

# 05장. 자원관리

- 01. 자원이란
- 02. 자원 다루기
- 03. 시스템 자원 참조하기
- 04. 다중 응용 프로그램 구성 관리

컴퓨터공학과 변영철 교수

# 자원이란

- 자원 - 텍스트 문자열, 이미지, 아이콘, 오디오, 동영상 등 응용 프로그램이 사용하는 자산(asset)
- 잘 만들어진 응용 프로그램은 자원을 소스 코드에 박아 넣는 하드 코딩을 사용하지 않고 코드와 자원을 분리함
- 소스 코드 바깥(res 폴더)에 저장하고 프로그래밍적으로 적절한 식별자와 메서드 호출을 통하여 접근함
- 그림으로써 읽기 쉽고 유지보수가 쉬워짐

폴더	용도
/res/drawable/	그래픽 자원들
/res/layout/	사용자 인터페이스 자원들
/res/values/	문자열, 색상 값 같은 단순 자료들

# aapt와 R.java

- aapt(Android Asset Packaging Tool)
  - 이클립스에서 개발자가 res 폴더에 자원을 추가하면 plug-in이 aapt(SDK의 tools 폴더)를 이용해서 **자동으로 자원 컴파일** (/res/raw 폴더에 있는 자원 파일은 컴파일하지 않음)
  - 기본적으로, aapt 도구는 응용프로그램을 에뮬레이터나 폰에 설치하기 **위한 설치 패키지를 생성**
  - 이 도구는 res 폴더의 자원들을 컴파일하여 **자원 참조 테이블인 R.java 클래스**를 gen 폴더에 생성
  - 소스 코드에서는 R.java를 참조하여 자원을 액세스
  - 이클립스 및 plug-in을 이용하지 않을 경우 수작업으로 자원을 컴파일하고 설치 패키지를 생성해야 함

# R.java를 통한 간접 접근 이유

- 가령, 스마트폰 LCD 해상도에 따라서 응용이 사용하는 icon이 다름 (현재는 세가지 해상도가 존재)
  - drawable\_low.icon, drawable\_medium.icon, drawable\_high.icon
- 프로그램에서 아이콘을 직접 접근해야 할 경우 세 자원 중 하나를 지정해야 함
- 해상도에 관계없이 한 가지 방법(drawable.icon)으로 접근하고, 아이콘이 3개 존재하여도 LCD 해상도에 따라 안드로이드 Resource Manager가 자동으로 적절한 것을 선택하는 것이 바람직
- 또한 다양한 국가에 따라 사용하는 언어 등 리소스가 달라질 수 있으므로 이 자원 또한 시스템 설정에 따라 Resource Manager가 직접 선택하도록 할 필요가 있음(Internationalization)
- 이렇듯 다양한 상황에 관계없이 동일한 방법으로(코드 변경없이) 접근하려면 리소스를 직접 접근해서는 불가능하고 대신 R.java에 선언되어 있는 레퍼런스로 간접적으로 접근하여야 함

## 01. 자원이란(4)

# 자원 형식

- 모든 자원 파일 이름은 영문 소문자와 숫자, 밑줄로만 이루어져야 함

자원 형식	필수 디렉터리	파일 이름	핵심 XML 요소 이름 (적용되는 경우에만)
문자열	/res/values/	<b>strings.xml</b> (권장됨)	<string>
문자열 배열	/res/values/	<b>arrays.xml</b> (권장됨)	<string-array>
색상값	/res/values/	<b>colors.xml</b> (권장됨)	<color>
크기(dimension)	/res/values/	<b>dimens.xml</b> (권장됨)	<dimen>
단순 표시물 (drawable)	/res/values/	<b>drawables.xml</b> (권장됨)	<drawable>
비트맵 그래픽	/res/drawable/	예: <b>img.png</b> , <b>img.jpg</b> , <b>img.9.png</b> , <b>img.gif</b> , <b>red_oval.xml</b>	지원되는 그래픽 파일 또는 도형 등의 표시 물을 정의하는 XML 파일들.
애니메이션 시퀀스 (트위닝)	/res/anim/	예: <b>fancy_anim1.xml</b> , <b>fancy_</b> <b>anim2.xml</b>	<set>, <alpha>, <scale>,<translate>, <rotate>
메뉴 파일	/res/menu/	예: <b>my_menu1.xml</b> , <b>more_options.</b> <b>xml</b>	<menu>
XML 파일	/res/xml/	예: <b>some.xml</b> , <b>more.xml</b>	개발자가 정의.
원본(raw) 파일	/res/raw/	예: <b>some_audio.mp3</b> , <b>some_video.</b> <b>mp4</b> , <b>some_text.txt</b>	
레이아웃 파일	/res/layout/	예: <b>start_screen.xml</b> , <b>main_</b> <b>screen.xml</b> , <b>help_screen.xml</b>	다양함. 반드시 레이 아웃 요소이어야 함.
스타일 및 테마	/res/values/	<b>styles.xml</b> , <b>themes.xml</b> (권장됨)	<style>

# 저장 방식 및 접근 방법

- 문자열, 색상 등 단순자원
  - /res/values 폴더에 XML 파일로 저장
  - 하나의 XML 파일 보다는 여러 XML 파일에 나누어 저장하는 것이 바람직 (strings.xml, colors.xml, bright\_color.xml 등)
- 그래픽, 애니메이션, 메뉴, 파일
  - 단순 자원들 외에 애니메이션, 그래픽, 임의의 XML 파일, 원본 파일들도 자원으로 저장할 수 있음
  - 해당 형식마다 개별적인 디렉터리에 저장
  - 가령 flag.png 그래픽 파일은 /res/drawable 디렉터리에 저장되며 이로부터 R.drawable.flag라는 이름으로 참조됨
- 접근 방법

```
String myString = getResources().getString(R.string.hello);
```

## 02. 자원 접근 예제(1)

# 자원 접근 예제

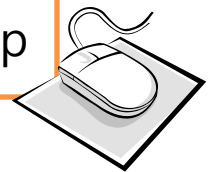
- 자원 정의 (/res/values/strings.xml)

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <color name="prettyTextColor">#ff0000</color>
  <dimen name="textPointSize">14pt</dimen>
  <drawable name="redDrawable">#F00</drawable>
  <string name="app_name">Resource Roundup</string>
  <string name="hello">Hello World, ResourceRoundup</string>
</resources>
```

- 참조 테이블 자동 생성 (/gen/R.java)

```
public final class R {
    public static final class color {
        public static final int prettyTextColor=0x7f050000; }
    public static final class dimen {
        public static final int textPointSize=0x7f060000; }
    public static final class drawable {
        public static final int icon=0x7f020000;
        public static final int redDrawable=0x7f020001; }
    public static final class string {
        public static final int app_name=0x7f040000;
        public static final int hello=0x7f040001;
    }
}
```

ResourceRoundup



## 02. 자원 접근 예제(2)

# 자원 접근 예제

- 자원 접근 코드

```
int myColor = getResources().getColor(R.color.prettyTextColor);  
float myDimen = getResources().getDimension(R.dimen.textPointSize);  
ColorDrawable myDraw = (ColorDrawable)getResources().getDrawable(R.drawable.redDrawable);  
String myString = getResources().getString(R.string.hello);
```



## 02. 자원 접근 예제(3)

# 자원 접근 예제

- 문자열 배열 정의 (/res/values/strings.xml)

```
<resources>
  <string-array name="flavors">
    <item>바닐라</item>
    <item>초코</item>
    <item>딸기</item>
  </string-array>
</resources>
```

- 접근 코드

```
String[] aFlavors = getResources().getStringArray(R.array.flavors);
```

### 03. 자원 다루기(1)

## 스타일/서식 문자열

- 스타일 문자열

- 문자열에 볼드, 이탤릭, 밑줄 서식 지원

```
<string name="txt"> <b>볼드</b>, <i>이탤릭</i>, <u>밑줄</u> </string>
```

- 스타일 문자열 접근 코드

- getString() 메서드를 사용해서 문자열 자체만 가져오는
- 문자 스타일을 유지하려면 getText() 메서드를 사용해야 함

```
CharSequence myBoldStr = getResources().getText(R.string.boldhello);
```

- styled & 포맷 문자열

```
<string name="styledformatted">Your score is %1$d out of %2$d!  
You &lt;i>%3$s</i>.</string>
```

### 03. 자원 다루기(2)

## 스타일/서식 문자열

- styled & 포맷 문자열 접근 코드

```
String styledformatted = getResources().getString(R.string.styledformatted);  
Log.d(DEBUG_TAG, "Styled & formatted: " + styledformatted);
```

```
String result = String.format(styledformatted, 5, 5, escapedWin);  
Log.d(DEBUG_TAG, result);
```

```
CharSequence html = Html.fromHtml(result);  
Log.d(DEBUG_TAG, "Styled results: " + html);
```

```
original: Your score is %1$d out of %2$d! You  &lt;i>%3$s&lt;/i>  
result:   Your score is 5 our of 5! You &lt;i>Won&lt;/i>  
html:    Your score is 5 out of 5! You <i>Won</i>
```

## 문자열 배열

- /res/values/arrays.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string-array name="flavors">
    <item>바닐라</item>
    <item>초코</item>
    <item>딸기</item>
    <item>모카</item>
    <item>청국장</item>
  </string-array>
  <string-array name="soups">
    <item>야채수프</item>
    <item>뉴잉글랜드 클램차우더</item>
    <item>유기농 닭국수</item>
  </string-array>
</resources>
```

```
String[] aFlavors = getResources().getStringArray(R.array.flavors);
```

## 02. 자원 다루기(6)

# 색상

- /res/values/colors.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
    <color name="prettyTextColor">#ff0000</color>
    <color name="prettyBackgroundColor">#00ff00</color>
</resources>
```

```
int myResourceColor = getResources().getColor(R.color.prettyTextColor);
```

## 02. 자원 다루기(7)

# 크기

- /res/values/dimens.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <dimen name="FourteenPt">14pt</dimen>
  <dimen name="OneInch">1in</dimen>
  <dimen name="TenMillimeters">10mm</dimen>
  <dimen name="TenPixels">10px</dimen>
</resources>
```

```
float myDimension = getResources().getDimension(R.dimen.OneInch);
```

# 단순 Drawable

- 도형을 정의하는 xml : 단순 Drawable
  - /res/drawable/red\_oval.xml
  - 빨간색 원을 정의한 Drawable

```
<?xml version="1.0" encoding="utf-8"?>
<!-- This is a simple XML definition for a Drawable of subclass ShapeDrawable -> A red oval. -->
<shape xmlns:android="http://schemas.android.com/apk/res/android" android:shape="oval">
    <solid android:color="#f00"/>
</shape>
```

- Drawable을 표시하도록 설정한 위젯

```
<ImageView
    android:id="@+id/ImageView02"
    android:layout_gravity="center"
    android:layout_width="200px"
    android:layout_height="200px"
    android:src="@drawable/red_oval"> </ImageView>
```

# 단순 Drawable

- 애니메이션 xml
  - /res/anim/pulse.xml

```
<?xml version="1.0" encoding="utf-8" ?>
<set xmlns:android="http://schemas.android.com/apk/res/android" android:shareInterpolator="false">
  <set>
    <rotate android:fromDegrees="0" android:toDegrees="-1440" android:pivotX="50%"
      android:pivotY="50%"
      android:duration="10000" />
  </set>
</set>
```

- 단순 Drawable을 표시하는 위젯에 애니메이션 구동

```
ImageView redoal_view = (ImageView)findViewById(R.id.ImageView02);
Animation pulse_anim = AnimationUtils.loadAnimation(this, R.anim.pulse);
redoal_view.startAnimation(pulse_anim);
```



# 이미지

- /res/drawable 폴더의 flag.png(일반 이미지)

```
import android.widget.ImageView;
...
ImageView flagImageView = (ImageView)findViewById(R.id.ImageView01);
flagImageView.setImageResource(R.drawable.flag);
```

- /res/drawable 폴더의 pyramid.png(9-패치 이미지)

```
import android.graphics.drawable.NinePatchDrawable;
...
NinePatchDrawable nine_p=
    (NinePatchDrawable)getResources().getDrawable(R.drawable.pyramid);
int iStretchyHeightInPixels = nine_p.getIntrinsicHeight(); //원래 높이
int iStretchyWidthInPixels = nine_p.getIntrinsicWidth(); //원래 너비
```

비레되지 않음	수평으로만 비레	비레되지 않음
수직으로만 비레됨	수평, 수직으로 비레됨	수직으로만 비레됨
비레되지 않음	수평으로만 비레	비레되지 않음

# 애니메이션

- /res/anim/ 폴더의 spin.xml (애니메이션 파일)
  - 대상 그래픽(위젯)을 10초 동안 반시계 방향으로 네 번 회전시키는 애니메이션 xml 파일

```
<?xml version="1.0" encoding="utf-8" ?>
<set xmlns:android="http://schemas.android.com/apk/res/android"
    android:shareInterpolator="false">

    <set>
        <rotate android:fromDegrees="0"
            android:toDegrees="-1440"
            android:pivotX="50%"
            android:pivotY="50%"
            android:duration="10000"

        />
    </set>
</set>
```

# 애니메이션

- 애니메이션 수행 코드

```
import android.view.animation.Animation;
import android.view.animation.AnimationUtils;
import android.widget.ImageView;
...
ImageView flagImageView = (ImageView)findViewById(R.id.ImageView01);
flagImageView.setImageResource(R.drawable.flag);
...
Animation an = AnimationUtils.loadAnimation(this, R.anim.spin);
flagImageView.startAnimation(an);
```

## 02. 자원 다루기(9)

# 메뉴

- /res/menu 폴더에 speed.xml

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/start"
        android:title="Start!"
        android:orderInCategory="1"> </item>
  <item android:id="@+id/stop"
        android:title="Stop!"
        android:orderInCategory="4"> </item>
  <item android:id="@+id/accel"
        android:title="Vroom! Accelerate!"
        android:orderInCategory="2"> </item>
  <item android:id="@+id/decel"
        android:title="@string/hello"
        android:orderInCategory="3"> </item>
</menu>
```

MENU 버튼을  
누를 경우 실행됨

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

# XML 파일

- /res/xml 폴더에 my\_pets.xml

```
<?xml version="1.0" encoding="utf-8"?>
<pets>
  <pet name="Bit" type="Bunny" />
  <pet name="Nibble" type="Bunny" />
  <pet name="Stack" type="Bunny" />
  <pet name="Queue" type="Bunny" />
  <pet name="Heap" type="Bunny" />
  <pet name="Null" type="Bunny" />
  <pet name="Nigiri" type="Fish" />
  <pet name="Sashimi II" type="Fish" />
  <pet name="Kiwi" type="Lovebird" />
</pets>
```

# XML 파일

- 접근 코드

```
XmlResourceParser myPets = getResources().getXml(R.xml.my_pets);
```

```
import org.xmlpull.v1.XmlPullParserException;
import android.content.res.XmlResourceParser;
...
int eventType = -1;
while (eventType != XmlResourceParser.END_DOCUMENT) {
    if(eventType == XmlResourceParser.START_DOCUMENT) {
        Log.d(DEBUG_TAG, "Document Start");
    } else if(eventType == XmlResourceParser.START_TAG) {
        String strName = myPets.getName();
        if(strName.equals("pet")) {
            Log.d(DEBUG_TAG, "Found a PET");
            Log.d(DEBUG_TAG,
                "Name: "+myPets.
                getAttributeValue(null, "name"));
            Log.d(DEBUG_TAG,
                "Species: "+myPets.
                getAttributeValue(null, "type"));
        }
    }
    eventType = myPets.next();
}
```

## 02. 자원 다루기(11)

# Raw 파일

- /res/raw 폴더에 limerick.txt

A wonderful bird is the pelican.  
His bill will hold more than his belican.  
He can take in his beak.  
Enough food for a week.  
But I'm darned if I see how the helican!  
- Dixon Lanier Merritt, 1910

- 접근 코드

```
// SOME RAW FILE HANDLING
InputStream istream = getResources().openRawResource(R.raw.limerick);
try {
    String strFile = inputStreamToString(istream);
    Log.d(DEBUG_TAG, "Here's the limerick from the Input Stream:\n" + strFile);
} catch (Exception e) {
    Log.e(DEBUG_TAG, "InputStreamToString broke.", e);
}
```

# Raw 파일

```
public String inputStreamToString(InputStream is) throws IOException
{
    StringBuffer sBuffer = new StringBuffer();
    DataInputStream dataIO = new DataInputStream(is);
    String strLine = null;

    while((strLine=dataIO.readLine()) != null){
        sBuffer.append(strLine+"\n");
    }

    dataIO.close();
    is.close();

    return sBuffer.toString();
}
```



# 자원 참조

/res/values/strings.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string name="app_name">Soup List</string>
  <string name="chicken_soup">
    Organic Chicken Noodle</string>
  <string name="minestrone_soup">
    Veggie Minestrone</string>
  <string name="chowder_soup">
    New England Lobster Chowder</string>
</resources>
```

/res/values/arrays.xml

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <string-array name="soups">
    <item>@string/minestrone_soup</item>
    <item>@string/chowder_soup</item>
    <item>@string/chicken_soup</item>
  </string-array>
</resources>
```

# 레이아웃

- /res/layout 폴더의 main.xml
  - /res/values/string.xml, colors.xml, dims.xml

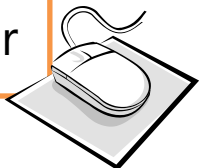
```
<?xml version="1.0" encoding="utf-8"?>
<LinearLayout
    xmlns:android="http://schemas.android.com/apk/res/android"
    android:orientation="vertical"
    android:layout_width="fill_parent"
    android:layout_height="fill_parent"
    android:background="@color/background_color">

    <TextView android:id="@+id/TextView01"
        android:layout_width="fill_parent"
        android:layout_height="fill_parent"
        android:text="@string/test_string"
        android:textColor="@color/text_color"
        android:gravity="center"
        android:textSize="@dimen/ThirtySixPt">
    </TextView>
</LinearLayout>
```



ResourceViewer

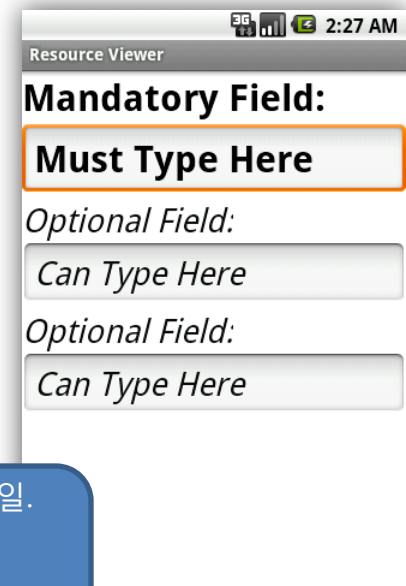
/res/layout/main.xml



# 스타일

- 위젯 하나에 적용됨
- 텍스트 색상, 크기, 속성을 스타일로 정의한 모습

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
  <style name="mandatory_text_field_style">
    <item name="android:textColor">#000000</item>
    <item name="android:textSize">14pt</item>
    <item name="android:textStyle">bold</item>
  </style>
  <style name="optional_text_field_style">
    <item name="android:textColor">#0F0F0F</item>
    <item name="android:textSize">12pt</item>
    <item name="android:textStyle">italic</item>
  </style>
</resources>
```



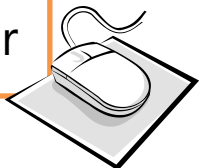
- 위젯에 위 스타일을 적용하려면

```
<TextView
  android:id="@+id/TextView01"
  style="@style/mandatory_text_field_style"
  ... />
```

테마도 스타일과 동일.  
유일한 차이는  
<Activity>에  
android:theme 특성에  
설정한다는 것이 다름

ResourceViewer

/res/layout/form.xml



## 시스템 자원

- android.R
  - fade-in, fade-out 애니메이션
    - android.R.anim.fade\_in, fade\_out
  - 표준 시스템 색상들
  - thumbnail 이미지와 아이콘들
  - drawable과 레이아웃
  - 기본 문자열들
  - 스템 스타일과 테마들

# 자원 자동 설정

- 자원의 간접참조(R.java)를 통하여 로캘 및 기기 구성에 따라 적절한 자원이 자동으로 선택됨
- 기기의 LCD 해상도에 따라 drawable-ldpi, drawable-mdpi, drawable-hdpi에 있는 아이콘을 자동 선택
- 언어에 따른 자원 폴더가 따로 존재할 경우 특정 로캘(지역, 언어)을 선택하면(홈 > MENU > Settings > Lanaguage & Keyboard > Select locale) 응용의 문자열 및 관련된 설정이 자동으로 바뀜
- 자원 자동설정을 위한 폴더 및 허용되는 값

## 02. 자원 다루기(15)

# 자원 자동 설정

디렉터리 한정사 범주	허용되는 값	비고
<b>언어</b> 반드시 첫 한정사이어야 함. 예: /res/values-en/	en, fr, es, zh, ja, ko, de, fi	ISO 639-1 두 글자 부호(열 칸의 예는 차례로 영어, 프랑스어, 스페 인어, 중국어, 일본어, 한국어, 독 일어, 핀란드어나, 다른 것들도 가능하다.)
<b>지역</b> 언어보다 앞에 나오면 안 됨. 예: /res/values-en-rUS/	rUS, rGB, rFR, rES, rMX, rCN, rJP, rKR, rDE, rFI	소문자 'r' 다음에 대문자 ISO 3166-1-alpha-2 지역 부호가 붙 은 형태. (열 칸의 예는 차례로 미 국, 영국, 프랑스, 스페인, 멕시코, 중국, 일본, 남한, 독일, 핀란드이 나, 다른 것들도 가능하다.)
<b>화면 방향</b> 지역보다 앞에 나오면 안 됨. 예: /res/drawable-port/	port, land, square	세로 모드, 가로 모드, 정사각형 화면
<b>화면 픽셀 밀도</b> 화면 방향보다 앞에 나오면 안 됨. 예: /res/drawable-port-92dpi/	92dpi, 108dpi	가능한 값들은 기기에 따라 다를 수 있음
<b>터치스크린 종류</b> 화면 픽셀 밀도보다 앞에 나오면 안 됨. 예: /res/values-port-notouch/	notouch, stylus, finger	터치스크린 없음, 스타일러스 전 용, 손가락 터치스크린
<b>키보드 모드</b> 터치스크린 종류보다 앞에 나오면 안 됨. 예: /res/drawable-keyshidden/	keysexposed, keyshidden	사용자에게 키보드 제공됨, 키보 드 제공되지 않음
<b>기본 텍스트 입력 방법</b> 키보드 모드보다 앞에 나오면 안 됨. 예: /res/values-port-notouch-12key/	nokeys, qwerty, 12key	키보드 없음(터치스크린 입력만), 완전한 QWERTY 키보드, 숫자키 패드
<b>기본 비 터치스크린 내비게이션 방법</b> 기본 텍스트 입력 방법보다 앞에 나오면 안 됨. 예: /res/values-notouch-wheel/	nonav, dpad, trackball, wheel	4키 방향 패드, 트랙볼(T-Mobile G1에 있는 것 같은), 스크롤 휠
<b>화면 크기</b> 항상 제일 마지막 한정사. 예: /res/drawables-en-port-640x480/	320x240, 640x480	가장 큰 크기를 제일 처음 지정해 야 함.

손가락 터치 스크린과  
qwerty 자판, 320x240  
해상도 LCD를 갖는 기  
기에서,  
로컬이 한국어로 설정  
되고 화면이 가로일  
경우 사용하는 값들

/res/values-ko-rKR-port-finger-qwerty-320x240

# 자원 자동 설정

## /res/values-en

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<string name="app_name">Resource Viewer</string>
<string name="hello">Hello everyone!</string>
</resources>
```

개발자는 자원 폴더를 제대로 구성하기만 하면 설정에 따라 폴더를 선택하는 것은 안드로이드 플랫폼 몫임

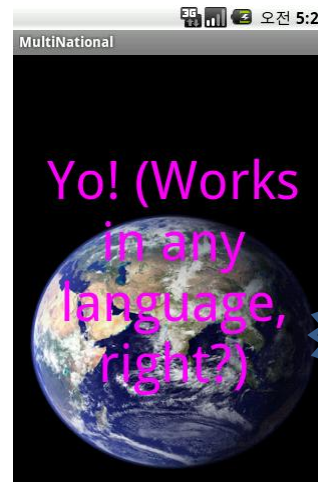
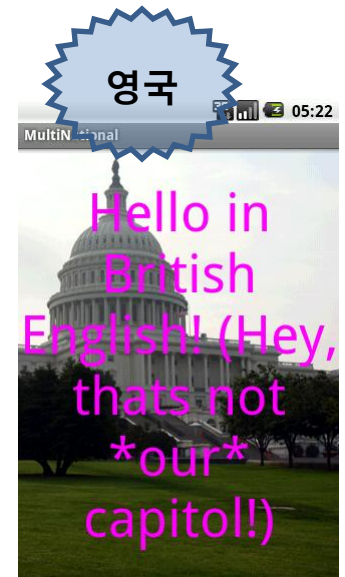
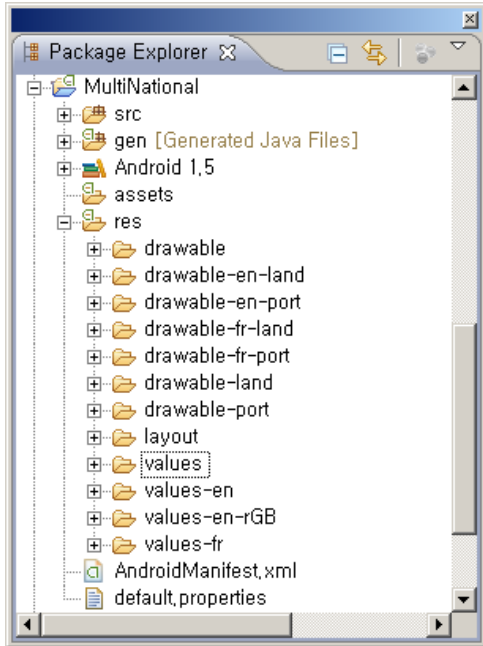
## /res/values-kr

```
<?xml version="1.0" encoding="utf-8"?>
<resources>
<string name="app_name">리소스 뷰어</string>
<string name="hello">모두들 안녕!</string>
</resources>
```

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

    <TextView
        android:layout_width="fill_parent"
        android:text="@string/hello"
        android:textSize="24pt"
        android:layout_height="fill_parent"
        android:gravity="center" />
</LinearLayout>
```

# 자원 자동 설정



한국

Multinational

