Images, layout descriptions, binary blobs and string dictionaries can be included

in your application as resource files. Various Android APIs are designed to

operate on the resource IDs instead of dealing with images, strings or binary blobs

directly.

For example, a sample Android app that contains a user interface layout (main.xml),

an internationalization string table (strings.xml) and some icons (drawable-XXX/icon.png)

would keep its resources in the "Resources" directory of the application:

Resources/

drawable-hdpi/

icon.png

drawable-ldpi/

icon.png

drawable-mdpi/

icon.png

layout/

main.xml

values/

strings.xml

In order to get the build system to recognize Android resources, set the build action to

"AndroidResource". The native Android APIs do not operate directly with filenames, but

instead operate on resource IDs. When you compile an Android application that uses resources,

the build system will package the resources for distribution and generate a class called

"Resource" that contains the tokens for each one of the resources included. For example,

for the above Resources layout, this is what the Resource class would expose:

public class Resource {

public class drawable {

public const int icon = 0x123;

}

public class layout {

public const int main = 0x456;

}

public class strings {

public const int first\_string = 0xabc;

public const int second\_string = 0xbcd;

}

}

You would then use R.drawable.icon to reference the drawable/icon.png file, or Resource.layout.main

to reference the layout/main.xml file, or Resource.strings.first\_string to reference the first

string in the dictionary file values/strings.xml.