



Multiple APK Support

Multiple APK support is a new option in Google Play (Android Market) by which multiple versions of APKs can be uploaded for a single application.

The topics covered in this chapter are as follows:

- What is and why to use Multiple APK Support
- How to use Multiple APK support

What is Multiple APK support

Multiple APK support is a Google Play (Android Market) feature which allows developers to publish multiple APKs of a single application for different configurations. Normally, one APK file is produced for an application. However, there may be some cases where one APK file is not enough. Using this feature, your application is listed in Google Play as a single application, but when a user downloads your application, the APK file which is compatible with the user's device will be installed. Users will not be aware of multiple APKs. Users will see a single description of the application on Google Play, and they will rate and comment on a single application in Google Play, not the APK that they installed.

Configurations for multiple APK

Each APK that is developed should target a different device configuration. This could be achieved by defining filters in our `AndroidManifest.xml` file. The filters that can be used for producing a multiple APK are as follows:

- **Open GL texture compression formats:** This format can be specified in the `AndroidManifest.xml` file by the `<supports-gl-texture>` element. For instance, one APK could support the Ericsson texture compression and another APK could support the generic paletted texture compression. Then these formats should be specified in each APK's `AndroidManifest.xml` file. One APK should contain the `<supports-gl-texture android:name="GL_OES_compressed_ETC1_RGB8_texture" />` element in its `AndroidManifest.xml` file and another APK should contain the `<supports-gl-texture android:name="GL_OES_compressed_paletted_texture" />`. Using this, the device will download the appropriate APK according to its configuration. If the device supports Ericsson texture compression, it will download the first APK.
- **Screen Size:** This filter could be specified in the `AndroidManifest.xml` file by the `<supports-screens>` or `<compatible-screens>` elements. These elements should not be used together. We can use these filters to define the screen size that our APK supports. For instance, one of our APKs might be compatible with phones (small and normal screens) and the other APK might be compatible with tablets (large and xlarge screens). The first APK should contain the following element in the `AndroidManifest.xml` file:

```
<supports-screens
  android:smallScreens="true"
  android:normalScreens="true" />
```

The second APK should contain the following element in `AndroidManifest.xml`:

```
<supports-screens
  android:largeScreens="true"
  android:xLargeScreens="true" />
```

According to its screen size, a device will install the first or the second APK.

- **Platform Version:** This filter could be specified in `AndroidManifest.xml` by the `<uses-sdk>` element. This filter is used to specify the API Level that our APK supports. For instance one of our APKs could be compatible with API Level 4 to 10, and another APK could be compatible with API Level 11 or later. Then the first APK should contain the following in the `AndroidManifest.xml` file:

```
<uses-sdk android:minSdkVersion="4" android:maxSdkVersion="10" />
```

The second APK should contain the following in the `AndroidManifest.xml` file:

```
<uses-sdk android:minSdkVersion="11" />
```

Using these filters, a device will download the compatible APK according to its Android version.

Creating Multiple APKs

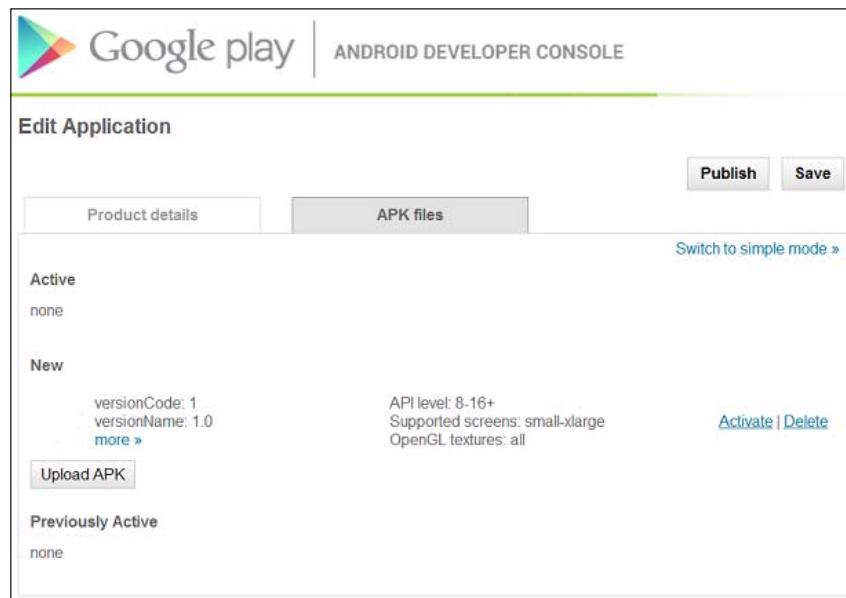
There are some rules that we need to follow for creating multiple APKs:

- Each APK should have the same package name.
- Each APK should be signed with the same release key.
- Each APK should have a different version code.
- An APK's configuration should not overlap completely with another APK's configuration. However, overlapping slightly is allowed. For instance, two APKs could have the same screen configurations, but they could be different in other configurations. Then the device will download the APK with the higher version code.
- An APK that supports a higher level API Level should have a higher version code.
- Do not leave gaps. For instance, an APK might support small and normal screen sizes and another APK might support xlarge screen sizes. Then devices with large screen sizes will not be able to see this application in Google Play because there is no APK that supports large screen sizes. Either one of the APKs should support large screen sizes, or a third APK should be created for large screen sizes.
- After creating APKs for different device configurations, we need to upload them to Google Play.

The default mode in the APK **Files** tab is simple mode. In this mode, only one APK can be active. The following screenshot is of the simple mode:



In order to upload multiple APKs, we need to switch to the advanced mode by clicking on the **Switch to advanced mode** link. You can't activate more than one APK in the simple mode; the developer's console doesn't allow this. In advanced mode, you can upload and activate more than one APK. You can upload APKs by clicking on the **Upload APK** button. The following screenshot is of the advanced mode:



In advanced mode, multiple APKs could be uploaded and activated.

Multiple APK or single APK

There are some drawbacks of multiple APKs. When you have a single APK, you only deal with that particular one, and it is easier than dealing with multiple APKs. If you have multiple APKs, you have to maintain multiple Android projects and source code. When you make a change in the application, you have to do the same change in each APK. However, if you have a single APK, you only maintain one source code and this is easier. In conclusion, maintaining a single APK is easier than maintaining multiple APKs. Furthermore, it is a best practice that you support different configurations by providing different resources with a single APK. Only one APK is necessary for a vast majority of apps. However, there may be some cases as in the *Configurations for Multiple APK* section that you need multiple APKs. Then you should make a trade-off and decide which way is suitable for your Android project.

Summary

In this chapter, we learned what multiple APK support is. Then, we learned the rules that we need to follow for creating multiple APKs and some of the very small use cases where this might become necessary.

