



# Publishing Android Applications

# Preparing For Publishing

- ▶ Google has made it relatively easy to publish your Android application so that it can be quickly distributed to end users.
- ▶ The steps to publishing your Android application generally involve the following:
  1. Export your application as an APK (Android Package) file.
  2. Generate your own self-signed certificate and digitally sign your application with it.
  3. Deploy the signed application.
  4. Use the Android Market for hosting and selling your application.

# Versioning

- ▶ Beginning with version 1.0 of the Android SDK, the AndroidManifest.xml file of every Android application includes the android:versionCode and android:versionName attributes:

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="net.learn2develop.LBS"
    android:versionCode="1"
    android:versionName="1.0">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
    <uses-library android:name="com.google.android.maps" />
        <activity android:name=".MainActivity"
            android:label="@string/app_name">
            <intent-filter>
                <action android:name="android.intent.action.MAIN" />
                <category android:name="android.intent.category.LAUNCHER" />
            </intent-filter>
        </activity>
    </application>
    <uses-sdk android:minSdkVersion="7" />
    <uses-permission android:name="android.permission.INTERNET" />
    <uses-permission android:name="android.permission.ACCESS_FINE_LOCATION" />
    <uses-permission android:name="android.permission.ACCESS_COARSE_LOCATION" />
</manifest>
```

# Versioning

- ▶ The `android:versionCode` attribute represents the version number of your application.
- ▶ For every revision you make to the application, you should increment this value by **1** so that you can programmatically differentiate the newest version from the previous one.
- ▶ This value is never used by the Android system, but is useful for developers as a means to obtain the version number of an application.
- ▶ However, the `android:versionCode` attribute is used by Android Market to determine if there is a newer version of your application available.

# Versioning

- ▶ The `android:versionName` attribute contains versioning information that is visible to the users.
- ▶ It should contain values in the format: `<major>.<minor>.<point>`.
- ▶ If your application undergoes a major upgrade, you should increase the `<major>` by 1.
- ▶ For small incremental updates, you can either increase the `<minor>` or `<point>` by 1.
- ▶ For example, a new application may have a version name of "1.0.0".
- ▶ For a small incremental update, you might change to "1.1.0" or "1.0.1".
- ▶ For the next major update, you might change it "2.0.0".

# Versioning

- ▶ If you are planning to publish your application on the Android Market ([www.android.com/market/](http://www.android.com/market/)), the `AndroidManifest.xml` file must have the following attributes:
  - ▶ `android:versionCode` (within the `<manifest>` element)
  - ▶ `android:versionName` (within the `<manifest>` element)
  - ▶ `android:icon` (within the `<application>` element)
  - ▶ `android:label` (within the `<application>` element)
- ▶ The `android:label` attribute specifies the name of your application.
- ▶ This name will be displayed in the Settings ⇔ Applications ⇔ Manage Applications section of your Android device.

# Versioning

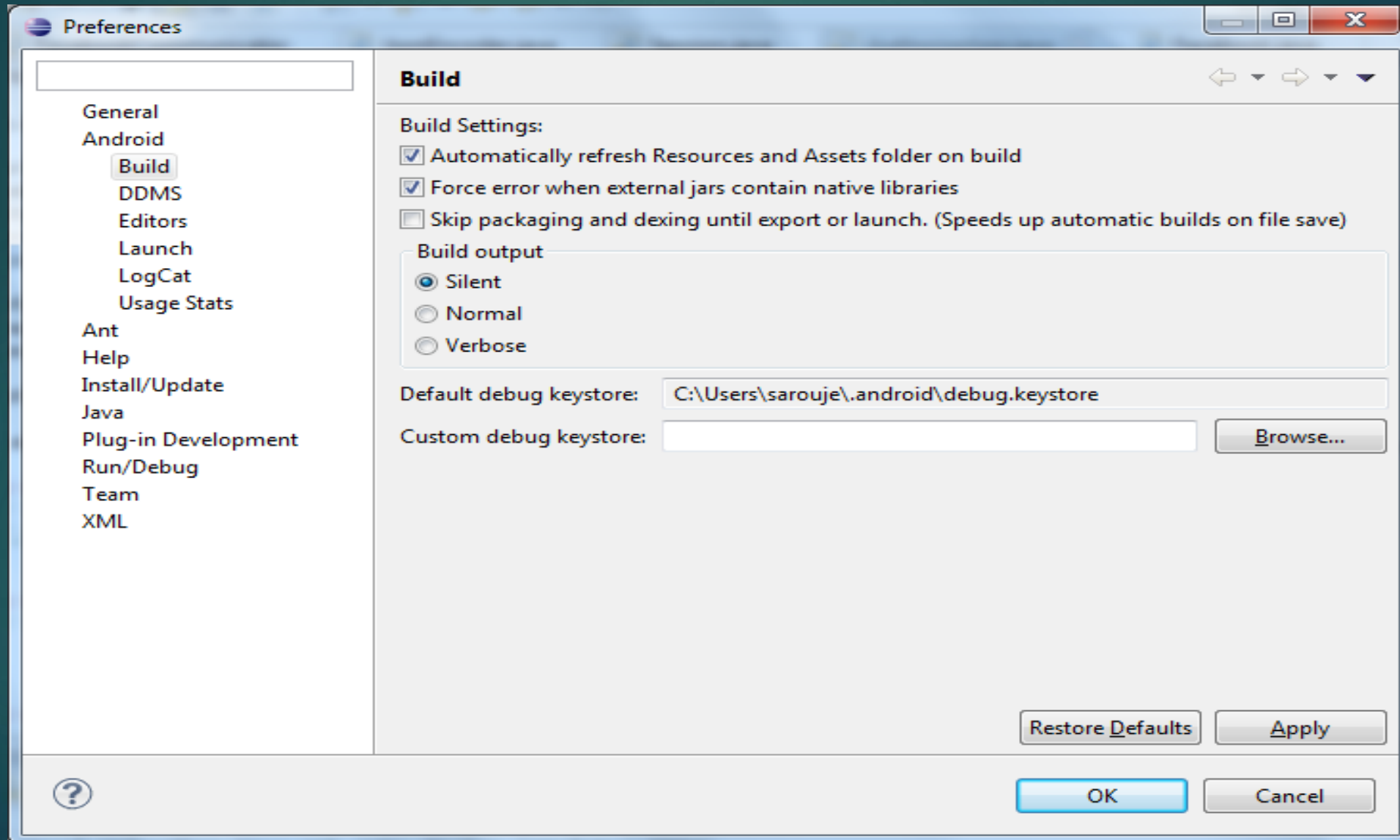
- ▶ In addition, if your application needs a minimum version of the SDK, you can specify it in the `AndroidManifest.xml` file using the `<uses-sdk>` element:
- ▶ `<uses-sdk android:minSdkVersion="7" />`
- ▶ This line indicates that the application requires a minimum of SDK version 7, which is Android 2.1.
- ▶ In general, it is always good to set this version number to the lowest one that your application can support.
- ▶ This ensures that a wider range of users will be able to run your application.

# Digitally Signing Your Android Applications

- ▶ All Android applications must be digitally signed before they are allowed to be deployed onto a device (or emulator).
- ▶ You can generate your own self-signed certificate and use it to sign your Android applications.
- ▶ When you use Eclipse to develop your Android application and then press F11 to deploy it to an emulator, Eclipse automatically signs it for you.
- ▶ You can verify this by going to Windows ⇔ Preferences in Eclipse, expanding the Android item, and selecting Build.
- ▶ Eclipse uses a default debug keystore (appropriately named "debug.keystore") to sign your application.
- ▶ A keystore is commonly known as a digital certificate.



# Digitally Signing Your Android Applications



# Digitally Signing Your Android Applications

- ▶ If you are publishing an Android application, you must sign it with your own certificate.
- ▶ Applications signed with the debug certificate cannot be published.
- ▶ While you can manually generate your own certificates using the `keytool.exe` utility provided by the Java SDK, Eclipse has made it easy for you by including a wizard that walks you through the steps to generate a certificate.
- ▶ It will also sign your application with the generated certificate (which you can also sign manually using the `jarsigner.exe` tool from the Java SDK).

# Deploying APK Files

- ▶ Once you have signed your APK files, you need a way to get them onto your users' devices.
- ▶ The following sections describe the various ways to deploy your APK files.
- ▶ Three methods are covered:
  1. **Deploying manually using the adb.exe tool**
  2. **Hosting the application on a web server**
  3. **Publishing through the Android Market**
- ▶ Besides the above methods, you can install your applications on users' devices through e-mails, SD card, etc. As long as you can transfer the APK file onto the user's device, you can install the application.

# Using the adb.exe Tool

- ▶ Once your Android application is signed, you can deploy it to emulators and devices using the adb.exe (Android Debug Bridge) tool (located in the platform-tools folder of the Android SDK).
- ▶ Using the command prompt in Windows, navigate to the "<Android\_SDK>\platform-tools" folder.
- ▶ To install the application to an emulator/device (assuming the emulator is currently up and running or a device is currently connected), issue the following command:
  - ▶ **adb install "C:\Users\...\Desktop\Project-name.apk"**
- ▶ The adb.exe tool is a very versatile tool that enables you to control Android devices (and emulators) connected to your computer.
- ▶ By default, when you use the adb command, it assumes that currently there is only one connected device/emulator.

# Publishing on the Android Market

- ▶ So far, you have learned how to package your Android application and distribute it in various ways — via web server, the adb.exe file, e-mail, SD card, and so on.
- ▶ However, these methods do not provide a way for users to discover your applications easily.
- ▶ A better way is to host your application on the Android Market, a Google-hosted service that makes it very easy for users to discover and download applications for their Android devices.
- ▶ Users simply need to launch the Market application on their Android device in order to discover a wide range of applications that they can install on their devices.

# Publishing on the Android Market

## ▶ Creating a Developer Profile

- ▶ The first step toward publishing on the Android Market is to create a developer profile at <http://market.android.com/publish/Home>.
- ▶ For this, you need a Google account (such as your Gmail account).
- ▶ Once you have logged in to the Android Market, you first create your developer profile.
- ▶ Click Continue after entering the required information.
- ▶ For publishing on the Android Market, you need to pay a one-time registration fee.
- ▶ Click the Google Checkout button to be redirected to a page where you can pay the registration fee. After paying, click the Continue link.
- ▶ Next, you need to agree to the Android Market Developer Distribution Agreement. Check the “I agree” checkbox and click the “I agree. Continue” link.

# Publishing on the Android Market

## ▶ Submitting Your Apps

- ▶ Once you have set up your profile, you are ready to submit your application to the Android Market.
- ▶ If you intend to charge for your application, click the Setup Merchant Account link located at the bottom of the screen.
- ▶ Here you enter additional information such as bank account and tax ID.
- ▶ For free applications, click the Upload Application link.

# Publishing on the Android Market

- ▶ You will be asked to supply some details for your application. Among the information needed, the following are compulsory:
- ▶ The application in APK format
- ▶ At least two screenshots. You can use the DDMS perspective in Eclipse to capture screenshots of your application running on the Emulator or real device.
- ▶ A high-resolution application icon. This size of this image must be 512×512 pixels.
- ▶ The other information details are optional, and you can always supply them later.



# Publishing on the Android Market

- ▶ The next set of information you need to supply, includes the title of your application, its description, as well as recent changes' details (useful for application updates).
- ▶ You can also select the application type and the category in which it will appear in the Android Market.
- ▶ On the last dialog, you indicate whether your application employs copy protection, and specify a content rating.
- ▶ You also supply your website URL and your contact information.
- ▶ When you have given your consent to the two guidelines and agreements, click Publish to publish your application on the Android Market.

THANK YOU