue blocks.

Developers ▾ | Design | **Develop** | Distribute

Training | API Guides | Reference | **Tools** | Google Services

Get the Android SDK

The Android SDK provides you the API libraries and developer tools necessary to build, test, and debug apps for Android.

If you're a new Android developer, we recommend you download the ADT Bundle to quickly start developing apps. It includes the essential Android SDK components and a version of the Eclipse IDE with built-in **ADT (Android Developer Tools)** to streamline your Android app development.

With a single download, the ADT Bundle includes everything you need to begin developing apps:

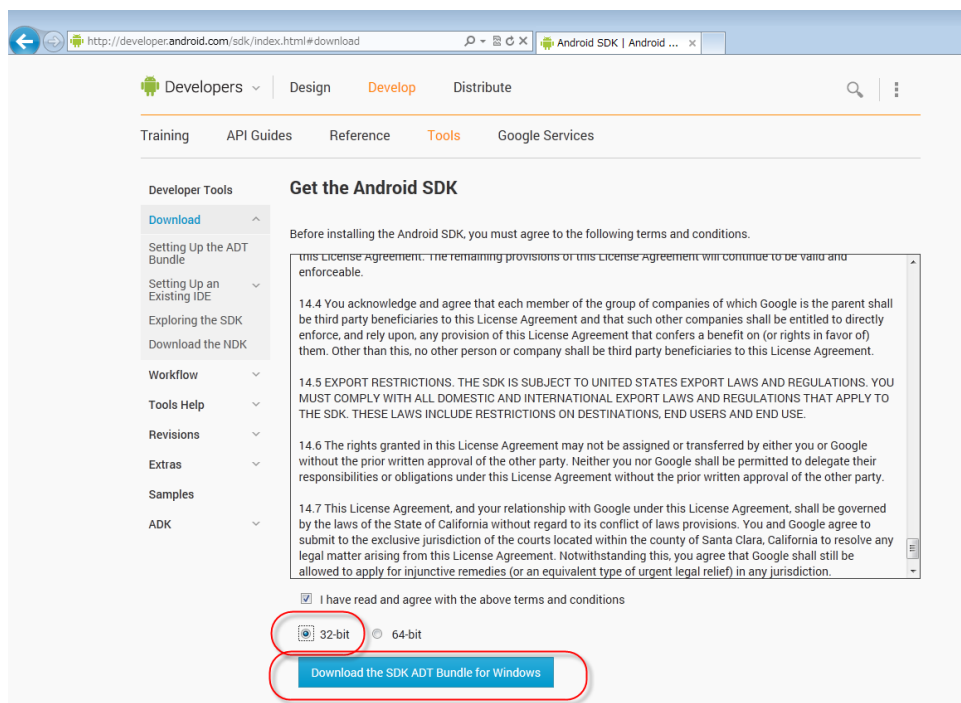
- Eclipse + ADT plugin
- Android SDK Tools
- Android Platform-tools
- The latest Android platform
- The latest Android system image for the emulator

Download the SDK
ADT Bundle for Windows



ダウンロードする。

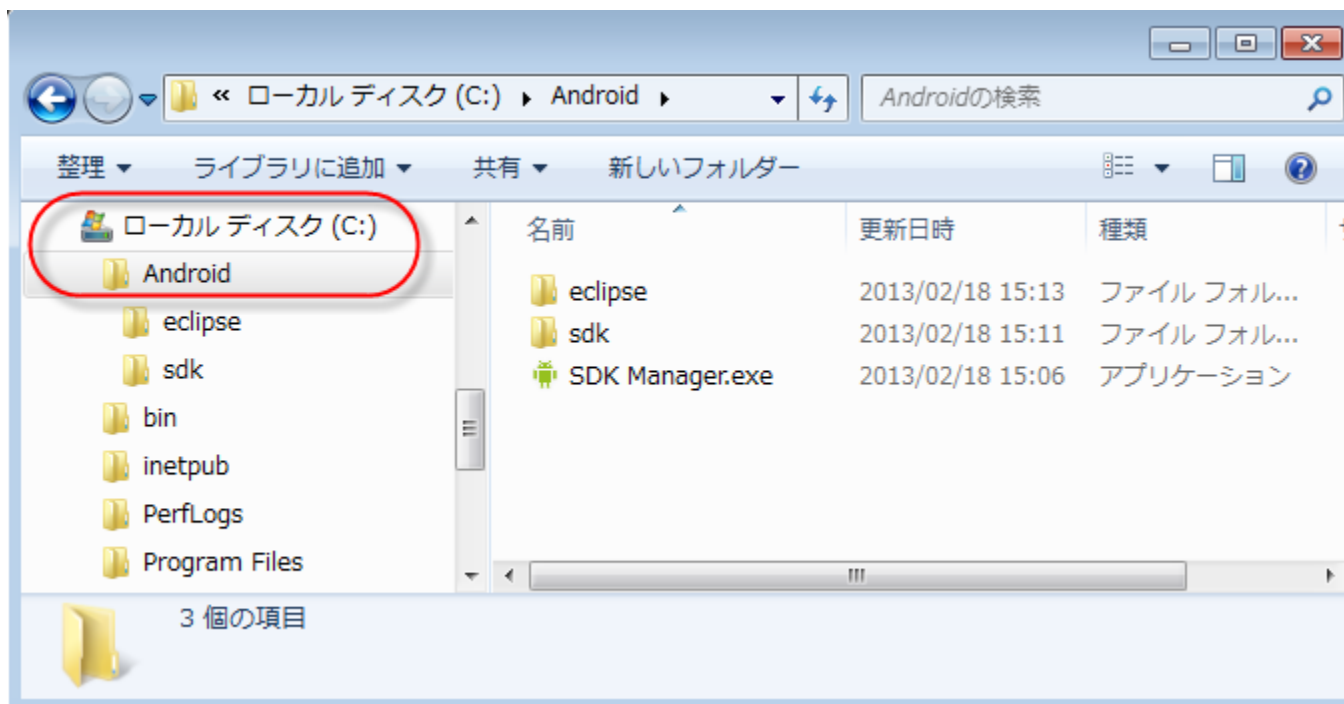
- Agreement に合意して、OS/ビット数に応じたモジュールをダウンロードします。
- adt-bundle-windows-x86.zip がダウンロードされます。





ZIP解凍

- ・ C:¥Android に解凍します。

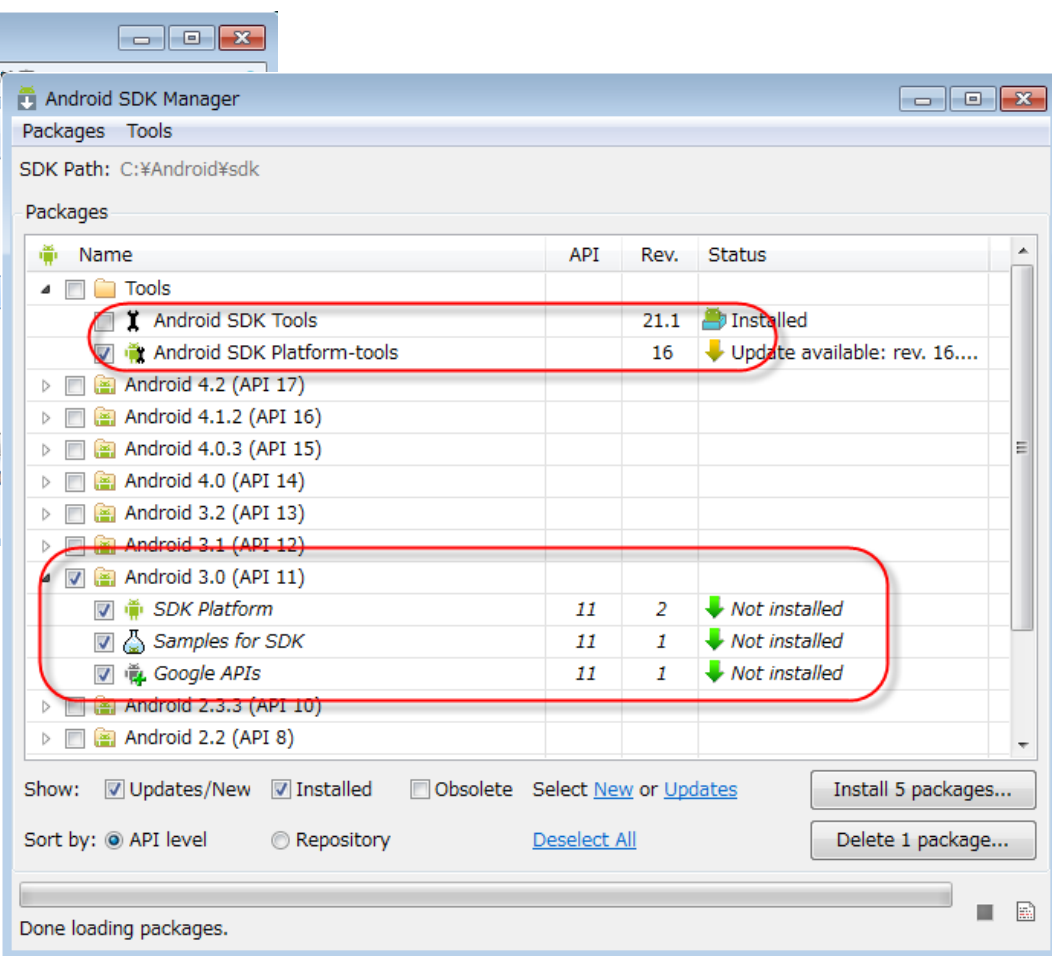
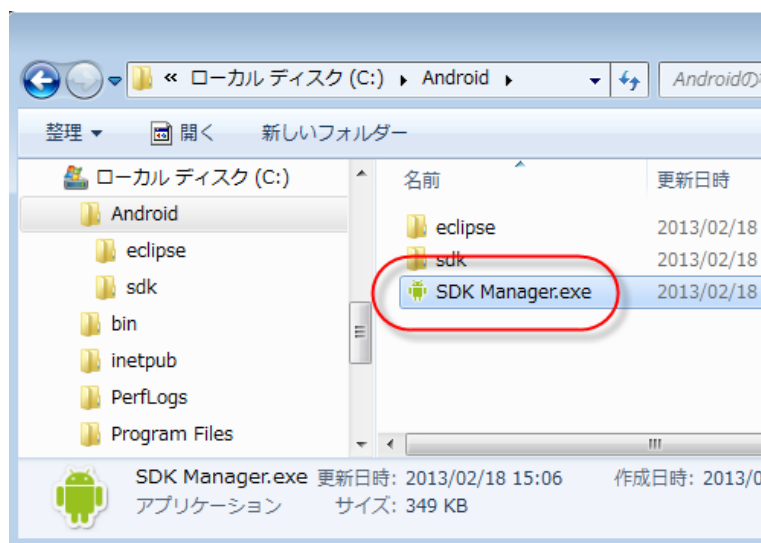


参考：別のフォルダに解凍した場合には、以後の記述中の箇所もそれに合わせて変更してください。



SDK Manager を起動

- Android SDK Platform-tools、Android 3.0(API11) を選択して、「Install 5 packages」ボタンを押します。

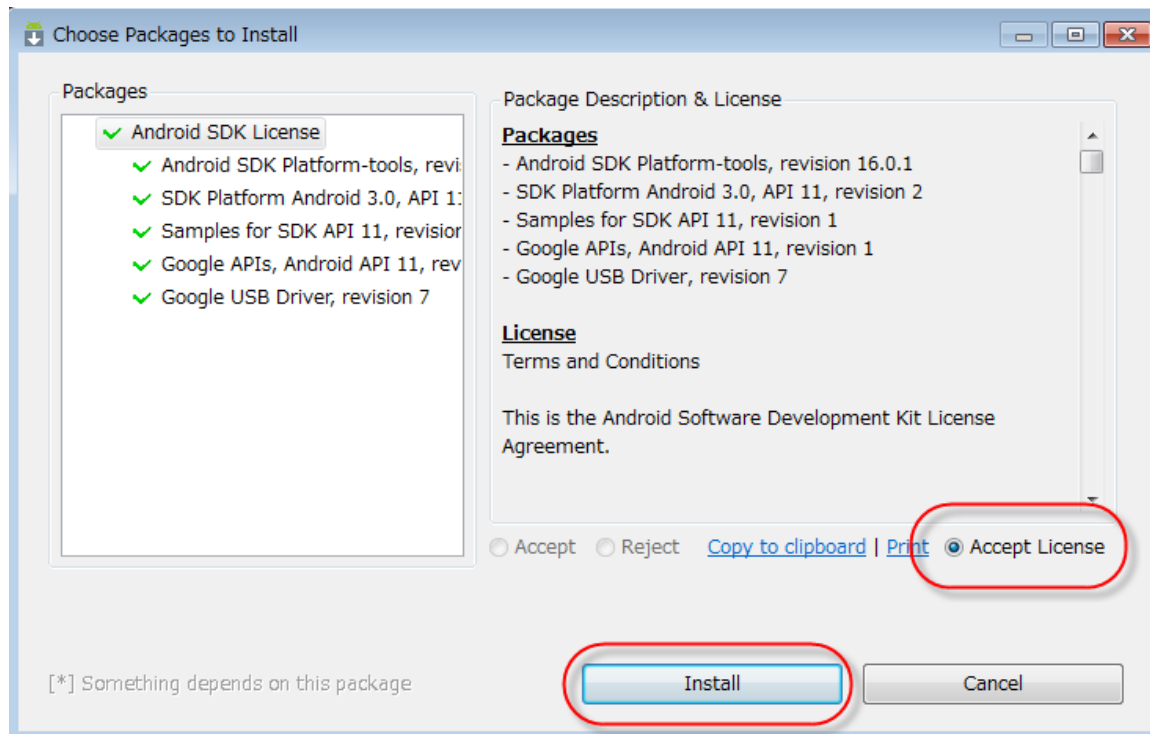


注意:

- SDKのバージョンは、Android 3.0 (API11) を選んでください。
- 環境やバージョンにより、packages 数は違うかもしれません



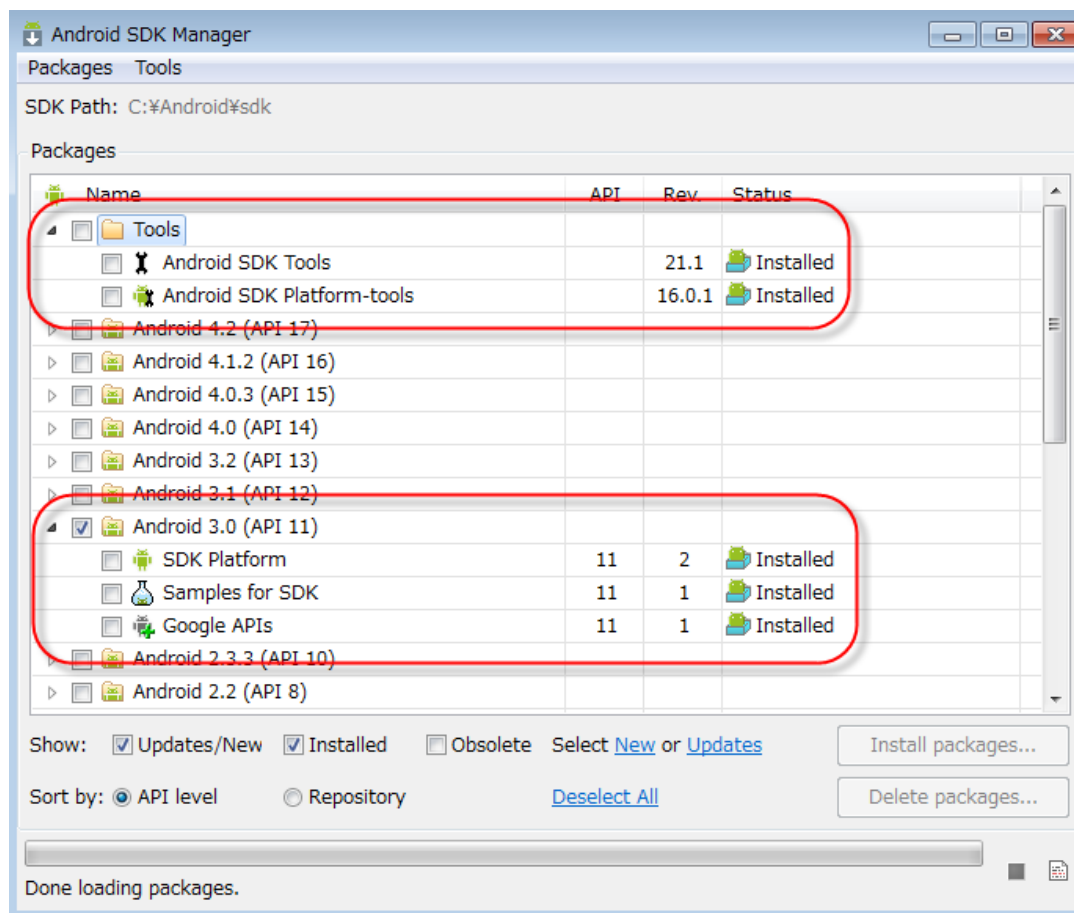
- Accept License をチェックして、Install ボタンを押します。→ インストールが始まります。





インストールパッケージの確認

- インストールが完了したら、選択したものが“installed”になっているか確認します。





Android SDK パスの追加

- 環境変数 PATH に、以下のパスを追加します。
 - C:¥Android¥sdk¥tools
 - C:¥Android¥sdk¥platform-tools



Android SDKインストール完了の確認

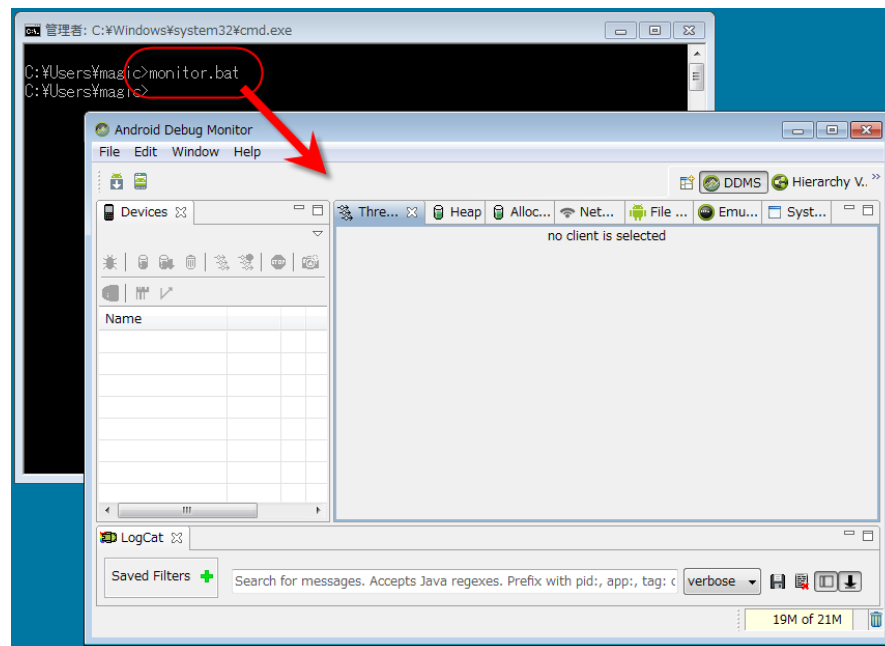
コマンドラインから以下を実行してみ、正しくインストール・設定されたか、確認してください。

- adb version を実行
→ adb (Android Debug Bridge) のバージョンが表示されます。
- monitor.cmd を実行
→ Android Debug Monitor が開きます。

※ 確認したら閉じてください。

```
C:\Users¥magic>adb version
Android Debug Bridge version 1.0.31

C:\Users¥magic>
```





とりあえずデフォルトのままビルド

Java および Android SDKのインストールができれば、とりあえず、ほぼデフォルトの設定のままのモジュールを、以下の手順で作成してみましょう。

1. 作業環境のバックアップ: RIAModules¥Android 以下を、バックアップ用コピーします。
2. 最小限の設定の変更: Setting.properties を変更します。
3. ビルドの実行: Build.cmd を実行します。

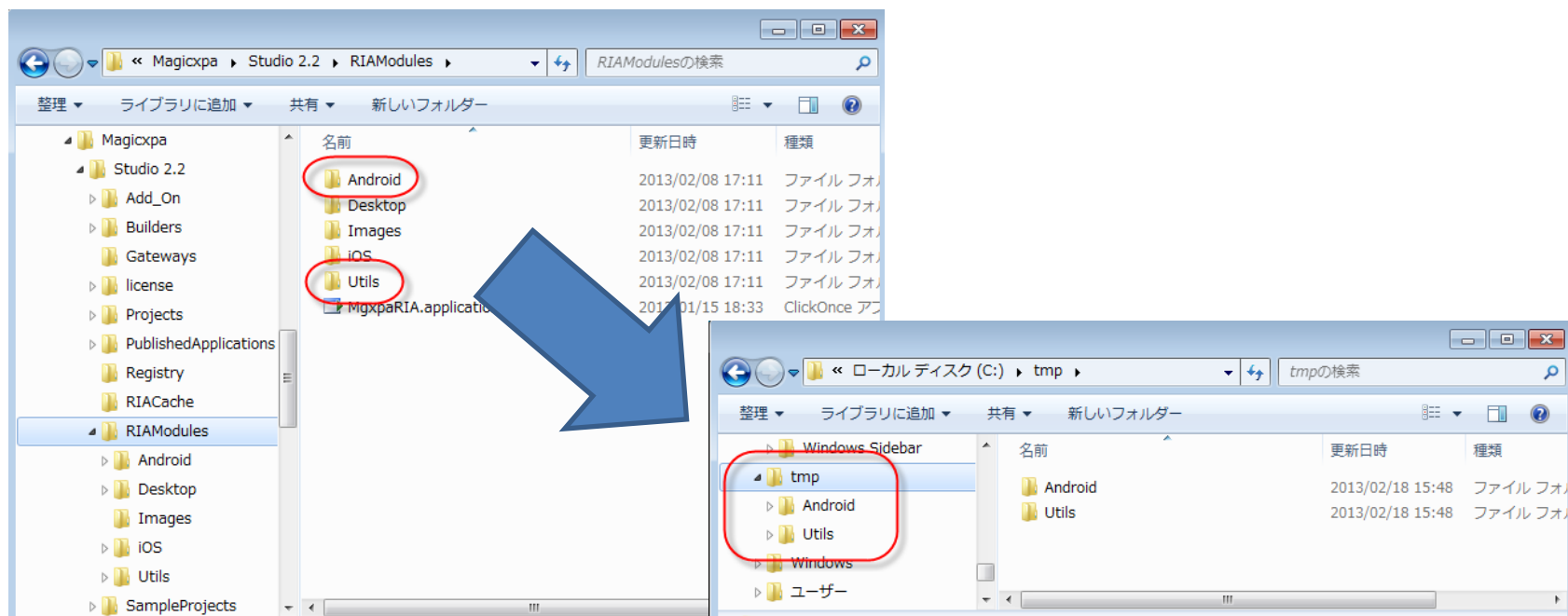
以下に詳しく説明します。



作業環境のバックアップ

- 作業は、一時ファイルにコピーを作って行いましょう。(オリジナルは、そのままにしておきます)
- 試行錯誤でおかしくなったら、オリジナルから再度コピーしましょう。
- C:\Program Files\Magicxpa\Studio 2.2\RIAModules の下にある
 - Android
 - Utils

を一時ファイル (ここでは c:\tmp) にコピーします。





最小限の設定の変更

- ビルドの設定ファイル
C:¥tmp¥Android¥Source¥settings.properties をテキストエディタで編集します。
- sdk.dir → Android SDKのパス
 - 「¥」マークは「¥ ¥」にします。
 - 最後に「¥ ¥」をつけます
- client.title → アプリのタイトル
- package.name → 任意の名前で構いませんが、他のAndroidアプリとぶつからないようにする必要があります。一般に習慣的に、会社名URL（逆順）＋ アプリ名 などとしています。



Setting.properties 例

sdk.dir=**C:¥¥Android¥¥sdk¥¥**

target=android-11

client.title=**HelloMagic**

client.version.code=1

client.version.name=1.0

package.name=**jp.co.magicsoftware.hellomagic**

key.store=..¥¥test.keystore

key.store.password=test123

key.alias=mykey

key.alias.password=test123

output.dir=output

build.dir=build

Android SDK は、API11 (Android 3.0) がインストールされているものとしています。
それ以外のバージョンのSDKをインストールした場合には、target の値も変更する必要があります。



ビルド

- コマンドラインから、build.cmd 実行
→ output フォルダに、.apk ファイルが作成されます。

※ 30～60秒くらいかかります。

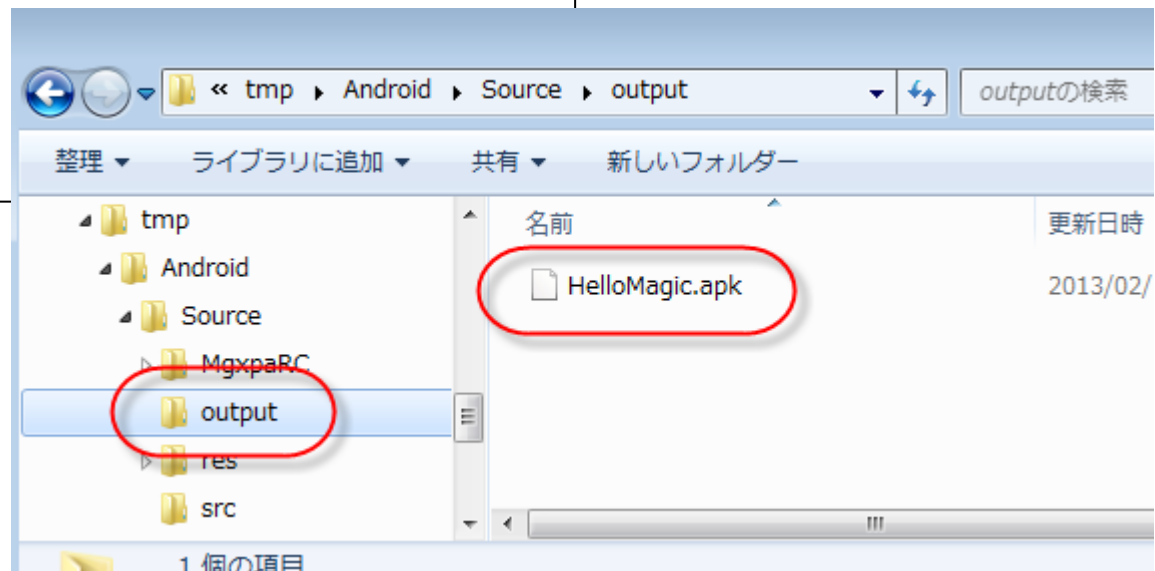
```
C:¥tmp¥Android¥Source>build
Magic RIA Build Tool
[echo] Building application...
[echo] Destination folder: output
[echo] Moving the signed APK to the output folder
[echo] Build Finished. Result file is: C:¥tmp¥Android¥Source¥output¥HelloMagic.apk
```

BUILD SUCCESSFUL

Total time: 38 seconds

続行するには何かキーを押してください...

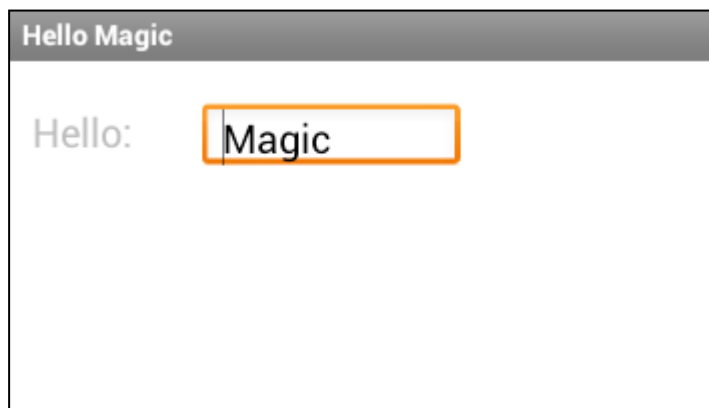
```
C:¥tmp¥Android¥Source>
```





確認

- Myapp.apk の場合と同様にして、動作確認します。
 - サーバ側の待機 (Studio で、「プロジェクトを実行」)
 - Android デバイスに .apk ファイルをコピー・インストール。
アプリ起動。
 - DevProps.txt をURLで指定
`http://MGSTUDIO/DevProps.txt`
 - → 無事動作すればOK！





カスタマイズできるもの

- 以上では、必要最小限の変更だけをしてモジュールをビルドしてきました。
- それ以外に、自社用にカスタマイズできる項目には、次のようなものがあります。
 - サーバ情報を予め指定する。
 - アプリケーション名を日本語にする。
 - アイコンを変更する。
 - ロゴを変更する。



サーバ情報を指定するには？

- デフォルトのビルドでは、最初にDevProp.txtのURLを入力しなければなりませんでした。
- これを省略するため、次のような手順で、.apk にあらかじめサーバ情報を埋め込んでおくことができます。
 - C:\¥tmp¥Android¥Source¥execution.properties を編集します。
 - Key="URL" の property で、val として、DevProps.txt の URL を指定します。
 - 再度、build.cmd を実行します → HelloMagic.apk 作成 → ダウンロード＋インストール → 起動時にURL入力ダイアログが出ないことを確認してください。

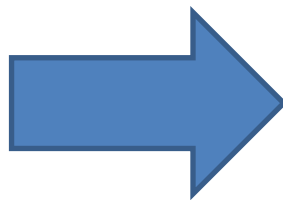
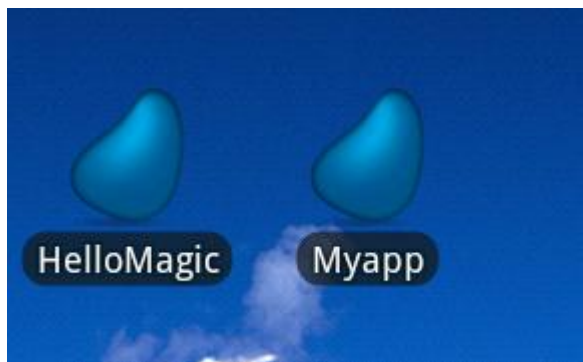
```
<properties>  
<property key="URL" val="http://MGSTUDIO/DevProps.txt"/>  
</properties>
```

参考：URL以外のproperty ("_Server" 等) は削除してかまいません。



アプリケーション名を日本語にするには？

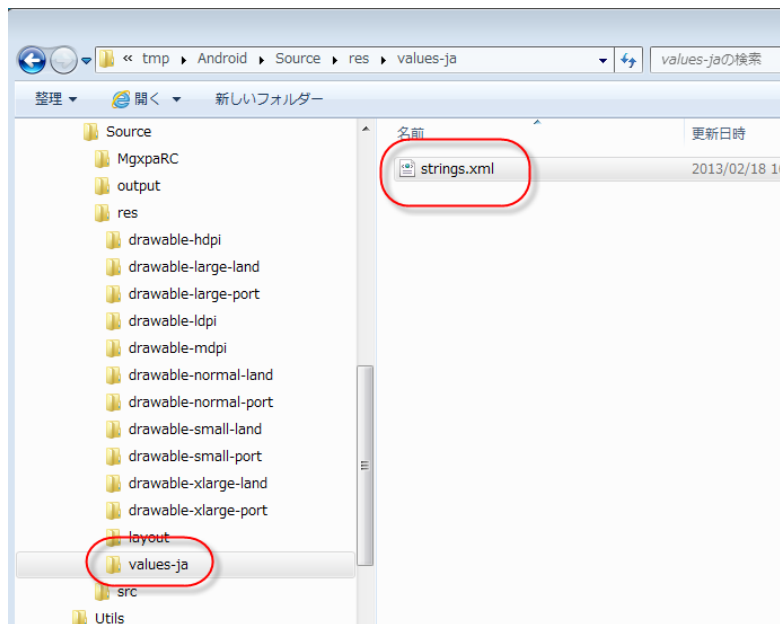
- 今まで、アプリケーション名は、MyApp とか、HelloMagic とか英語ばかりでした。
- 日本のお客様用に、日本語のアプリケーション名にするには、次の手順で名前を設定します。
 - strings.xml に日本語アプリ名指定。
 - Setting.properties にアプリ名指定追加。
 - Buid.xml 編集。





String.xml にアプリ名指定

1. C:\tmp\Android\Source\res\values-ja フォルダを作成します。
2. String.xml ファイルを作成します。
3. String タグを追加します。(内容は、下記参照)
4. UTF-8 で保存します。



```
<?xml version="1.0" encoding="utf-8"?>  
<resources>  
  <string name="app_name">魔法の国</string>  
</resources>
```

この行を追加



Setting.properties にアプリ名指定追加

```
sdk.dir=C:\¥¥Android¥¥sdk¥¥  
target=android-11  
  
client.title=HelloMagic  
client.version.code=1  
client.version.name=1.0  
  
package.name=jp.co.magicsoftware.hellomagic  
  
application.name=@string/app_name  
...
```

この行を追加



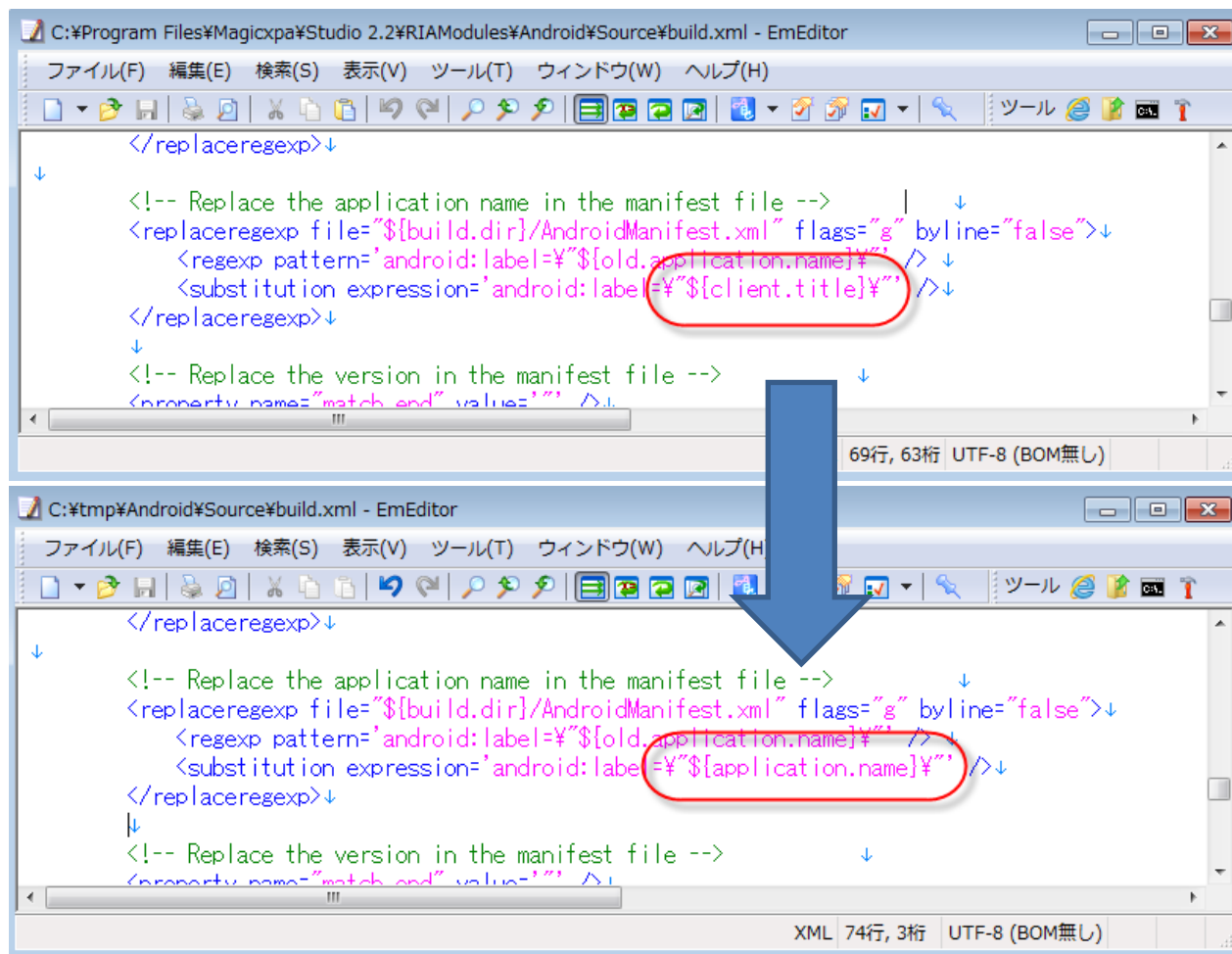
Build.xml 編集

- C:\tmp\Android\Source\build.xml の 70行目あたり。

- Client.title

→

application.name





Build.cmd 実行 → 確認

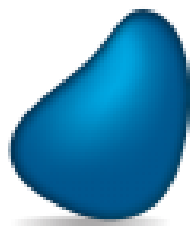
- 再度 build.cmd を実行して、.apk ファイルをコピー・インストールします。
→ タイトルが日本語になっていることを確認してください。





アイコン・ロゴの変更

アイコンというのは、Android 画面
上の起動アイコンのイメージです。



ロゴは、起動時の画面背景イメー
ジです。





アイコンを変更するには・・・

- アイコンを変更するには、res¥drawable-xxxx サブフォルダ内のアイコンファイルを置き換えてください。

フォルダ名	用途	ピクセルサイズ
drawable-hdpi	高解像度用	72x72
drawable-mdpi	中解像度用	48x48
drawable-ldpi	低解像度用	36x36

※ 画面解像度が異なるデバイス用に、異なるアイコンファイルを用意しておきます。



ロゴを変更するには・・・

- ・ res¥drawable-xxxサブフォルダ内の logo.png ファイルを差し替えてください。
- ・ 画面解像度が異なるデバイス用に、異なるロゴファイルを用意しておきます。
- ・ 各解像度ごとに、横方向 (ランドスケープ)用、縦方向 (ポートレート)用の2種類を用意します。

フォルダ名	用途	方向	ピクセルサイズ 例
drawable-large-land	タブレットVGA	横	640 × 480
drawable-large-port	タブレットVGA	縦	480 × 640
drawable-normal-land	スマートフォンHVGA	横	470 × 320
drawable-normal-port	スマートフォンHVGA	縦	320 × 470
drawable-small-land	スマートフォンQVGA	横	426 × 320
drawable-small-port	スマートフォンQVGA	縦	320 × 426
drawable-xlarge-land	タブレットSVGA	横	960 × 720
drawable-xlarge-port	タブレットSVGA	横	720 × 960



THANK
YOU!