Publishing

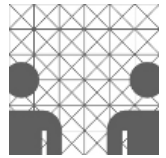
Seminar „Android: Plattform für mobile Geräte“

Von Artiom Wulis

SS2010

Universität Hamburg

MIN Faculty
Department of Informatics
Scientific Computing



Index

Publishing Voraussetzungen

- Publishing Checkliste
  - Versionierung
  - Signierung

Publishing auf:

- Android Market
- Endgerät

Index

Publishing Voraussetzungen

- Publishing Checkliste
  - Versionierung
  - Signierung

Publishing auf:

- Android Market
- Endgerät

Publishing Voraussetzungen

Publishing Checkliste:

1. Anwendung ausgiebig Testen
2. Endbenutzer-Lizenzvereinbarung hinzufügen
3. Icon und Label spezifizieren
4. Logger und Debugger ausschalten,
   temp. Dateien löschen
5. Anwendung versionieren
6. Anwendung signieren

Publishing Checkliste

1. Anwendung ausgiebig Testen

- Auf physischen Geräten testen

Publishing Checkliste

1. Anwendung ausgiebig Testen

- Auf physischen Geräten testen
- Seltene Geräte können weitestgehend emuliert werden
  (Emulator Optionen: -dpi, -scale, -netspeed, -cpu-delay…)

Publishing Checkliste

2. Endbenutzer-Lizenzvereinbarung hinzufügen

- In google nach: ***android eula.java*** suchen
- Oder *http://code.google.com/p/apps-for-android*

Source path:
svn/trunk/Photostream/src/com/google/android/photostream/Eula.java

Publishing Checkliste

3. Icon und Label spezifizieren

- *android:icon="@drawable/icon"*
- *android:label="@string/app_name"*

Publishing Checkliste

3. Icon und Label spezifizieren

- *android:icon="@drawable/icon"*
- *android:label="@string/app_name"*
- In Android Manifest Application (*AndroidManifest.xml*)

```
<application
        android:icon="@drawable/icon"
        android:label="@string/app_name" >
 </application>
```

Publishing Checkliste

4. Logger und Debugger ausschalten, temp. Dateien löschen

- Evtl. Temp. Backup und Logdateien entfernen
- Alle Aufrufe zu den Log-Methoden löschen
- Debugger ausschalten: *android:debuggable="false"*
- In Android Manifest Application (AndroidManifest.xml)

```
<application
        android:debuggable="false" >
 </application>
```

Index

Publishing Voraussetzungen

- Publishing Checkliste
  - Versionierung
  - Signierung

Publishing auf:

- Android Market
- Endgerät

Publishing Checkliste

5. Anwendung versionieren

Warum versionieren?

Publishing Checkliste

5. Anwendung versionieren

Warum versionieren?

- Benutzer müssen Programmversion sehen können

Publishing Checkliste

5. Anwendung versionieren

Warum versionieren?

- Benutzer müssen Programmversion sehen können
- Andere Anwendungen brauchen Versionsangaben

Publishing Checkliste

5. Anwendung versionieren

Warum versionieren?

- Benutzer müssen Programmversion sehen können
- Andere Anwendungen brauchen Versionsangaben
- Android Market, ähnliche Seiten und „ Update Notifier“ benötigen auch Versionsangaben

Publishing Checkliste

5. Anwendung versionieren

Warum versionieren?

- Benutzer müssen Programmversion sehen können
- Andere Anwendungen brauchen Versionsangaben
- Android Market, ähnliche Seiten und
  „ Update Notifier“ benötigen auch Versionsangaben

Achtung!
Android System prüft nicht die Versionsangaben.

Publishing Checkliste

5. Anwendung versionieren

In Android Manifest (*AndroidManifest.xml*)

- *android:versionCode*
- *android:versionName*

Publishing Checkliste

5. Anwendung versionieren

android:versionCode

- Integer
- Ist für andere Anwendungen / Systeme evaluierbar
- Keine Unterscheidung zwischen kleineren / größeren Updates
- Sollte für Anwender nicht sichtbar sein
- Sollte nach jedem Release erhöht werden

Publishing Checkliste

5. Anwendung versionieren

android:versionName

- String
- Ist für andere Anwendungen / Systeme nicht evaluierbar
- Unterscheidung zwischen kleineren / größeren Updates möglich (z.B. 1.21 / 3a usw.)
- Sollte für Anwender sichtbar sein
- Sollte nach jedem Release erhöht werden

Publishing Checkliste

5. Anwendung versionieren

In Android Manifest (*AndroidManifest.xml*)

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
      package="com.example.package.name"
      android:versionCode="2"
      android:versionName="1.1">
    <application android:icon="@drawable/icon"
android:label="@string/app_name">
        ...
    </application>
</manifest>
```

Publishing Checkliste

5. Anwendung versionieren

Android Sdk Version festlegen

- *android:minSdkVersion*
- *android:targetSdkVersion*
- *android:maxSdkVersion*

Publishing Checkliste

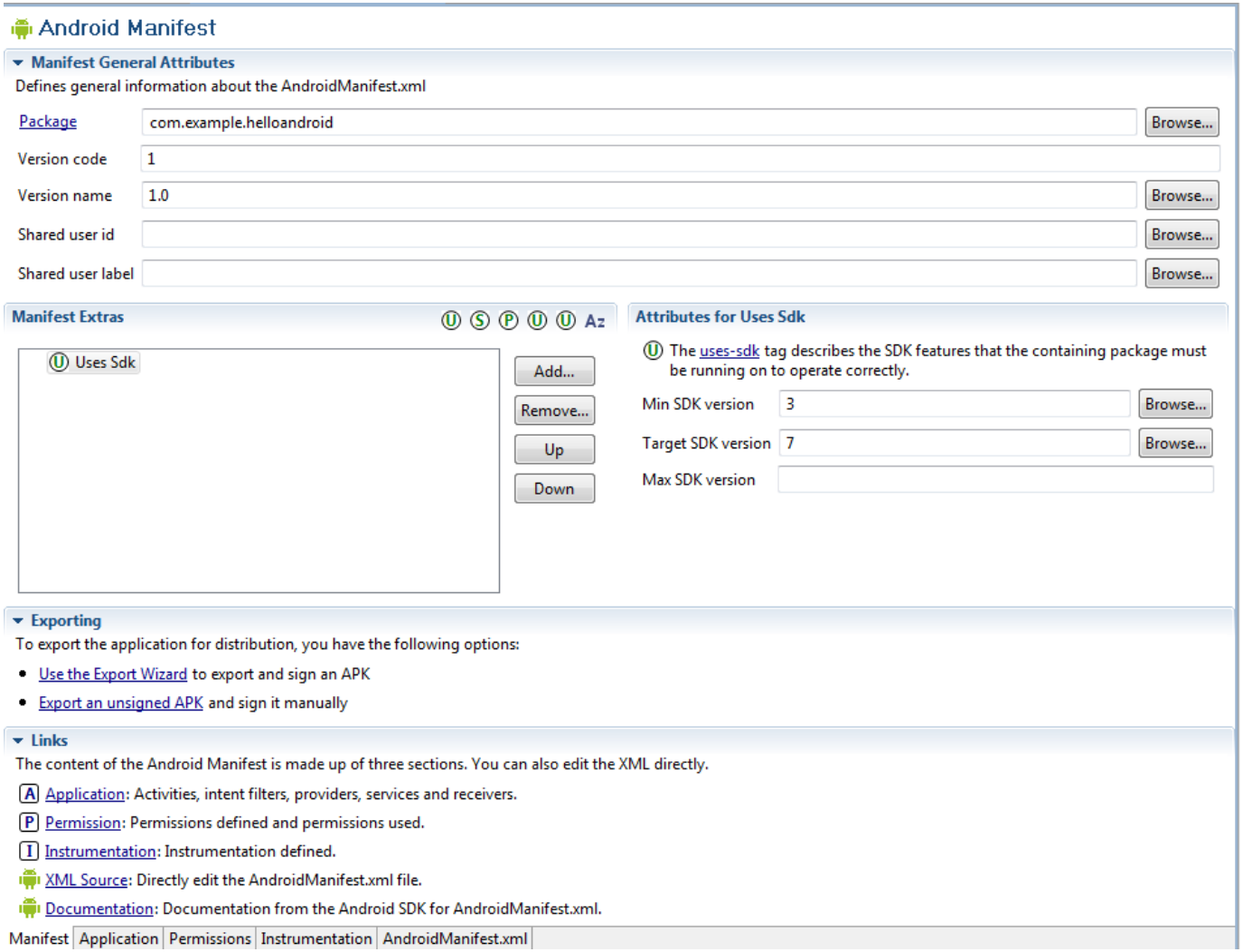
5. Anwendung versionieren

Android Sdk Version festlegen

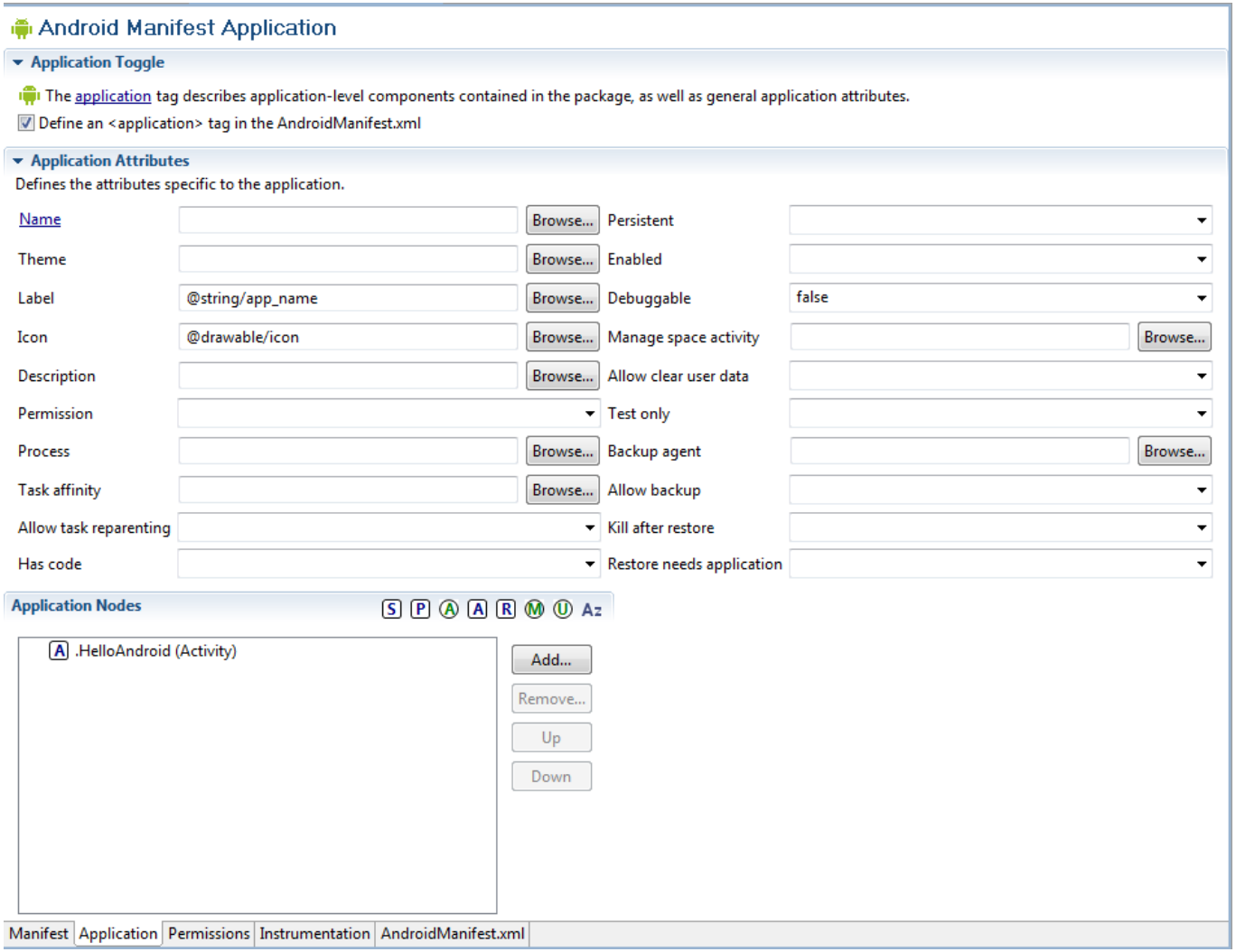
- *android:minSdkVersion*
- *android:targetSdkVersion*
- *android:maxSdkVersion*

```
<?xml version="1.0" encoding="utf-8"?>
<manifest
xmlns:android=http://schemas.android.com/apk/res/android
    …
   <application …</application>
   <uses-sdk android:minSdkVersion="3"
android:targetSdkVersion="7"/>
</manifest>
```

Android Manifest Eclipse Gui



Android Manifest Eclipse Gui



Index

Publishing Voraussetzungen

- Publishing Checkliste
  - Versionierung
  - Signierung

Publishing auf:

- Android Market
- Endgerät

Publishing Checkliste

6. Anwendung signieren

Android Anwendungen müssen signiert werden!

Publishing Checkliste

6. Anwendung signieren

Android Anwendungen müssen signiert werden!

- Sicherheit

Publishing Checkliste

6. Anwendung signieren

Android Anwendungen müssen signiert werden!

- Sicherheit
- Eigenes Zertifikat kann verwendet werden

Publishing Checkliste

6. Anwendung signieren

Android Anwendungen müssen signiert werden!

- Sicherheit
- Eigenes Zertifikat kann verwendet werden
- Android System prüft Ablaufdatum des Zertifikats

Publishing Checkliste

6. Anwendung signieren

Android Anwendungen müssen signiert werden!

- Sicherheit
- Eigenes Zertifikat kann verwendet werden
- Android System prüft Ablaufdatum des Zertifikats
- Standardtools wie Keytool und Jarsigner können benutzt werden (Java SDK)

Publishing Checkliste

6. Anwendung signieren

Strategien

Empfehlung: Alle Anwendungen mit dem selben Zertifikat signieren.

- Update / Upgrade

Publishing Checkliste

6. Anwendung signieren

Strategien

Empfehlung: Alle Anwendungen mit dem selben Zertifikat signieren.

- Update / Upgrade
- Modularität (ein Prozess/Anwendung, Module)

Publishing Checkliste

6. Anwendung signieren

Strategien

Empfehlung: Alle Anwendungen mit dem selben Zertifikat signieren.

- Update / Upgrade
- Modularität (ein Prozess/Anwendung, Module)
- Code/data sharing

Publishing Checkliste

6. Anwendung signieren

Strategien

Empfehlung: Gültigkeitsdauer des Schlüssels vorausdenkend vergeben (>25Jahre).

- Zukünftige Updates?

Publishing Checkliste

6. Anwendung signieren

Strategien

Empfehlung: Gültigkeitsdauer des Schlüssels vorausdenkend vergeben (>25Jahre).

- Zukünftige Updates?
- Mehrere Anwendungen mit gleichem Schlüssel signiert?

Publishing Checkliste

6. Anwendung signieren

Strategien

Empfehlung: Gültigkeitsdauer des Schlüssels vorausdenkend vergeben (>25Jahre).

- Zukünftige Updates?
- Mehrere Anwendungen mit gleichem Schlüssel signiert?
- Android Market? => Ende nach 22.10.2033

Publishing Checkliste

6. Anwendung signieren

Es gibt zwei Arten zu Signieren.

- „Release mode“
- „Debug mode“

Publishing Checkliste

6. Anwendung signieren

„Release mode“

- Passenden privaten Schlüssel generieren
- Anwendung in release mode compilieren
- Fertige .apk signieren

Publishing Checkliste

6. Anwendung signieren

„*Release mode*“

- Passenden privaten Schlüssel generieren
- Anwendung in release mode compilieren
- Fertige .apk signieren

Publishing Checkliste

6. Anwendung signieren

Passenden privaten Schlüssel generieren

Keytool.exe liegt in JavaSDK/bin Ordner

Shell:

keytool.exe -genkey -v -keystore my-release-key.keystore
-alias alias_name -keyalg RSA -validity 10000

Publishing Checkliste

6. Anwendung signieren

Passenden privaten Schlüssel generieren

Keytool.exe liegt in JavaSDK/bin Ordner

Shell:

*keytool.exe -**genkey** -v -keystore my-release-key.keystore*
-alias alias_name -keyalg RSA -validity 10000

generiert ein Schlüsselpaar

Publishing Checkliste

6. Anwendung signieren

Passenden privaten Schlüssel generieren

Keytool.exe liegt in JavaSDK/bin Ordner

Shell:

*keytool.exe -genkey **-v** -keystore my-release-key.keystore*
-alias alias_name -keyalg RSA -validity 10000

detalierte Ausgabe

Publishing Checkliste

6. Anwendung signieren

Passenden privaten Schlüssel generieren

Keytool.exe liegt in JavaSDK/bin Ordner

Shell:

keytool.exe -genkey -v ***-keystore*** *my-release-key.keystore*
-alias alias_name -keyalg RSA -validity 10000

generiert keystore (Schlüssel DB)

Publishing Checkliste

6. Anwendung signieren

Passenden privaten Schlüssel generieren

Keytool.exe liegt in JavaSDK/bin Ordner

Shell:

keytool.exe -genkey -v -keystore ***my-release-key.keystore***
-alias alias_name -keyalg RSA -validity 10000

Name der keystore (Schlüssel DB) Datei

Publishing Checkliste

6. Anwendung signieren

Passenden privaten Schlüssel generieren

Keytool.exe liegt in JavaSDK/bin Ordner

Shell:

keytool.exe -genkey -v -keystore my-release-key.keystore
-alias *alias_name -keyalg RSA -validity 10000*

vergibt ein Alias für den Schlüssel

Publishing Checkliste

6. Anwendung signieren

Passenden privaten Schlüssel generieren

Keytool.exe liegt in JavaSDK/bin Ordner

Shell:

keytool.exe -genkey -v -keystore my-release-key.keystore
*-alias **alias_name** -keyalg RSA -validity 10000*

Schlüsselalias

Publishing Checkliste

6. Anwendung signieren

Passenden privaten Schlüssel generieren

Keytool.exe liegt in JavaSDK/bin Ordner

Shell:

keytool.exe -genkey -v -keystore my-release-key.keystore
-alias alias_name ***-keyalg RSA*** *-validity 10000*

Verschlüsselungsalgorithmus

Publishing Checkliste

6. Anwendung signieren

Passenden privaten Schlüssel generieren

Keytool.exe liegt in JavaSDK/bin Ordner

Shell:

keytool.exe -genkey -v -keystore my-release-key.keystore
-alias alias_name -keyalg RSA ***-validity 10000***

Gültigkeitsdauer in Tagen

Shell Ausgabe

PS C:\Program Files\Java\jdk1.6.0_18\bin> keytool.exe -genkey -v -keystore my-release-key.keystore -alias alias_name -keyalg RSA -validity 10000

Geben Sie das Keystore-Passwort ein:

Geben Sie das Passwort erneut ein:

Wie lautet Ihr Vor- und Nachname? *[Unknown]: Heinz Mustermann*

Wie lautet der Name Ihrer organisatorischen Einheit? *[Unknown]: Android developer*

Wie lautet der Name Ihrer Organisation? [Unknown]: *Uni HH*

Wie lautet der Name Ihrer Stadt oder Gemeinde? [Unknown]: *Hamburg*

Wie lautet der Name Ihres Bundeslandes oder Ihrer Provinz? [Unknown]: *Hamburg*

Wie lautet der Landescode (zwei Buchstaben) für diese Einheit? [Unknown]: *DE*

Ist CN=Heinz Mustermann, OU=Android developer, O=Uni HH, L=Hamburg, ST=Hamburg, C=DE richtig? [Nein]: *ja*

Erstellen von Schlüsselpaar (Typ RSA, 1.024 Bit) und selbstunterzeichnetem Zertifikat (SHA1withRSA) mit einer Gültigkeit von 10.000 Tagen

für: CN=Heinz Mustermann, OU=Android developer, O=Uni HH, L=Hamburg, ST=Hamburg, C=DE

Geben Sie das Passwort für <alias_name> ein.

(EINGABETASTE, wenn Passwort dasselbe wie für Keystore):

Geben Sie das Passwort erneut ein:

[my-release-key.keystore wird gesichert.]

Publishing Checkliste

6. Anwendung signieren

„Release mode“

- Passenden privaten Schlüssel generieren
- Anwendung in release mode compilieren
- Fertige .apk signieren

Publishing Checkliste

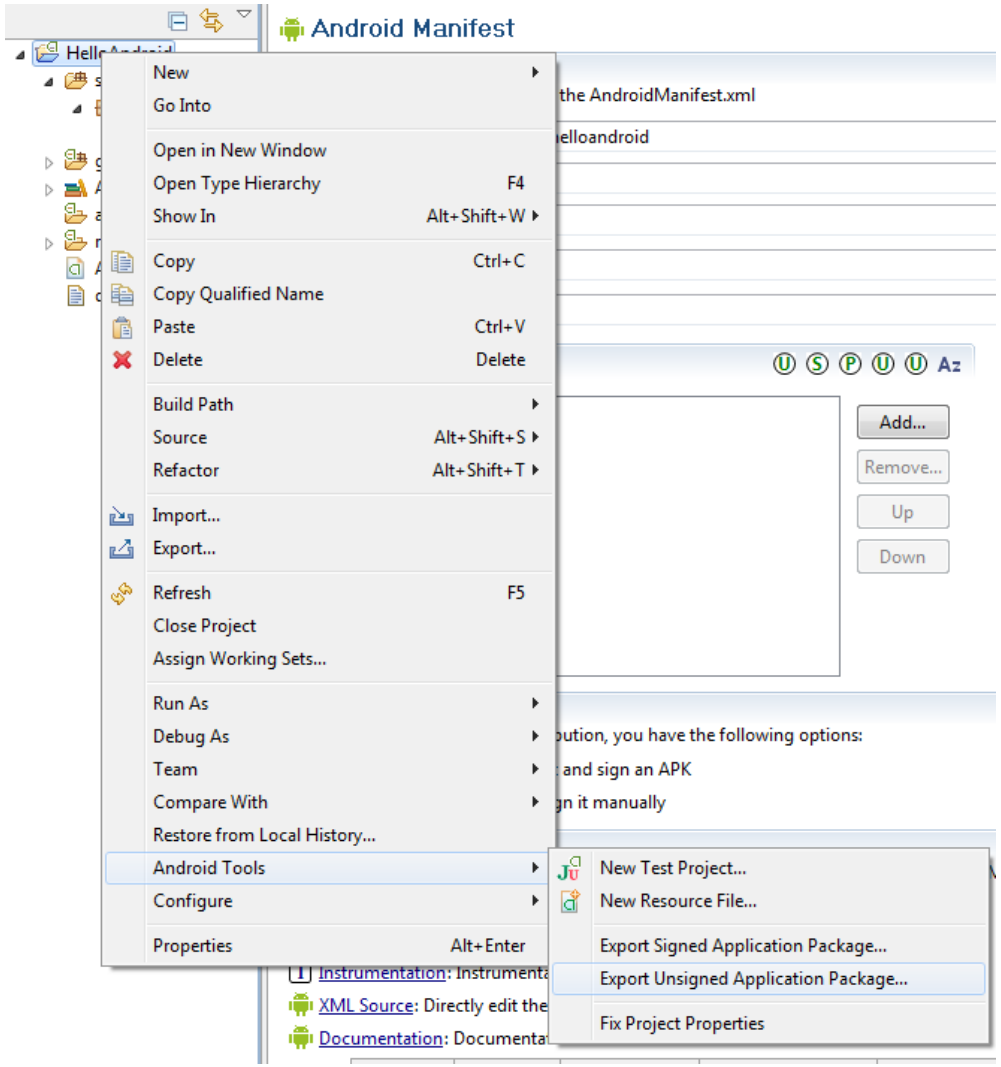
6. Anwendung signieren

Anwendung in release mode compilieren

In Eclipse:

Rechtsklick auf das Projekt in *Package Explorer* > *Android Tools* > *Export Unsigned Application Package*

Anwendung in release mode compilieren



Publishing Checkliste

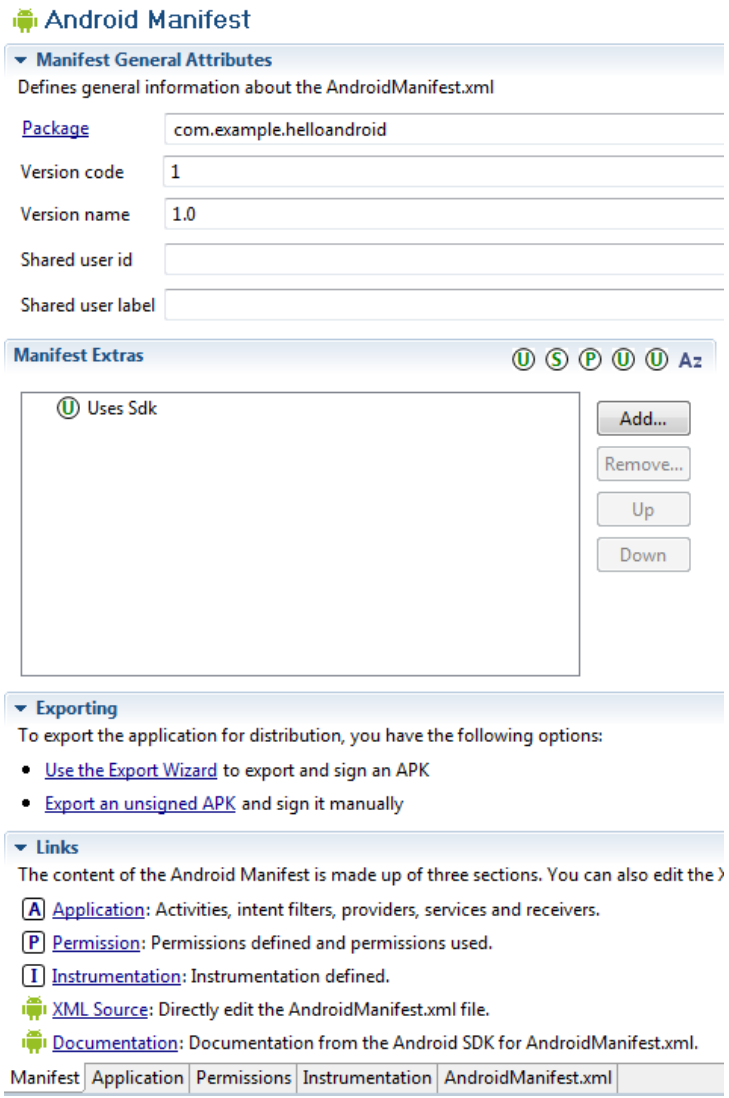
6. Anwendung signieren

Anwendung in release mode compilieren

Alternative

In *Android Manifest > Exporting > Export an unsigned APK*

Anwendung in release mode compilieren



Publishing Checkliste

6. Anwendung signieren

„Release mode“

- Passenden privaten Schlüssel generieren
- Anwendung in release mode compilieren
- Fertige .apk signieren

Publishing Checkliste

6. Anwendung signieren

Fertige .apk signieren

jarsigner.exe liegt in JavaSDK/bin Ordner

Shell:

jarsigner.exe -verbose -keystore my-release-key.keystore my_application.apk alias_name

Shell Ausgabe

```
PS C:\Program Files\Java\jdk1.6.0_18\bin> jarsigner.exe  -verbose -keystore my-release-key.keystore
        C:\HelloAndroid.apk alias_name
Enter Passphrase for keystore:
Enter key password for alias_name:
   adding: META-INF/ALIAS_NA.SF
   adding: META-INF/ALIAS_NA.RSA
  signing: res/layout/main.xml
  signing: AndroidManifest.xml
  signing: resources.arsc
  signing: res/drawable-hdpi/icon.png
  signing: res/drawable-ldpi/icon.png
  signing: res/drawable-mdpi/icon.png
  signing: classes.dex
```

Publishing Checkliste

6. Anwendung signieren

Signierte .APK optimieren

zipalign.exe liegt in AndroidSDK/tools Ordner

Shell:

```
zipalign -v 4 your_project_name-unaligned.apk
your_project_name.apk
```

Publishing Checkliste

6. Anwendung signieren

Es geht auch alles einfacher mit Eclipse ADT plugin ☺

Projekt in *Package Explorer* auswählen > *Export* > *Android* > *Export Android Application*.

Keystore kann erstellt oder ausgewählt werden

Publishing Checkliste

6. Anwendung signieren

„Debug mode“

- Hilfreich während der Entwicklungsphase

Publishing Checkliste

6. Anwendung signieren

„***Debug mode“***

- Hilfreich während der Entwicklungsphase
- 365 Tage gültig, danach neu erstellen

Publishing Checkliste

6. Anwendung signieren

„Debug mode“

- Hilfreich während der Entwicklungsphase
- 365 Tage gültig, danach neu erstellen
- Automatisches signieren in Debug Modus

Publishing Checkliste

6. Anwendung signieren

„Debug mode“

- Hilfreich während der Entwicklungsphase
- 365 Tage gültig, danach neu erstellen
- Automatisches Signieren in Debug Modus
- standardisiert

Keystore name:	"debug.keystore"
Keystore password:	"android"
Key alias:	"androiddebugkey"
Key password:	"android"
CN:	"CN=Android Debug,O=Android,C=US"

Index

Publishing Voraussetzungen

- Publishing Checkliste
  - Versionierung
  - Signierung

Publishing auf:

- Android Market
- Endgerät

Publishing auf Android Market

Voraussetzungen vom Android Market Server:

- Anwendung muss signiert sein und
  Ablaufdatum nach 22.10.2033
- *android:versionCode* und *android:versionName* müssen definiert sein
- *android:icon* und *android:label* müssen definiert sein

Publishing auf Android Market

Weitere Voraussetzungen:

- Google Konto
- Registrierung als Entwickler auf Android Market (25$)
- Eigene Webseite?

Index

Publishing Voraussetzungen

- Publishing Checkliste
  - Versionierung
  - Signierung

Publishing auf:

- Android Market
- Endgeräten

Publishing auf Endgeräten

Es gibt mehrere Möglichkeiten

- Adb.exe (Android SDK/tools)
- DDMS tool in Eclipse
- Kopieren der .apk Anwendung auf SD Karte

Publishing auf Endgeräten

Adb.exe (Android SDK/tools)

Shell:

adb.exe –e install Myapp.apk (Emulator)

adb.exe –d install Myapp.apk (USB Gerät)

Evtl.

im Android Gerät > *Einstellungen* > *Anwendungen*>

Unbekannte Quellen Aktivieren

Publishing auf Endgeräten

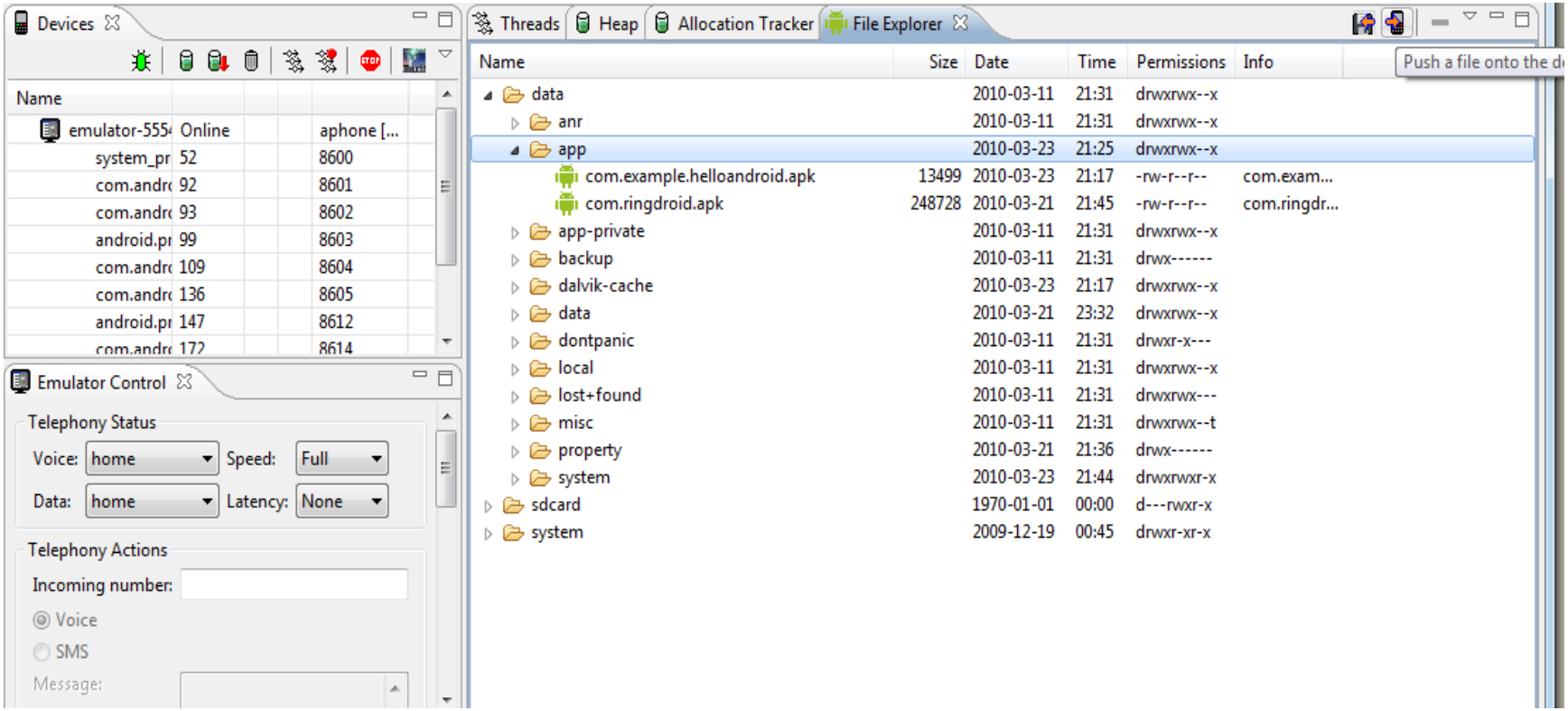
DDMS tool in Eclipse

DDMS Ansicht öffnen > Unter *Devices* Gerät auswählen > *File Explorer* > *Data/app* >

Knopf („*push a file onto the device*“) >

Fertige .apk Datei auswählen

Publishing auf Endgeräten



Zusammenfassung

Publishing Voraussetzungen

- Versionierung
  - Anwendungversionierung
  - SDK Versionen
- Signierung
  - Strategien
  - Möglichkeiten

Publishing

- Android Market
- Endgerät

Quellen

- http://www.android.com
- http://www.devx.com/wireless/Article/39972/1954
- http://www.androiddevelopment.org/2009/01/19/signing-an-android-application-for-real-life-mobile-device-usage-installation/

Ende

```
package com.example.helloandroid;

import android.app.Activity;
import android.os.Bundle;
import android.widget.TextView;

public class HelloAndroid extends Activity {
    public void onCreate(Bundle savedInstanceState) {

    super.onCreate(savedInstanceState);
      TextView tv = new TextView(this);
      tv.setText("Fragen?");
      setContentView(tv);
  }
}
```