Samsung Training Program

Digital Map-based Applications

Contents

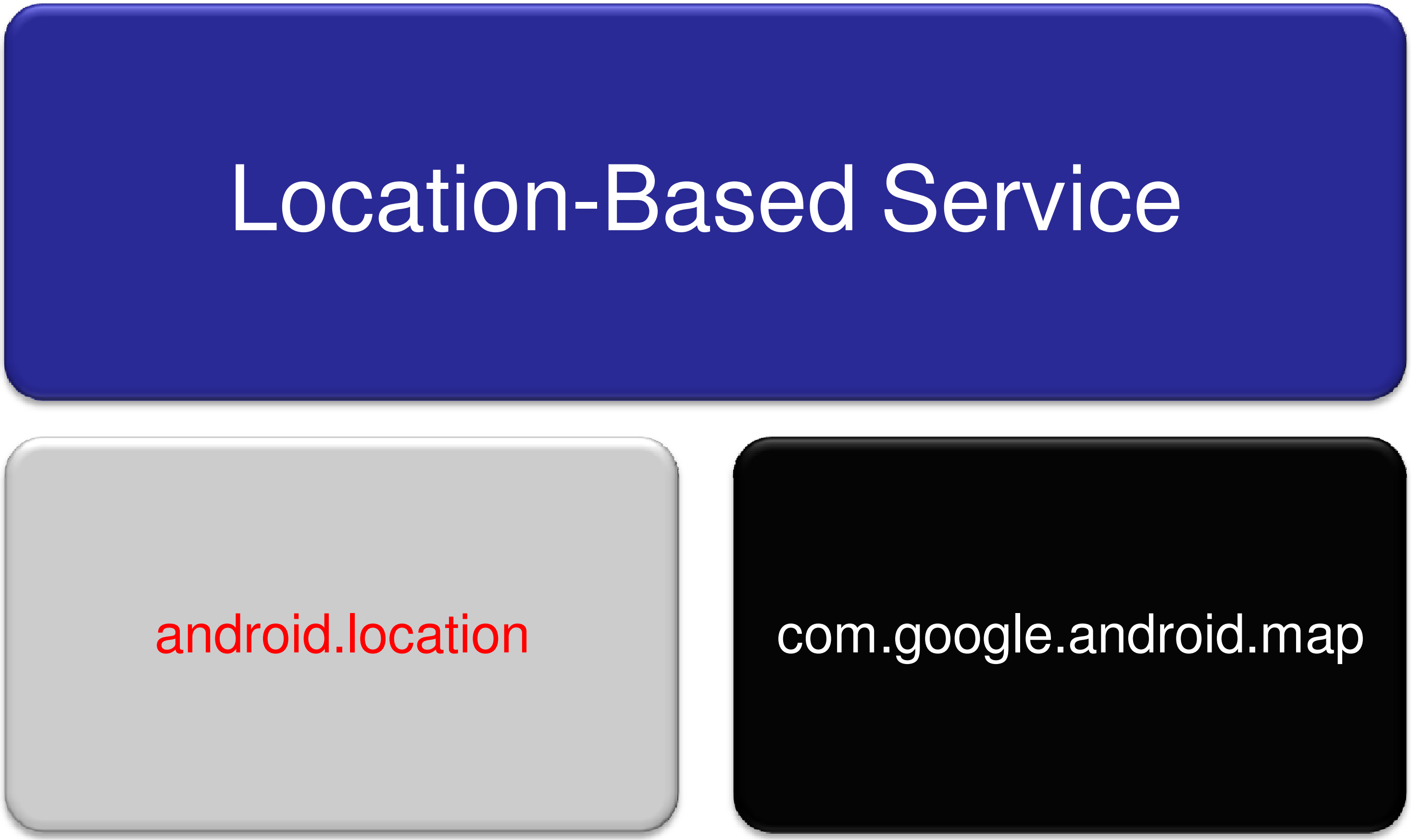
- Introduction
- Location services
- Google Map APIs
- Roadmap to integrate Map

Introduction

- Android provides a location framework that your application can use to determine the device's location and bearing and register for updates
- A Google Maps external library is available that lets you display and manage Maps data

Location-Based Services

The Location-Based API includes two packages android.location & com.google.android.maps that provide an initial look at the support in the Android platform for building location-based services.



Location-Based Service

android.location

com.google.android.map

Location services

- Access to the location services is provided through the classes in the android.location package. The central component of the location framework is the LocationManager system service, which provides APIs to determine location and bearing
- Google provides a Maps external library that includes the com.google.android.maps package. The classes offer built-in downloading, rendering, and caching of Maps tiles, as well as a variety of display options and controls.

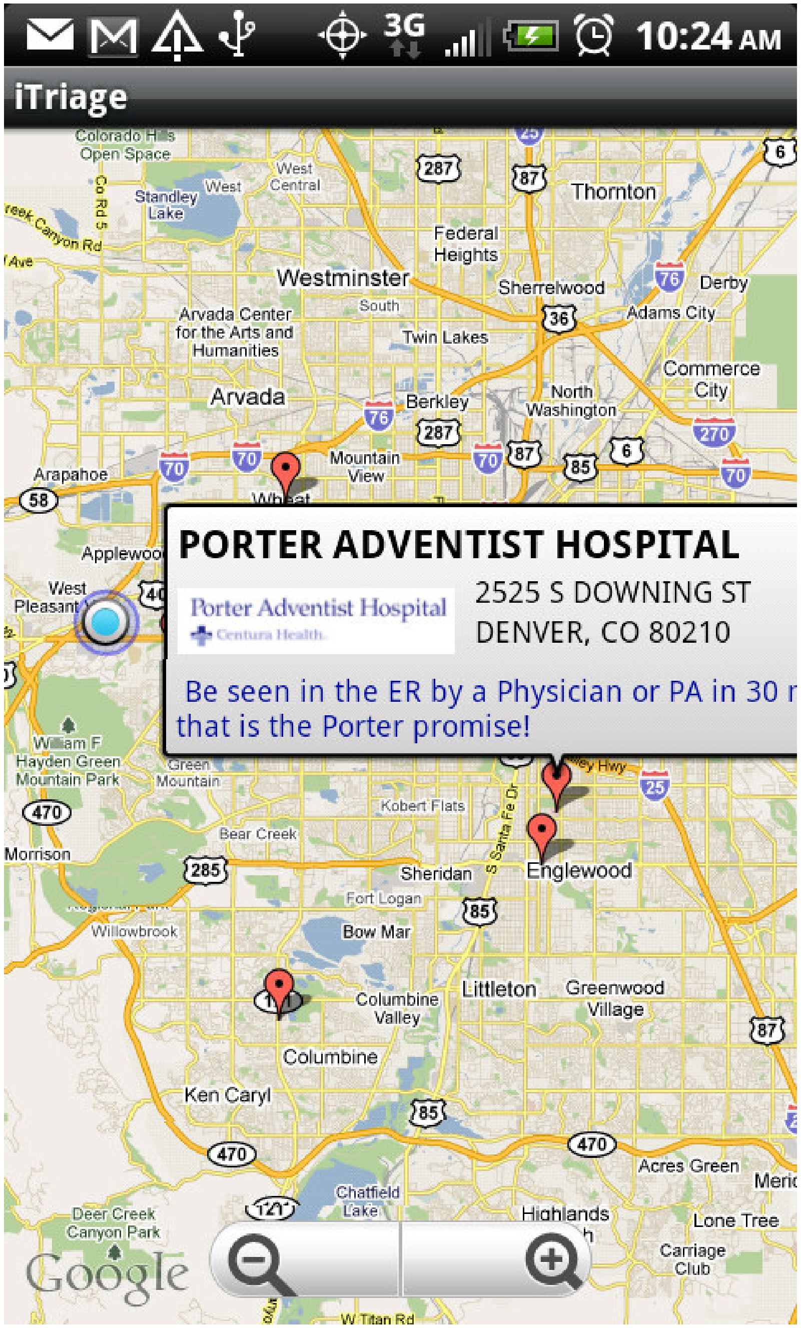
Tool Prerequisites

- Eclipse with Android SDK and AVD Manager
- Google API:

  http://code.google.com/android/add-ons/google-apis
- Java
- XML
- Android device[or Emulator] to test

MapView

- Adding Google APIs
- Obtaining Maps Key
- Using MapActivity
- Showing current location
- Adding Overlay items



PORTER ADVENTIST HOSPITAL

Porter Adventist Hospital
Centura Health

2525 S DOWNING ST
DENVER, CO 80210

Be seen in the ER by a Physician or PA in 30 r
that is the Porter promise!

MapView: Obtaining Maps Key

- Generate Certificate
- keytool -genkey -v -keystore droid_kids.keystore -alias droid_kids -keyalg RSA -keysize 2048 -validity 10000

- Get Fingerprint of the certificate

  keytool -list -alias droid_kids -keystore droid_kids.keystore

  Fingerprint is :

  E0:9D:58:9F:B2:CD:5C:9C:42:8B:60:0F:23:BC:24:11

- Register the fingerprint with Google Maps Service

http://code.google.com/android/maps-api-signup.html

Key is: 0zu3JWra9vK5LxswGR1V4Wh3SztudX-UjgLWLJA

MapView: Create Maps Activity

- Include Maps library

```
<uses-library android:name="com.google.android.maps" />
```

- Add permissions

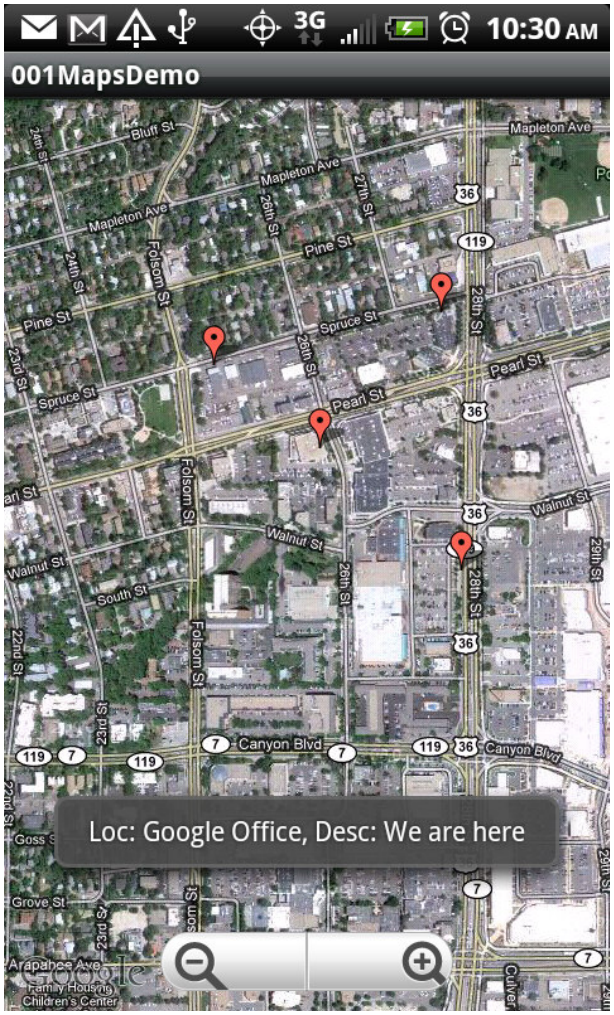
```
<uses-permission android:name="android.permission.INTERNET" />
<uses-permission android:name="android.permission.ACCESS_FINE_LOCATION"
/>
<uses-permission
android:name="android.permission.ACCESS_COARSE_LOCATION" />
```

- Create a MapView layout and add the key

```
<com.google.android.maps.MapView
        xmlns:android="http://schemas.android.com/apk/res/android"
        android:id="@+id/mapview"
        android:layout_width="fill_parent"   android:layout_height="fill_parent"
        android:clickable="true"
                              ="
          "/>
```

MapView Coding

- Layers
- Zoom
- Center
- Overlays
- MyLocation



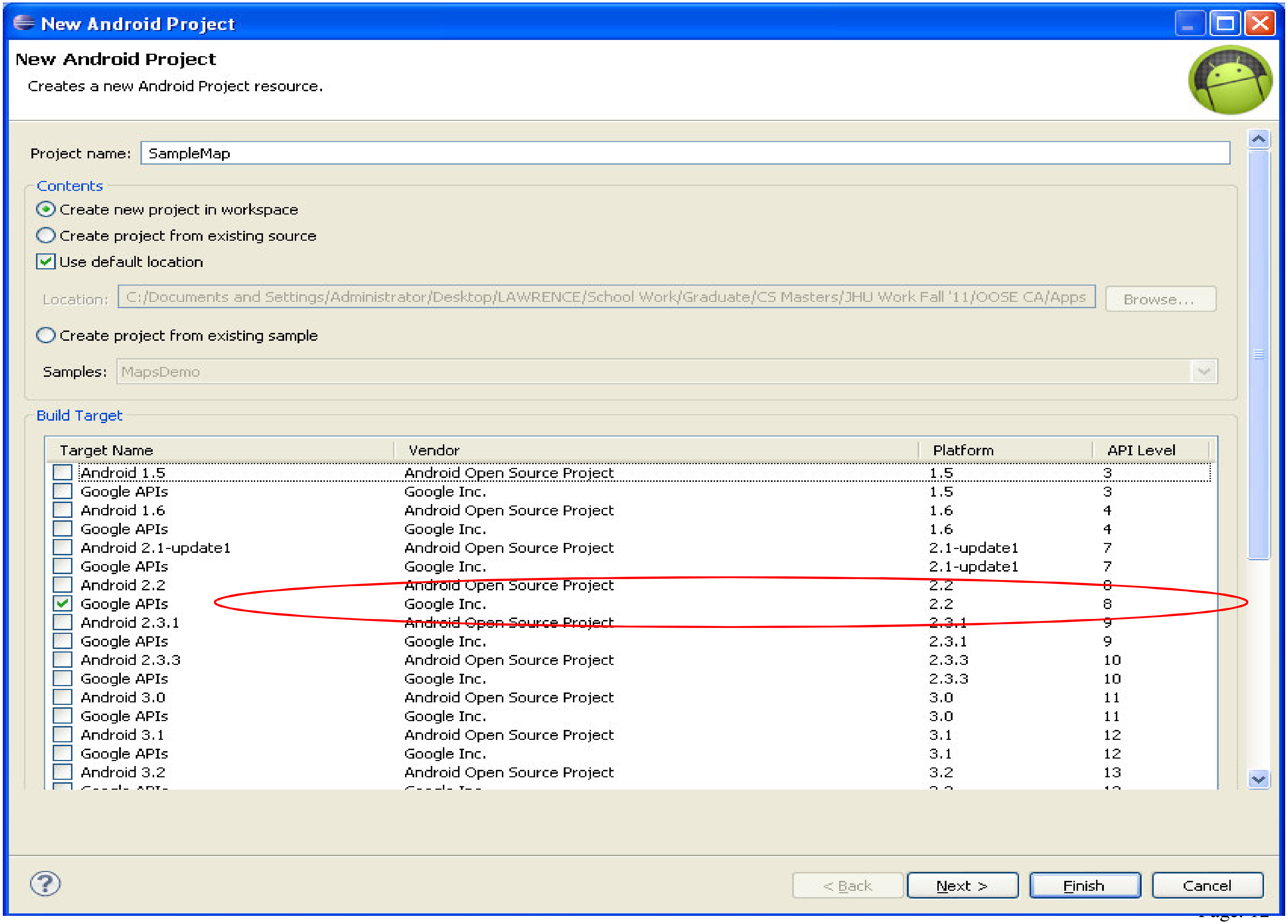
001MapsDemo

Loc: Google Office, Desc: We are here

Steps in creating apps

- **Setup project to use ‘Google API’ version**
- **Edit Manifest file**
  - To indicate the app will use maps and the internet
- **Get a maps API key**
- **Note: Google Maps API can display a map and draw overlays, but is not the full Google Maps experience you enjoy on the web**
  - For example, there does not seem to be inherent support for drawing smooth routes between points…however, you can draw lines between points and almost any type of overlay, but that’s different than street routes

Project setup



New Android Project

New Android Project

Creates a new Android Project resource.

Project name: SampleMap

Contents

● Create new project in workspace

○ Create project from existing source

☑ Use default location

Location: C:/Documents and Settings/Administrator/Desktop/LAWRENCE/School Work/Graduate/CS Masters/JHU Work Fall '11/OOSE CA/Apps

○ Create project from existing sample

Samples: MapsDemo

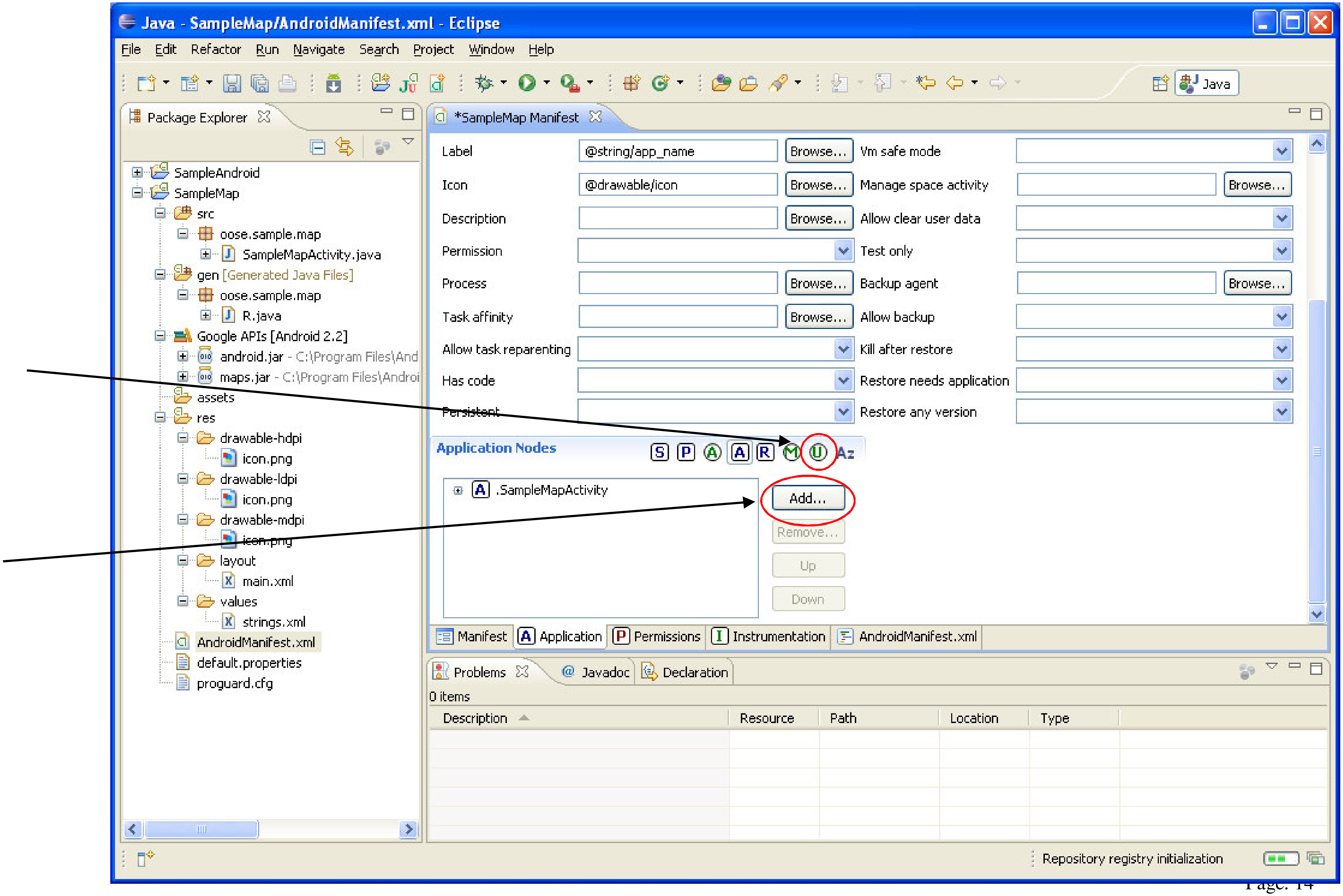
Build Target

	Target Name	Vendor	Platform	API Level
☐	Android 1.5	Android Open Source Project	1.5	3
☐	Google APIs	Google Inc.	1.5	3
☐	Android 1.6	Android Open Source Project	1.6	4
☐	Google APIs	Google Inc.	1.6	4
☐	Android 2.1-update1	Android Open Source Project	2.1-update1	7
☐	Google APIs	Google Inc.	2.1-update1	7
☐	Android 2.2	Android Open Source Project	2.2	8
☑	Google APIs	Google Inc.	2.2	8
☐	Android 2.3.1	Android Open Source Project	2.3.1	9
☐	Google APIs	Google Inc.	2.3.1	9
☐	Android 2.3.3	Android Open Source Project	2.3.3	10
☐	Google APIs	Google Inc.	2.3.3	10
☐	Android 3.0	Android Open Source Project	3.0	11
☐	Google APIs	Google Inc.	3.0	11
☐	Android 3.1	Android Open Source Project	3.1	12
☐	Google APIs	Google Inc.	3.1	12
☐	Android 3.2	Android Open Source Project	3.2	13

Manifest manipulation (1)

- **Open Manifest file**
- **Add map library tag**
  - Add the ‘Uses Library’ com.google.android.maps
- **Indicate the app will access the internet**
  - Add the ‘Permission’ android.permission.INTERNET
- **End goal is to add the following two lines to XML file, under the <manifest> and <application tags>, respectively**
  - Under the <manifest> tag
    - <uses-permission android:name=*"android.permission.INTERNET"*></uses-permission>
  - Under the <application> tag
    - <uses-library android:name=*"com.google.android.maps"*></uses-library>
- **Following is GUI way to add them**

Manifest manipulation (2)



Label: @string/app_name

Icon: @drawable/icon

Description:

Permission:

Process:

Task affinity:

Allow task reparenting:

Has code:

Persistent:

Vm safe mode:

Manage space activity:

Allow clear user data:

Test only:

Backup agent:

Allow backup:

Kill after restore:

Restore needs application:

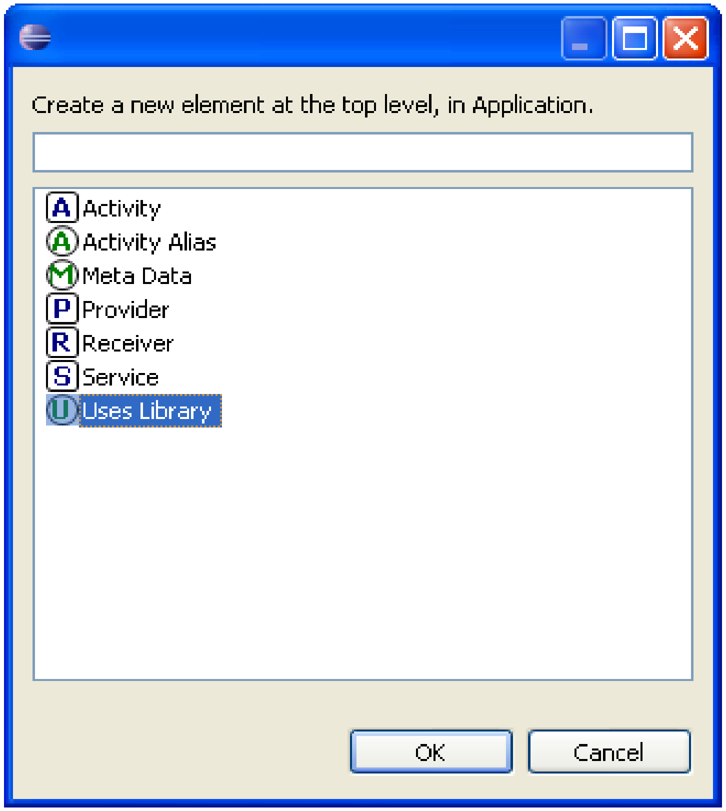
Restore any version:

0 items

Description	Resource	Path	Location	Type

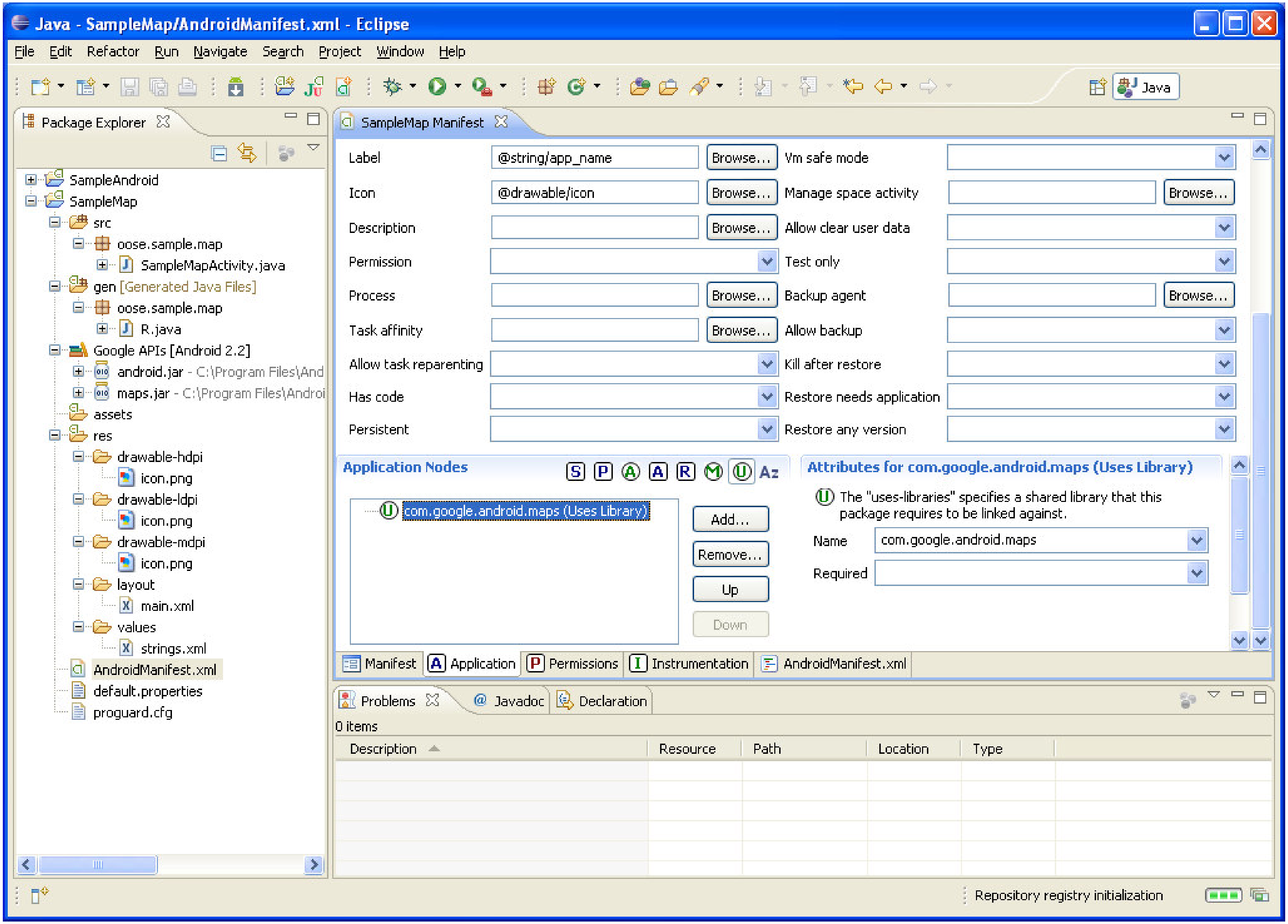
Manifest manipulation (3)

- Select ‘Add’ under ‘Uses Library’ (last slide)
- Then select ‘Uses Library at this prompt
- Set name as: com.google.android.maps (next slide) and save



Create a new element at the top level, in Application.

Manifest manipulation (4)



Label: @string/app_name

Icon: @drawable/icon

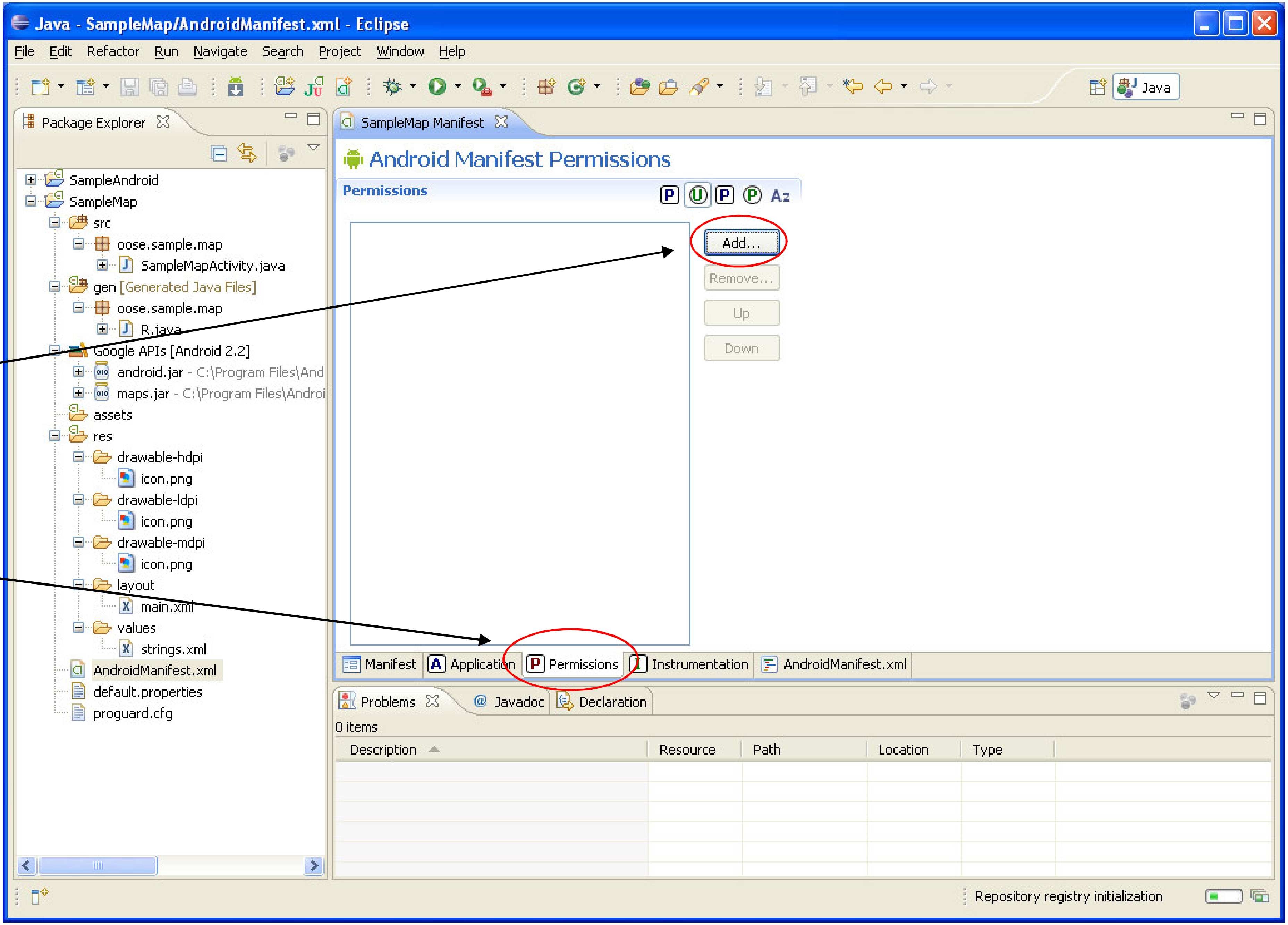
Attributes for com.google.android.maps (Uses Library)

The "uses-libraries" specifies a shared library that this package requires to be linked against.

Name: com.google.android.maps

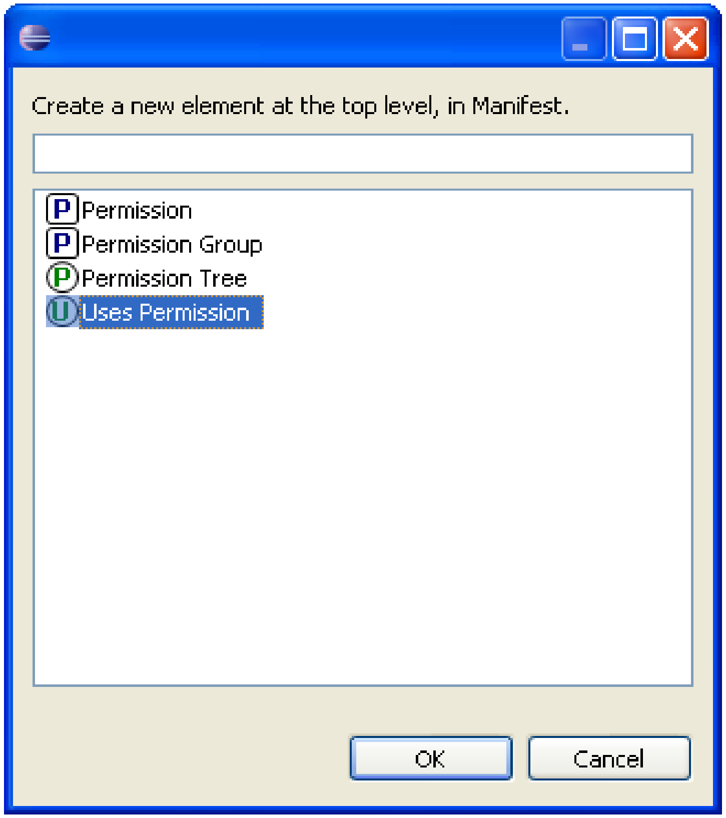
0 items

Manifest manipulation (5)



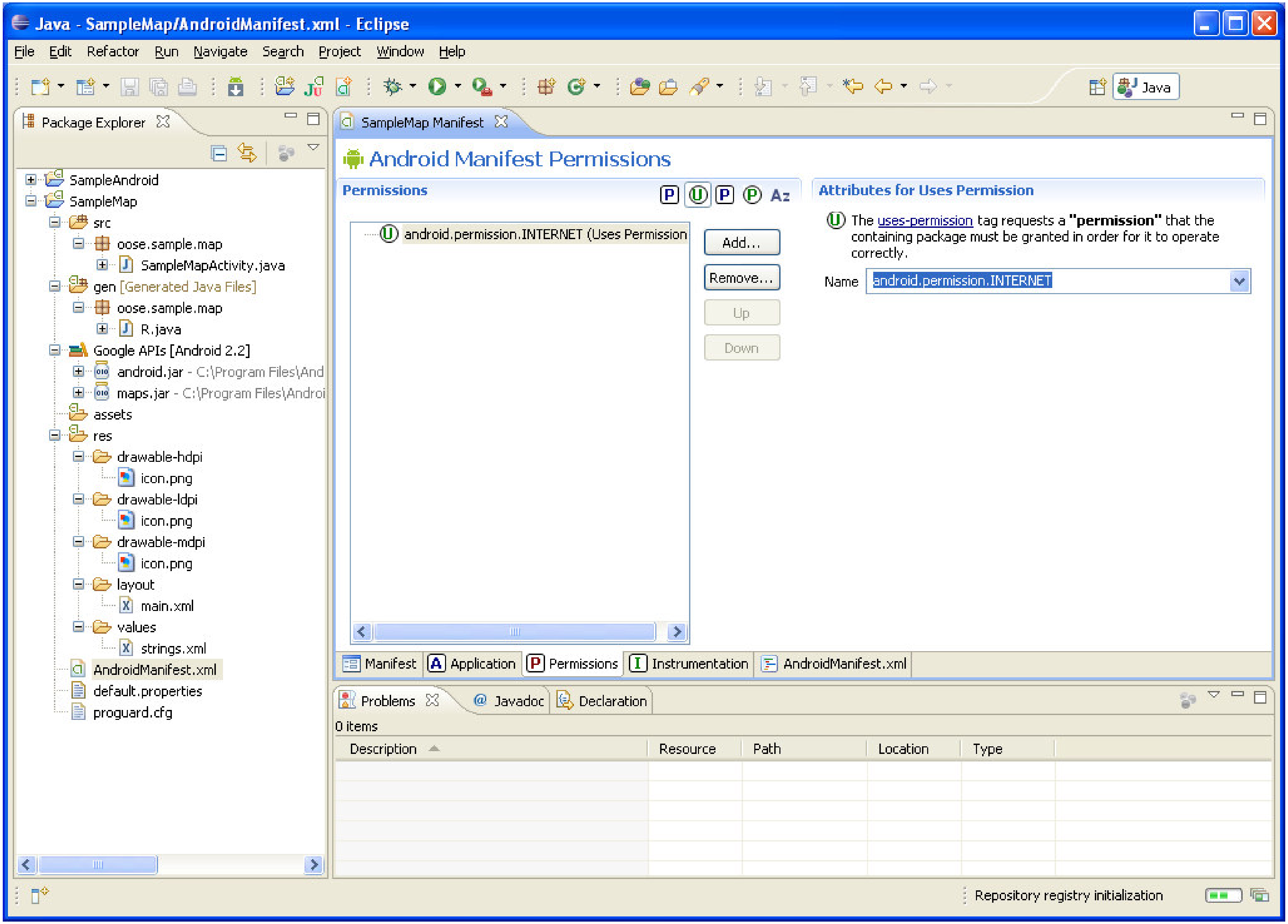
Manifest manipulation (6)

- Select ‘Permissions’ and then ‘Add’ (last slide)
- Select ‘Uses Permissions’ at this prompt
- Set name to: android.permission.INTERNET and save (next slide)



Create a new element at the top level, in Manifest.

Manifest manipulation (7)



Android Manifest Permissions

Permissions

android.permission.INTERNET (Uses Permission

Attributes for Uses Permission

The uses-permission tag requests a **"permission"** that the containing package must be granted in order for it to operate correctly.

Name android.permission.INTERNET

0 items

Maps API Key (1)

- **All Android applications need to be signed**
  - The debug mode signs for you with special debug certificate
- **All MapView elements in map applications need to have an API key associated with them**
  - That key must be registered with the certificate used to sign the app
- **When releasing app, need to sign with a release certificate and get a new API Key**

Maps API Key (2)

- **For debug mode, get the MD5 fingerprint of the debug certificate**
  - Locate the ‘keystore’
    - Windows XP: C:\Documents and Settings\<user>\.android\debug.keystore
    - Linux: ~/.android/debug.keystore
  - Use Keytool (comes with Java, in the bin directory with the other Java tools, should put that dir on system PATH) to get fingerprint
    - keytool -list –v -alias androiddebugkey -keystore “<path_to_debug_keystore>” -storepass android -keypass android
      - If don’t include –v option, then will probably get only 1 fingerprint, and if it’s not MD5, then need –v (Java 7 needs –v)
    - Extract the MD5 fingerprint, SHA will not work unfortunately
- **Go to https://code.google. com/android/maps-api-signup.html, agree to terms and paste MD5 fingerprint, you will then be given an API Key**

MapView: XML

- **Need to put MapView tag in XML**
  - com.google.android.maps.MapView
  - MapView is the basic view that represents a Google Map display
  - Must include API Key in XML, inside a layout

    ```
    <com.google.android.maps.MapView
        android:id="@+id/mapview"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:clickable="true"
        android:apiKey=“<api key>”/>
    ```
- **Maps API Reference**

  http://code.google.com/android/add-ons/google-apis/reference/index.html

Exercises

- **Tutorial 18: Putting Lunch on the Map**