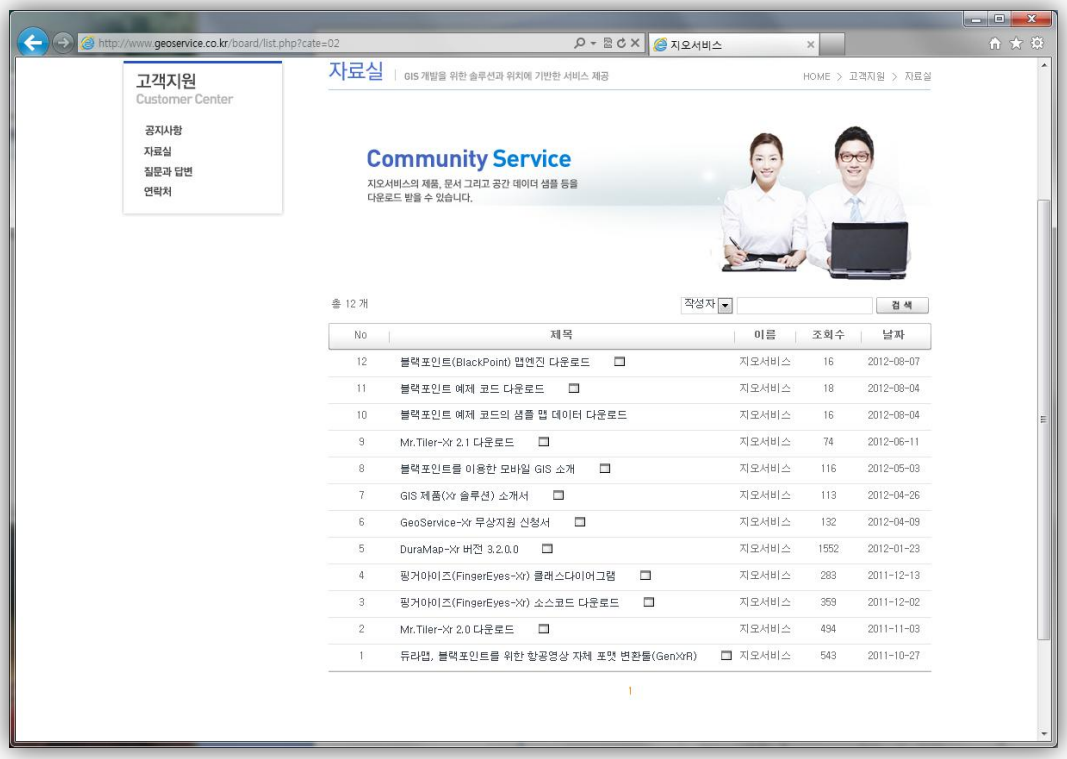
1 블랙포인트 튜토리얼

1.1 블랙포인트 다운로드

블랙포인트는 안드로이드 기반에서 GIS 시스템 개발을 위한 맵 엔진입니다. 자바 언어로 개발되어 있으며 jar 파일 형태로 제공됩니다. 블랙포인트에 대한 jar 파일은 블랙포인트의 개발 회사인 지오서비스의 자료실에서 다운로드 받을 수 있습니다.

http://www.geoservice.co.kr/board/list.php?cate=02



지오서비스 자료실에서 블랙포인트 다운로드

또한 이 자료실에는 이 책에서 직접 만들어볼 예제들에 대한 완성된 샘플 코드 및 공간 데이터도 함께 다운로드 받을 수 있습니다.

블랙포인트는 개인 개발자 및 회사에서 모바일 GIS 시스템 개발을 위해 많이 사용되고 있는 맵 엔진입니다. 많은 개발자들의 피드백(feedback)을 통해 계속 발전해 가고 있습니다. 주기적으로 위의 사이트에 들려서 블랙포인트의 최신 버전을 다운로드 받으시기 바랍니다.

1.2 블랙포인트 라이선스

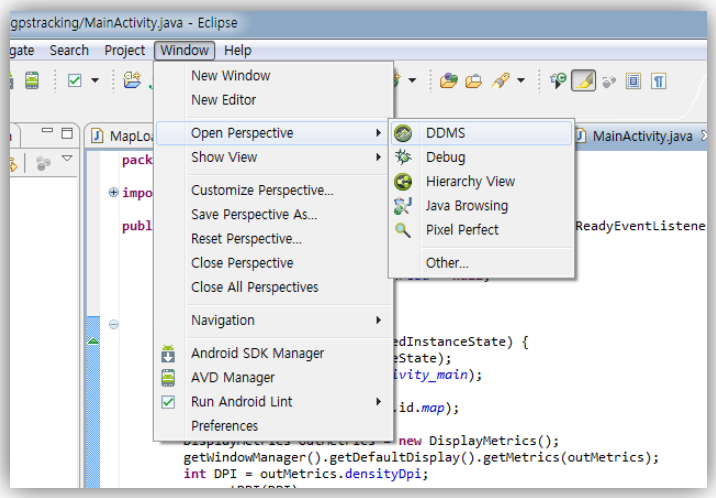
블랙포인트를 이용해 개발된 앱(App)이 상업적으로 판매되는 형태가 아니라면 자유롭게 라이선스 비용 없이 사용할 수 있습니다. 또한 개발에 궁금한 부분이나 기술 지원에 대해서 자유롭게 지원받을 수 있습니다.

그러나 블랙포인트를 이용해 개발된 앱이 상업적인 목적으로 판매 또는 제공되는 경우에 일정한 라이선스 비용을 지오서비스에 지불해야 합니다. 블랙포인트에 대한 상업적인 사용에 대한 라이선스 문의는 이메일(hjkim@geoservice.co.kr)을 통해 문의하시기 바랍니다.

1.3 개발 시작하기(Hello World !!)

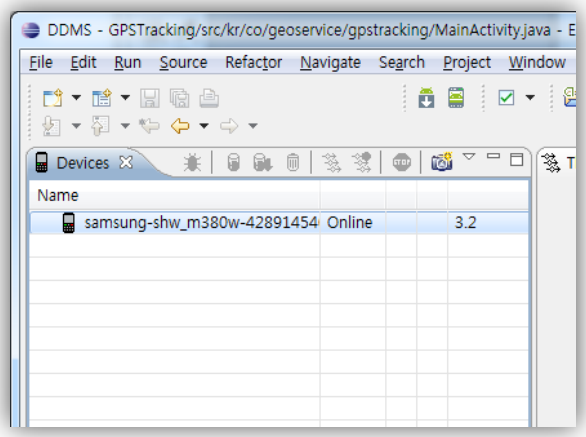
블랙포인트를 처음 사용해 보는 것으로써 간단한 애플리케이션을 만들어 보도록 하겠습니다. 안드로이드 개발은 이클립스(Eclipse)에서 이루어집니다.

블랙포인트를 이용한 실제 개발에 앞서 독자 여러분이 사용하는 디바이스에 설치된 안드로이드 운영체제의 버전을 확인해 보도록 하겠습니다. USB로 디바이스를 PC에 연결하고 이클립스 개발환경의 우측 상단에 DDMS를 클릭하거나 아래처럼 [Windows]-[Open Perspective]-[DDMS] 메뉴를 실행합니다. 만약 DDMS 메뉴가 보이지 않는다면 [Windows]-[Open Perspective]-[Other..] 메뉴를 통해 직접 DDMS 항목을 선택해도 됩니다.



DDMS로 전환

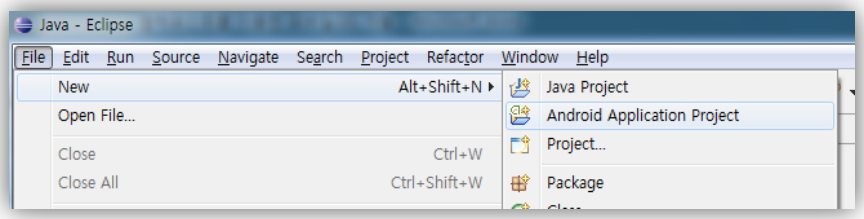
DDMS가 실행되면 전환된 화면의 좌측에 Deveice 정보에서 현재 USB에 연결된 디바이스의 버전을 볼 수 있습니다. 아래의 경우 3.2로 표시되어 있습니다.



개발 디바이스의 안드로이드 운영체제의 버전 확인

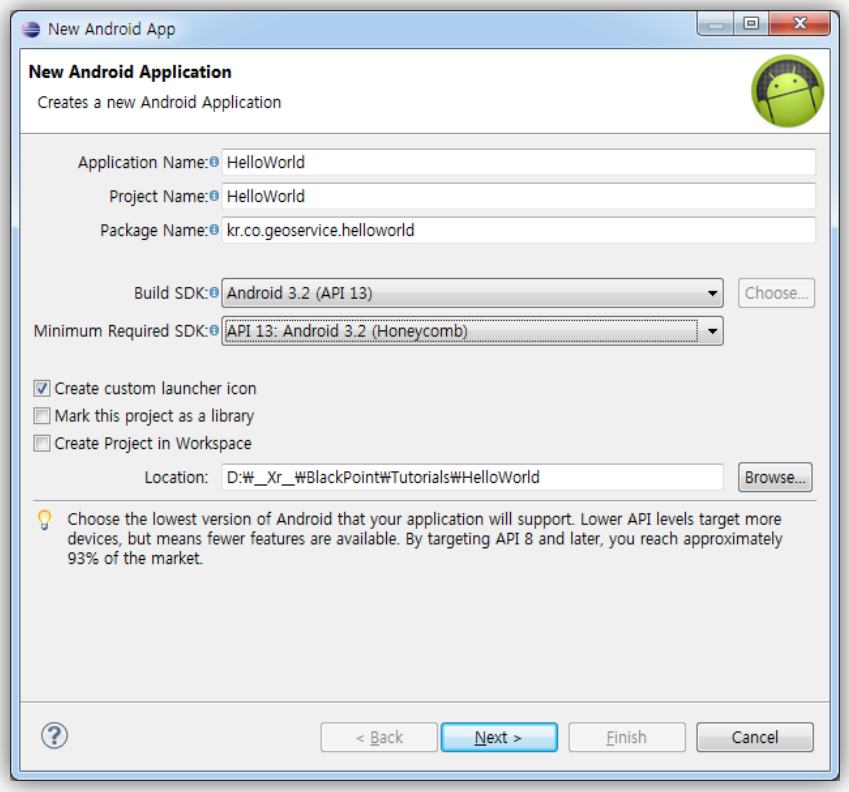
이제 디바이스의 안드로이드 운영체제를 확인했으니 이제 본격적으로 블랙포인트를 이용해 간단한 앱을 만들어 보겠습니다.

먼저 이클립스가 실행되어 있지 않다면 먼저 실행하고 [File]-[New]-[Android Application Project] 메뉴를 실행합니다. 또는 [File]-[New]-[Projects] 메뉴를 통해 Android Application Project 항목을 선택 해야 하는 경우도 있습니다.



[File]-[New]-[Android Project] 메뉴

해당 메뉴를 실행하면 다음과 같은 대화상자가 나타납니다. 단, 안드로이드 개발 SDK가 지속적으로 업데이트됨으로써 대화상자가 다를 수 있습니다. 하지만 큰 흐름이 비슷하므로 새로운 프로젝트 생성에 큰 어려움은 없습니다.

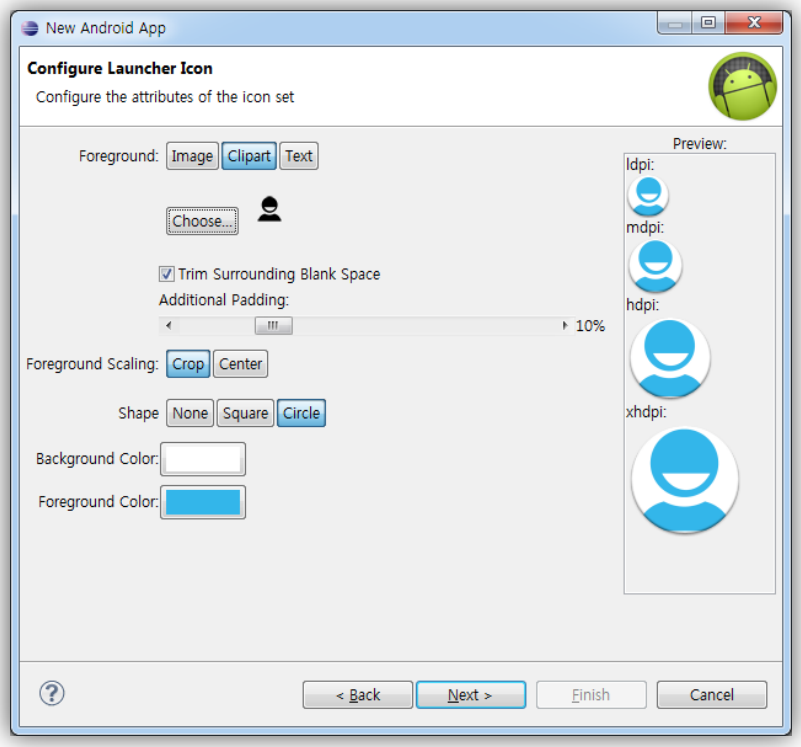


New Android App 대화상자

Application Name을 HelloWorld라고 입력합니다. 그리고 Package Name은 원하는 값으로 지정합니다. Package Name은 주로 홈페이지의 URL을 이용해 지정합니다. 예를 들어서 www.homepage.net이라는 회사라면 net.homepage.helloworld와 같이 구성합니다. 다음은 Build SDK를 지정합니다. 여기서는 갤럭시탭 10.1을 사용했습니다.

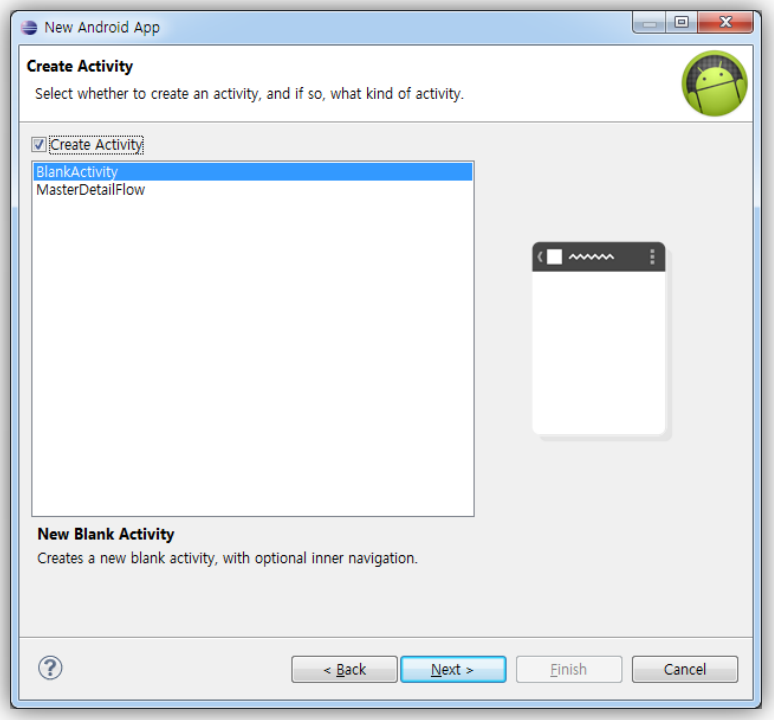
바로 앞서 살펴보았듯이 필자가 사용하는 갤럭시탭 10.1에서 사용 가능한 SDK는 3.2인 Honeycomb이므로 이에 맞게 New Android App 대화상자의 Build SDK를 선택합니다.

Minimum Required SDK에는 앞서 선택한 SDK와 같거나 보다 낮은 버전을 선택합니다. 여기서는 Build SDK와 동일한 SDK를 선택했습니다. Location은 독자가 원하는 폴더로 지정합니다. 그리고 Next 버튼을 누릅니다. 이제 다음과 같은 창이 나타나는데 앱의 아이콘과 전경색 및 배경색을 지정하는 화면입니다. 기본값을 유지하고 Next 버튼을 누릅니다.



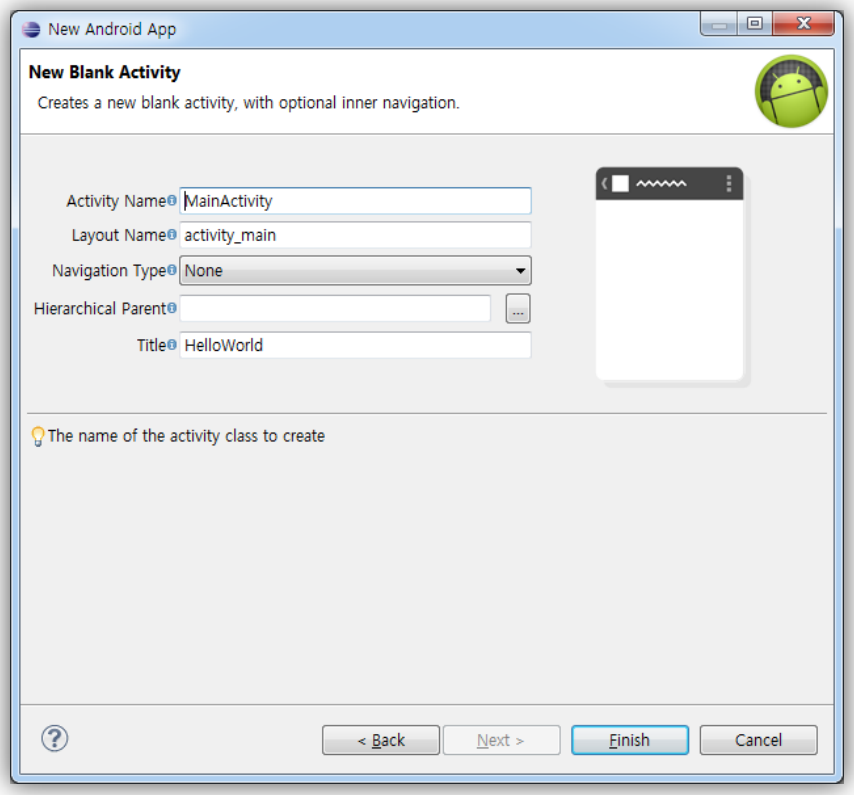
New Android App 대화상자에서 아이콘 설정

또 다시 다음과 같은 대화상자가 나타나며 생성할 Activity의 종류를 선택할 수 있습니다. 기본적인 BlankActivity를 선택하고 Next 버튼을 누릅니다.



New Android App 대화상자에서 Activity 종류 선택

이제 앞서 선택한 Activity에 대한 클래스 이름 등을 지정하는 대화상자가 나타납니다. 다음처럼 Title에 HelloWorld을 입력하고 Finish를 클릭합니다.

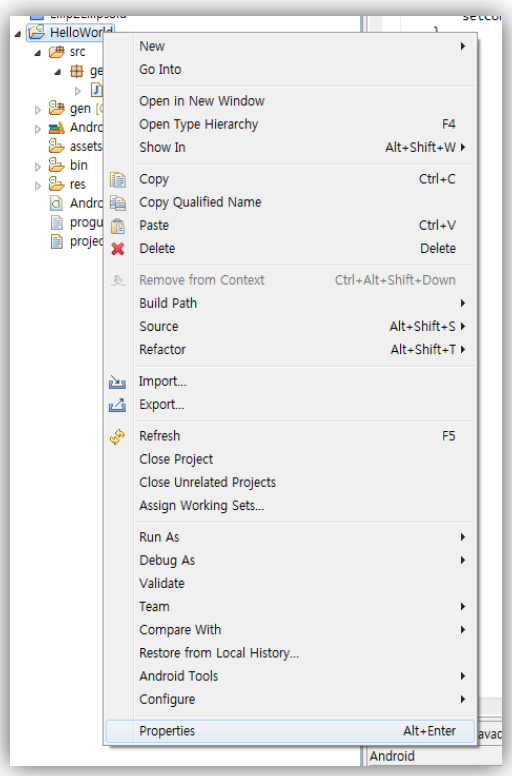


New Android App에서 Activity 정보 지정

이제 프로젝트 생성이 완료되었습니다.

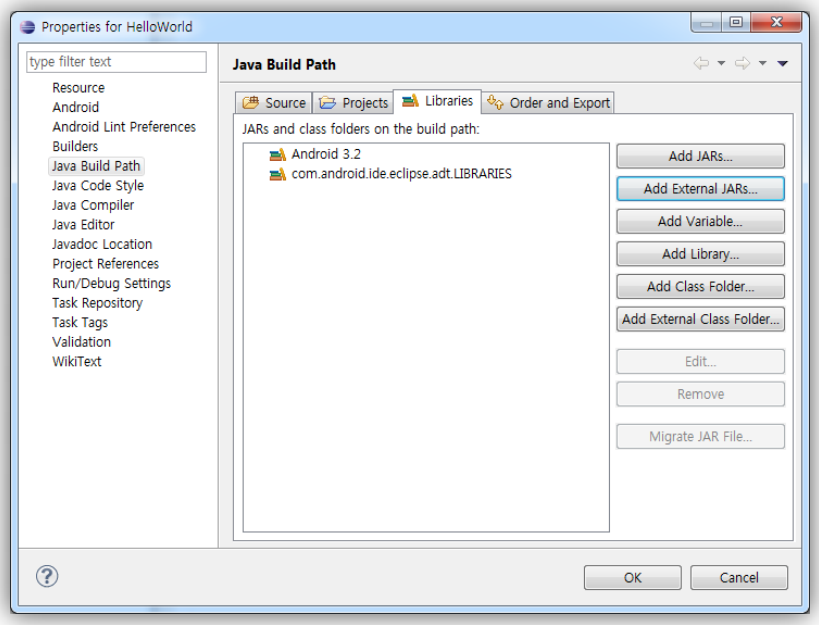
만약 이클립스가 현재 DDMS 모드라면 [Window]-[Open Perspective]-[Java] 메뉴를 실행시킵니다. 이클립스의 Package Explorer에 HelloWorld 프로젝트 폴더가 생성된 것을 볼 수 있습니다. 만약 Package Explorer가 표시되어 있지 않다면 [Window]-[Show View]-[Package Explorer]를 실행해 Package Explorer를 표시하도록 합니다.

Package Explorer에서 HelloWorld 폴더에서 마우스 오른쪽 버튼을 클릭해 팝업 메뉴를 나타나게 하고 Properties 메뉴를 실행합니다. 또는 HelloWorld 폴더를 선택하고 단축키로 Alt+Enter를 누릅니다.



HelloWorld 프로젝트 속성 설정 팝업 메뉴

Properties 메뉴를 실행하면 다음처럼 HelloWorld 프로젝트의 속성을 설정할 수 있는 대화상자가 나타나며 여기서 블랙포인트 사용을 위한 라이브러리 파일들을 등록해야 합니다.

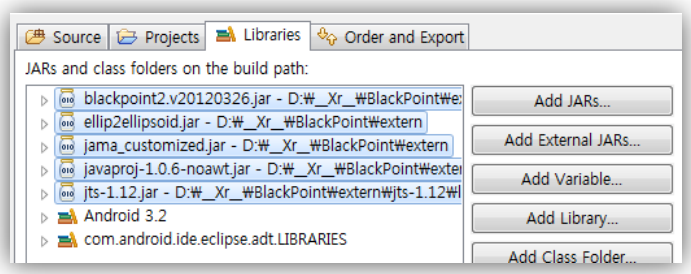


HelloWorld 프로젝트 속성 설정 대화상자

위의 대화상자에서 좌측의 목록에서 Java Build Path를 선택한 후 Libraries 탭을 선택해 나타나는 대화상자에서 오른쪽의 Add External JARs 버튼을 클릭하여 다음 5개의 jar 파일을 추가합니다. 이 jar 파일들은 지오서비스의 데이터실에서 다운로드 받을 수 있습니다.

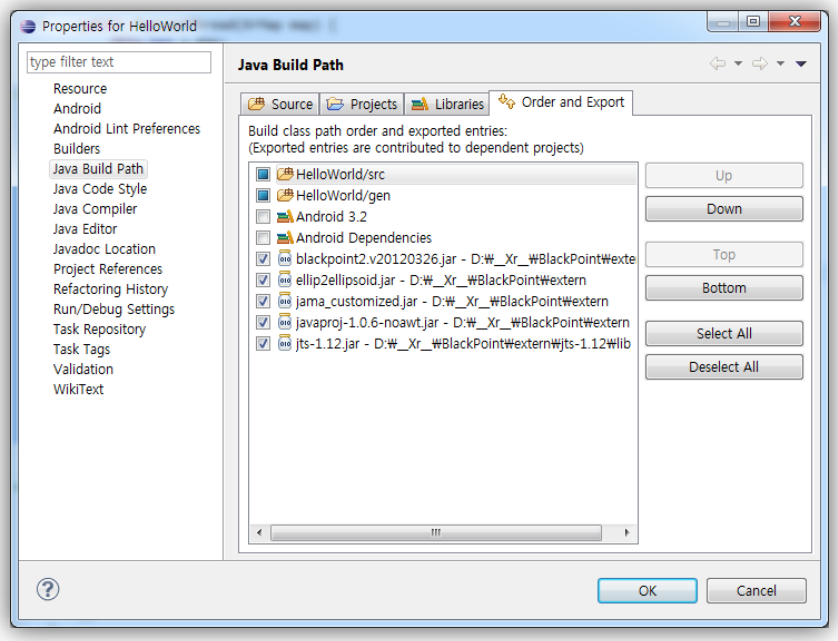
1. blackpoint2.v20120326.jar (버전 넘버는 다를 수 있습니다)
2. ellip2ellipsoid.jar
3. jama_customized.jar
4. javaproj-1.0.6-noawt.jar
5. jts-1.12.jar

5개의 jar 파일들을 추가하면 대화상자가 다음과 같이 변경된 것을 볼 수 있습니다.



HelloWorld 프로젝트 속성 설정 대화상자 중 라이브러리 설정

다음으로 이 대화상자에서 Order and Export 탭을 선택하여 대화상자를 다음처럼 변경합니다.

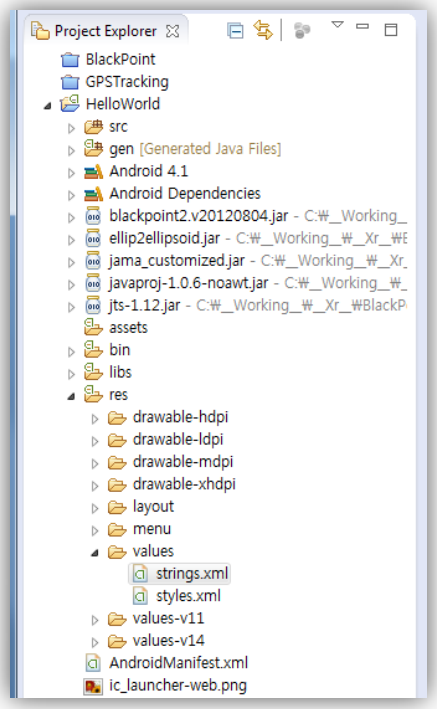


HelloWorld 프로젝트 속성 설정 대화상자 중 Order and Export 설정

위의 대화상자에서처럼 블랙포인트의 라이브러리 파일들(blackpoint2.v20120326.jar, ellip2ellipsoid.jar, jama_customized.jar, javaproj-1.0.6-noawt.jar, jts-1.12.jar)을 모두 체크합니다.

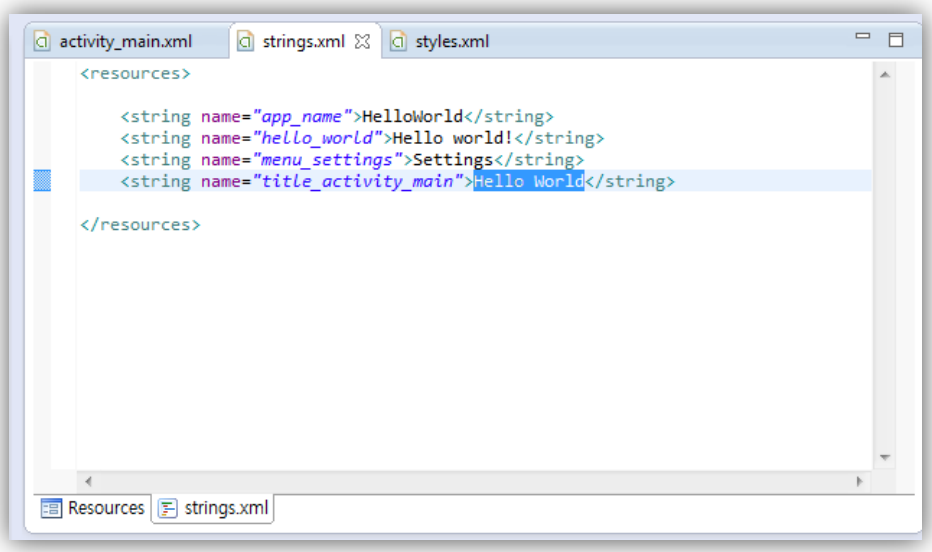
이제 OK 버튼을 눌러 프로젝트 설정을 종료합니다.

이제 앱의 타이틀을 Hello World로 변경하도록 하겠습니다. Project Explorer에서 HelloWorld의 서브 폴더 중 res를 더블클릭해서 res 폴더 안의 내용을 살펴봅니다. 이 res 폴더 안에 values라는 서브 폴더가 존재하며 이 values 폴더 안의 strings.xml을 더블클릭해 strings.xml을 오픈합니다.



앱의 타이틀을 변경하기 위해 Strings.xml 파일을 선택

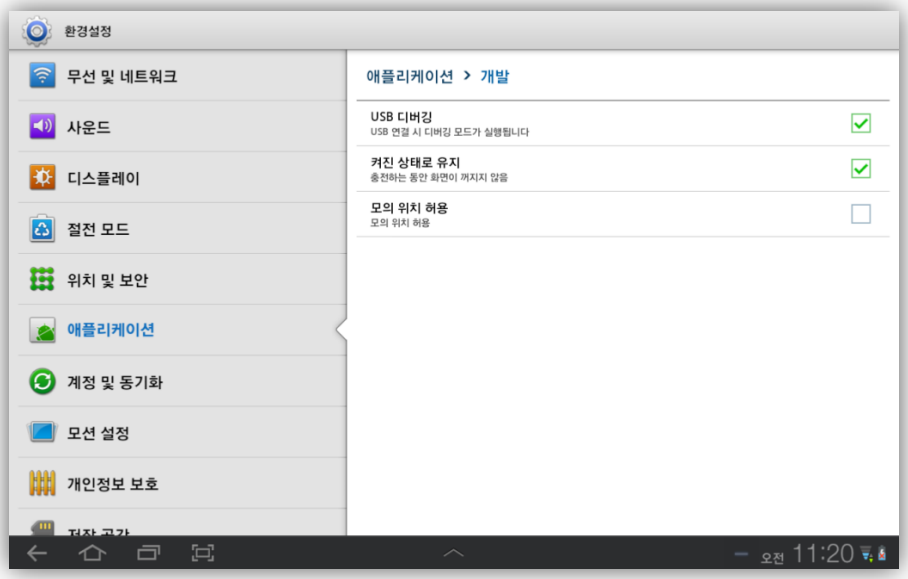
Strings.xml을 오픈하면 아래의 변경된 화면의 중간쯤에 Resources 탭과 strings.xml 탭이 보일 것입니다. 이 중에서 strings.xml 탭을 선택해 직접 strings.xml을 편집할 수 있도록 합니다.



strings.xml 탭을 선택해 직접 파일을 편집

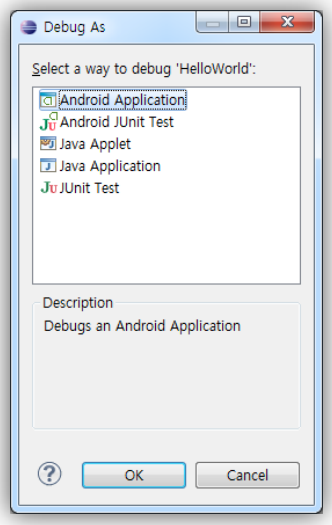
표시된 strings.xml 파일의 내용 중 위의 화면처럼 name이 title_activity_main인 string 요소의 값을 Hello World로 변경합니다.

이제 플리케이션을 실행하기에 앞서 모바일 디바이스가 USB 디버깅이 가능하도록 설정해야 합니다. 자신의 모바일 디바이스에서 환경설정을 실행합니다. 실행된 환경설정에서 개발관련 옵션을 살펴보면 USB 디버깅을 체크하는 항목이 있습니다. 이클립스에서 개발된 앱을 USB에 연결된 디바이스에 전송해 디버깅하기 위해서는 반드시 이 USB 디버깅이 체크되어야 합니다.



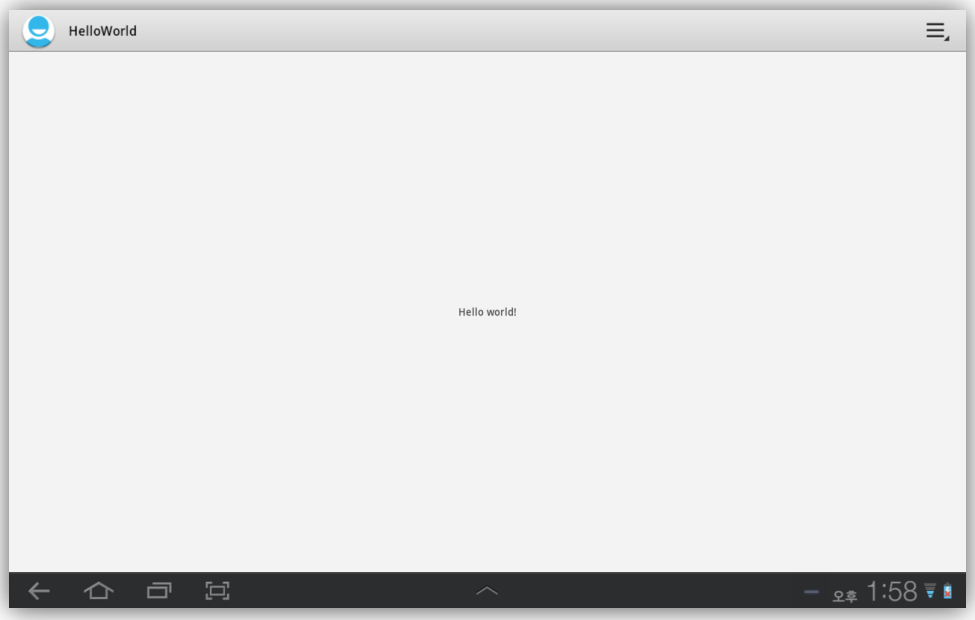
USB 디버깅 활성화

이제 F11 키를 눌러 실행해 봅니다. 프로젝트를 처음 실행할때는 다음과 같은 창이 뜨는 경우가 있습니다. 이 대화상자는 프로젝트를 실행할때 어떤 방식으로 디버깅할지를 이클립스에서 판단할 수 없을 경우 개발자에게 직접 묻는 것입니다.



HelloWorld 프로젝트 처음 실행할 때 디버그 방법 지정

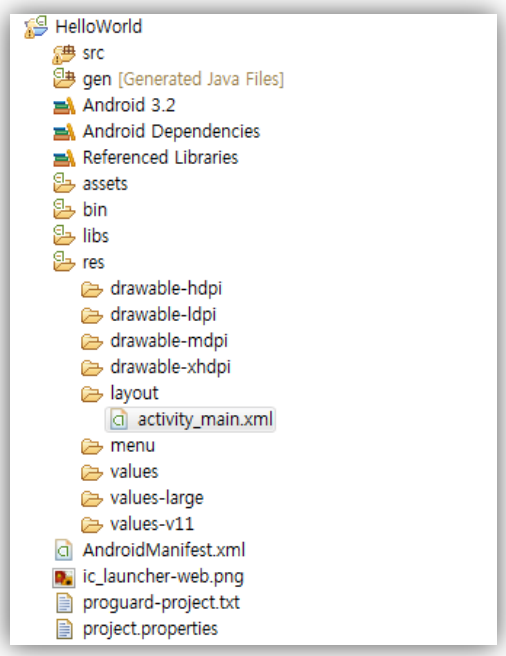
디버그 방법으로 Android Application을 선택하고 OK 버튼을 클릭합니다. 그러면 디바이스에 애플리케이션이 설치되고 다음과 같이 실행되는 것을 확인할 수 있습니다.



HelloWorld 프로젝트 실행화면

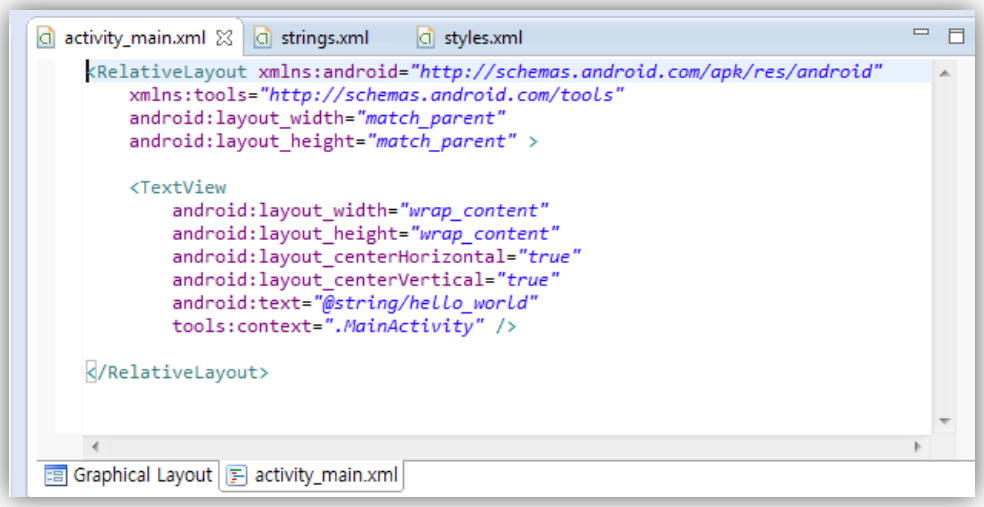
이제 블랙포인트에서 제공하는 컴포넌트인 맵뷰(XrMap 클래스)를 화면에 배치하고 간단한 SHP 파일을 레이어로 추가하여 화면에 표시해 보도록 하겠습니다.

Package Explorer에서 HelloWorld 프로젝트 폴더 안을 살펴보면 res/layout 폴더에 activity_main.xml이 존재하는 것을 알 수 있습니다. 이 파일을 더블 클릭하여 내용을 표시합니다.



HelloWorld 프로젝트의 레이아웃 파일인 activity_main.xml

변경된 화면의 중간 부분에 Graphical Layout 탭과 activity_main.xml 탭이 보이며 이 중에서 activity_main.xml 탭을 선택합니다.



activity_main.xml 탭을 선택해 activity_main.xml 파일을 직접 편집

화면에 표시된 activity_main.xml 내용을 아래처럼 변경합니다.

```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout xmlns:android="http://schemas.android.com/apk/res/android"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:orientation="vertical" >

          <geoservice.blackpoint.XrMap
              android:id="@+id/map"
              android:layout_width="fill_parent"
              android:layout_height="fill_parent" />

</LinearLayout>
```

activity_main.xml

XrMap이라는 지도 뷰를 화면에 모두 채워 표시되도록 했습니다.

이제 Permission을 추가해야 합니다. Package Explorer에서 AndroidManifest.xml 파일을 더블클릭해 열면 나타나는 화면에서 5개의 탭 중 가장 마지막 탭인 AndroidManifest.xml 탭을 선택합니다. AndroidManifest.xml 탭을 선택해서 표시된 AndroidManifest.xml 파일의 내용 중에서 <uses-sdk ..> 바로 아래 줄에 6개의 <uses-permission>을 추가합니다. 철자에 주의하시기 바랍니다. Permission 정보는 철자가 올바르지 않아도 에러로 처리되지 않으므로 무엇이 문제인지 찾기가 어렵습니다.

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

<uses-permission android:name="android.permission.GET_ACCOUNTS" />
<uses-permission android:name="android.permission.GET_TASKS" />

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

activity_main.xml

위의 각 Permission에 대한 내용은 다음과 같습니다.

1. **ACCESS_FINE_LOCATION** : GPS와 같은 장치를 이용해 정밀한 위치 확인을 허용합니다.
2. **GET_ACCOUNTS** : 모바일 디바이스의 사용자 계정 확인을 허용합니다.
3. **GET_TASKS** : 현재 화면에 실행되고 있는 TASK 확인을 허용합니다.
4. **INTERNET** : WiFi나 3G나 4G 등을 활용해 인터넷을 사용할 수 있도록 허용합니다.
5. **WRITE_EXTERNAL_STORAGE** : SD 저장 메모리에 데이터 쓰기를 허용합니다.

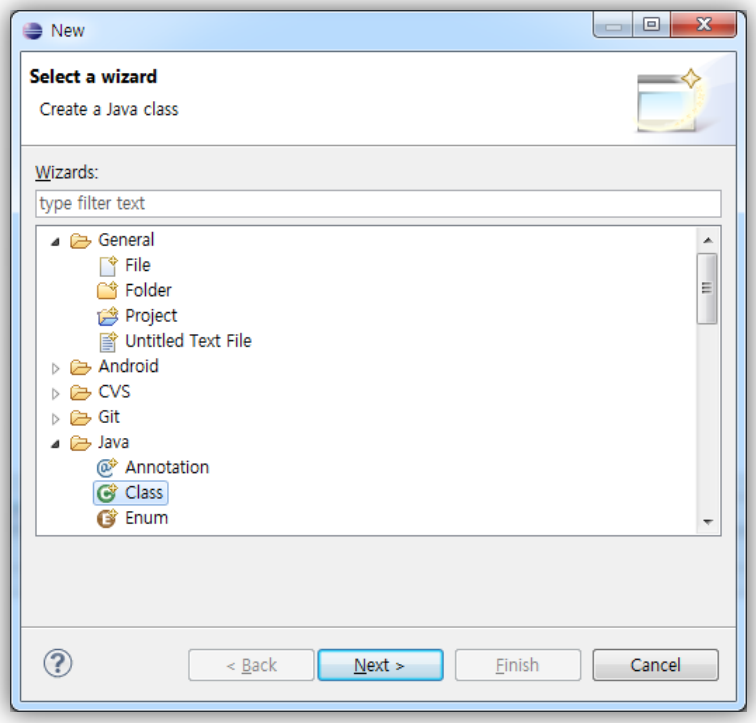
이제 수치지도 데이터인 SHP 파일을 통해 지도 레이어를 하나 추가해 보도록 하겠습니다.

레이어를 추가하는 것은 별도의 스레드(Thread)를 이용합니다. 공간 데이터는 대용량

입니다. 항공영상은 기가 바이트(Gigabyte) 단위의 대용량이며 수치지도는 수백 메가 바이트(Megabyte) 정도 됩니다. 이러한 대용량 공간 데이터를 앱이 처음 실행될때 레이어로 추가하게 되면 앱이 사용자의 응답에 반응하지 않는 상태로 수초정도 멈춰있게 됩니다.

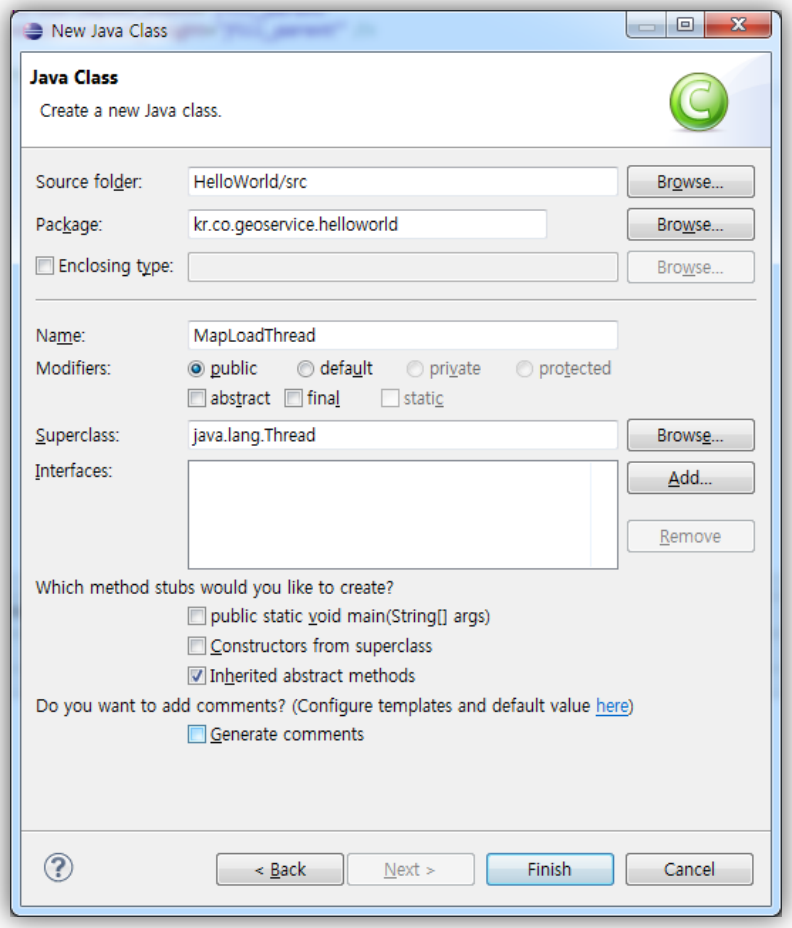
주 스레드에서 수초간 앱이 멈춰 있게 되면 ANR(Application Not Responding) 대화상자가 표시됩니다. 이런 문제를 해결하기 위해 별도의 스레드를 사용합니다. ANR 대화상자 표시를 막는 것 이외에 별도의 스레드를 사용하는 이유는 사용자에게 좋은 경험을 제공하기 위함입니다. 예를들어서 지도 데이터를 추가하고 있는 동안에 작업 진행 정도를 표시하는 대화상자를 표시해 주는 것은 사용자에게 좋은 경험입니다.

[File]-[New]-[Other] 메뉴를 실행하면 다음과 같은 대화상자가 표시됩니다.



새로운 클래스 추가

이 대화상자에서 Class 항목을 선택하고 Next 버튼을 클릭하면 다음과 같은 대화상자가 표시됩니다.



HelloWorld 프로젝트에 MapLoadThread 클래스 추가

위의 대화상자에서처럼 패키지(Package)를 kr.co.geoservice.helloworld로 지정하고 클래스의 이름을 지정하기 위해 Name에 MapLoadThread라고 입력하고 Superclass는 Thread라고 입력합니다. 그리고 Finish 버튼을 클릭합니다.

MapLoadThread 클래스를 다음처럼 입력합니다. 코드에 대한 자세한 설명은 여기서는 생략하고 다음 장에서 설명 하겠습니다.

```java
package kr.co.geoservice.helloworld;

import geoservice.blackpoint.MouseMode;
import geoservice.blackpoint.XrMap;
import geoservice.blackpoint.base.CoordMapper;
import geoservice.blackpoint.base.MBR;
import geoservice.blackpoint.managers.LayerManager;
import geoservice.blackpoint.view.layers.ShapeLayer;
import android.os.Environment;
import android.os.Handler;
import android.os.Message;


public class MapLoadThread extends Thread {
    private XrMap map = null;

    public MapLoadThread(XrMap map) {
        this.map = map;
    }

    public void run() {
        map.setMouseMode(MouseMode.NoneMode);

        LayerManager layerMan = map.getLayerManager();

        String ess = Environment.getExternalStorageState();
        String sdCardPath = null;
        if(ess.equals(Environment.MEDIA_MOUNTED)) {
            sdCardPath =
                Environment.getExternalStorageDirectory().getAbsolutePath();

            String rootDir = sdCardPath + "/mapdata/samples";

            ShapeLayer lyr = new ShapeLayer("도로", rootDir + "/roads.shp");
            layerMan.addLayer(lyr);
        }

        map.setMouseMode(MouseMode.MapViewMode);
        handler.sendEmptyMessage(0);
    }

    private Handler handler = new Handler() {
        public void handleMessage(Message msg) {
            CoordMapper coordMapper = map.getCoordMapper();
            LayerManager layerMan = map.getLayerManager();
            MBR mbr = layerMan.getMBR();
            coordMapper.zoomByMBR(mbr);
            map.update();
        }
    };
}
```

MapLoadThread.java

이제 MainActivity 클래스를 수정합니다. Package Explorer에서 MainActivity.java 파일을 더블 클릭하면 아래와 같이 기본 코드가 이미 입력되어 있는 것을 볼 수 있습니다.

```java
package kr.co.geoservice.helloworld;

import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;

public class MainActivity extends Activity {

    @Override
    public void onCreate(Bundle savedInstanceState) {
        super.onCreate(savedInstanceState);
        setContentView(R.layout.activity_main);
    }

    @Override
    public boolean onCreateOptionsMenu(Menu menu) {
        getMenuInflater().inflate(R.menu.activity_main, menu);
        return true;
    }
}
```

MainActivity.java

이 클래스가 OnMapReadyEventListener 인터페이스를 구현하도록 합니다. 이 인터페이스는 블랙포인트의 맵뷰가 레이아웃에 완전하게 배치되어 지도 레이어를 추가할 수 있는 준비가 되었을때를 위한 이벤트를 제공합니다.

```java
package geoservice.blackpoint.helloworld;

import geoservice.blackpoint.events.OnMapReadyEventListener;
import android.app.Activity;
import android.os.Bundle;
import android.view.Menu;

public class MainActivity extends Activity implements OnMapReadyEventListener
```

MainActivity.java

그리고 private 변수를 2개 추가합니다.

```java
private XrMap map = null;
private MapLoadThread dataLoadThread = null;
```

MainActivity.java

XrMap 타입인 map은 레이아웃(layout)에 배치된 블랙포인트의 지도뷰이고 MapLoadThread 타입인 dataLoadThread는 앞서 추가한 지도 데이터를 로딩하는 스레드 클래스 변수입니다.

이제 onCreate 함수를 아래의 코드처럼 변경합니다.

```java
@Override
public void onCreate(Bundle savedInstanceState) {
    super.onCreate(savedInstanceState);
    setContentView(R.layout.activity_main);

    map = (XrMap)findViewById(R.id.map);

    DisplayMetrics outMetrics = new DisplayMetrics();
    getWindowManager().getDefaultDisplay().getMetrics(outMetrics);
    int DPI = outMetrics.densityDpi;
    map.setDPI(DPI);

    map.setOnMapReadyListener(this);
}
```

MainActivity.java

그리고 OnMapReadyEventListener에서 구현해야할 인터페이스 함수인 onMapReady를 다음처럼 구현합니다. 이 onMapReady 함수에서 레이어를 추가하기 위한 MapLoadThread를 생성하고 스레드 실행을 시작(start)합니다.

```java
@Override
public void onMapReady(MapEvent event) {
    dataLoadThread = new MapLoadThread(map);
    dataLoadThread.start();
}
```

MainActivity.java

그리고 애플리케이션이 종료될 때 메모리를 정리하기 위해 onDestroy 함수를 추가합니다.

```java
@Override
public void onDestroy() {
    super.onDestroy();

    map.release();

    System.gc();
}
```

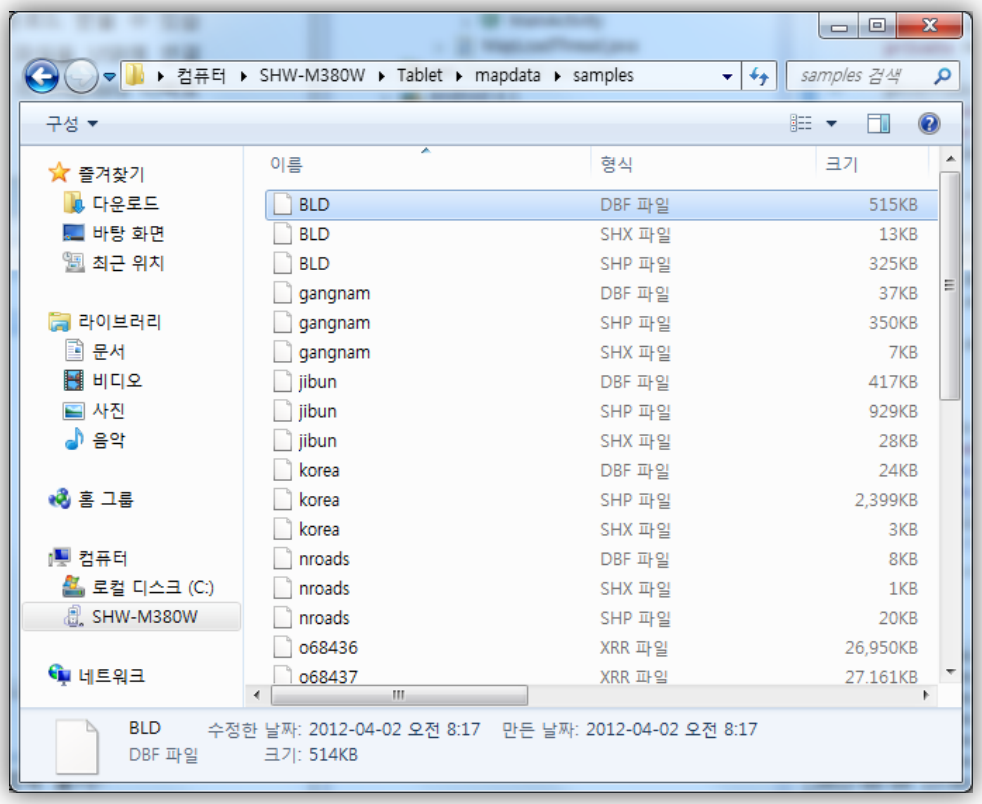
MainActivity.java

코드 추가를 통해 새로운 클래스가 많이 사용되었습니다. 이러한 클래스를 자바 컴파일러가 인식할 수 있도록 import 시켜줘야 합니다. 사용된 클래스를 자동으로 import 시켜 주는 기능을 이클립스에서 지원하는데 단축키인 소스코드 입력 창에서 Ctrl+Shift+O를 누르면 코스코드 상단에 아래의 코드처럼 import 문이 추가되어 변경됩니다.

```
import geoservice.blackpoint.XrMap;
import geoservice.blackpoint.events.MapEvent;
import geoservice.blackpoint.events.OnMapReadyEventListener;
import android.app.Activity;
import android.os.Bundle;
import android.util.DisplayMetrics;
import android.view.Menu;
```

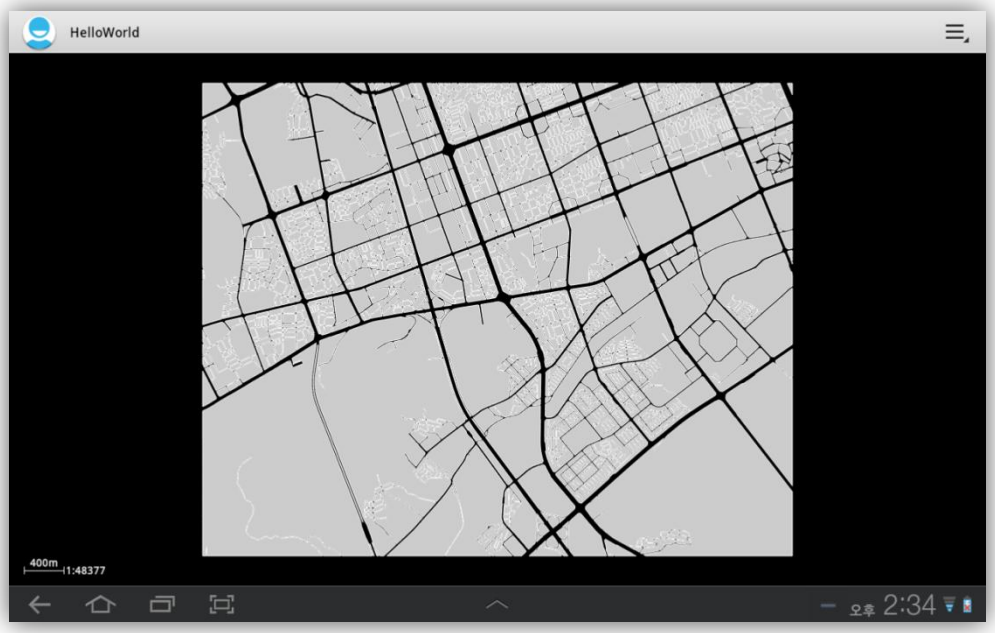
MainActivity.java

이제 소스 코드는 모두 완성되었습니다. 실행하기 위해 마지막으로 필요한 것은 공간 지도 데이터입니다. 이 책의 예제 코드에서 사용하는 모든 데이터는 지오서비스의 자료실(http://www.geoservice.co.kr/board/list.php?cate=02)에서 다운로드 받을 수 있습니다. 자료실에서 다운로드 받은 blackpoint_sample_map_data.zip 파일을 USB에 연결되 모바일 디바이스의 최상단 경로에 mapdata 디렉토리를 만들고 이 mapdata 디렉토리안에 samples 디렉토리를 이름으로 해서 압축을 풉니다.



디바이스에 mapdata/samples 폴더에 공간 데이터 압축 풀기

이제 모든 준비가 되었으므로 F11 키를 눌러 애플리케이션을 실행합니다.



HelloWorld 프로젝트 실행 화면

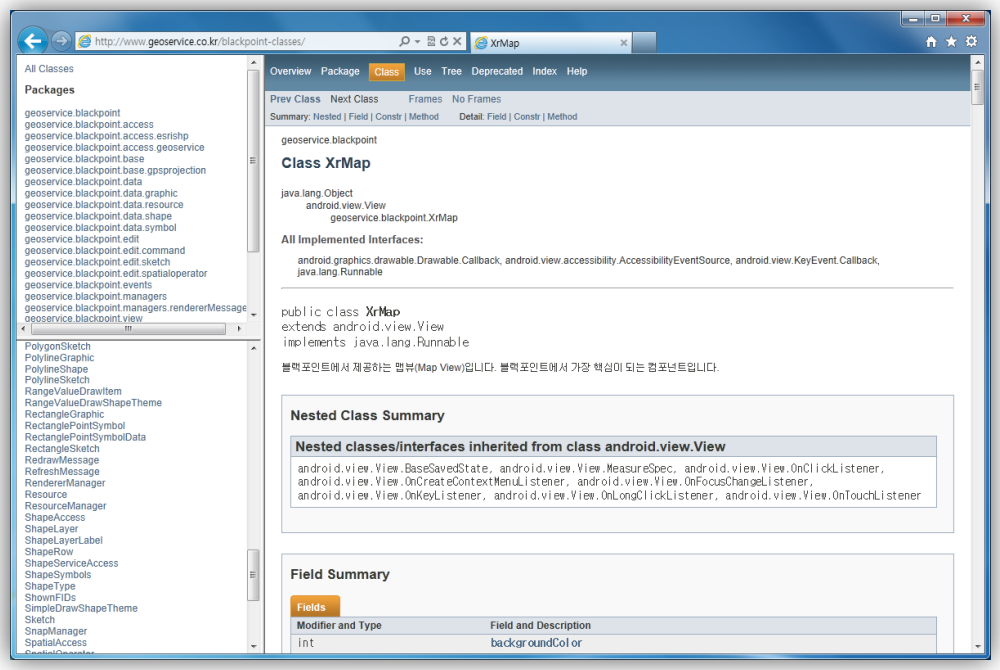
실행된 상태에서 화면 터치를 통해 지도 이동 및 확대/축소가 가능하며 손가락을 1초 정도 누른 상태에서 손가락을 좌우로 스크롤하면 지도가 회전됩니다.

2 블랙포인트의 클래스 설명

블랙포인트에서 제공하는 클래스에 대한 상세한 설명은 그 내용이 매우 많습니다. 이러한 내용을 이 책에서 수록하기 보다는 인터넷을 통해 쉽게 찾아볼 수 있도록 다음 URL을 통해 볼 수 있습니다.

http://www.geoservice.co.kr/blackpoint-classes

위의 URL을 인터넷 브라우저를 통해 살펴보면 다음과 같습니다.



블랙포인트에서 제공하는 클래스들의 명세서