74. An Introduction to Android Instant Apps

The previous chapters covered Android App Links and explained how these links can be used to make the content of Android apps easier to discover and share with other users. App links alone, however, are only part of the solution. A significant limitation of app links is that an app link only works if the user already has the corresponding app installed on the Android device. This shortcoming is addressed by combining app links with the Android Instant App feature.

This chapter will provide an overview of Android Instant apps in terms of what they are and how they work. The following chapters will demonstrate how to implement Instant App support in both new and existing Android Studio projects.

74.1 An Overview of Android Instant Apps

A traditional Android app (also referred to as an *installed app*) consists of an APK file containing all of the various components that make up the app including classes, resource files and images. When development on the app is completed, the APK file is published to the Google Play store where prospective users can find and install the app onto their devices.

When an app makes use of Instant Apps, that app is divided into one or more *feature modules*, each of which is contained within a separate *feature APK* file. Each feature consists of a specific area of functionality within the app, typically involving one or more Activity instances. The individual feature APKs are then bundled into an *instant app APK* which is then uploaded to the Google Play Developer Console.

The features within an app are assigned App Links which can be used to launch the feature. When the link is clicked or used in an intent launch, Google Play matches the URL with the feature module, downloads only the required feature APK files and launches the entry point activity as specified in the APK manifest file. This allows the user to quickly gain access to a particular app feature without having to manually go to the Google Play store and install the entire app. The user simply clicks the link and Android Instant

Apps handles the rest.

Consider a hotel booking app that displays detailed hotel descriptions. A user with the app installed can send an app link to a friend to display information about a particular hotel. If the app supports Instant Apps and the friend does not already have the app installed, clicking the link will automatically download the APK file for the hotel detail feature of the app and launch it on the device.

To avoid cluttering devices with Instant App features, Android will typically remove infrequently used feature modules installed on a device.

74.2 Instant App Feature Modules

To support Instant Apps, a project needs to be divided into separate feature modules. A feature should contain at least one activity and represent a logical, standalone subset of the app's functionality. A feature module can, in fact, be thought of as a sharable library containing the code for a specific app feature.

All projects must contain one *base feature module*. If an app only consists of one feature, then the base feature module will contain all of the app's functionality. If an app has multiple features, each feature will have its own feature module in addition to the base module. In multi-feature apps, the base feature will typically contain one feature together with any resources that need to be shared with the requested feature module. When an instant app feature is requested, the base feature module is always downloaded in addition to the requested feature. This ensures that any shared resources are available for the requested feature module.

74.3 Instant App Project Structure

An Android Studio project needs to conform to a specific structure if it is to support instant apps. In fact, the project needs to be able to support both traditional installed apps and instant apps. This project structure consists of both an *app module* and an *instant app* module. The app module is responsible for building the standard installable APK file that is installed when the user taps the Install button in the Google Play store. The instant app module, on the other hand is responsible for generating each of the individual feature APK files.

Both the app module and the instant app module are essentially containers

for the feature modules that make up the app functionality. This ensures that the same code base is used for both installed and instant app variants. The build files for both modules simply declare the necessary feature modules as dependencies. Figure 74-1, for example, shows the structure for a simple multi-feature project:

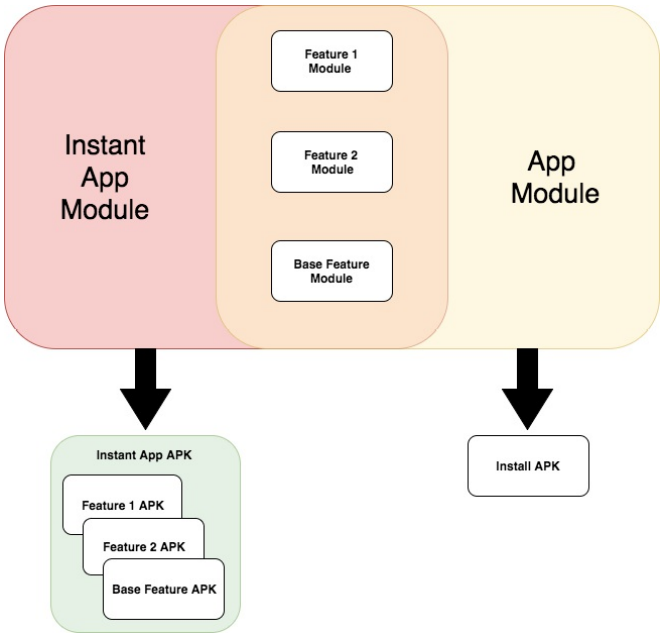

Figure 74-1

74.4 The Application and Feature Build Plugins

When a project is built, the build system uses the settings in the Gradle files for the app and instant app modules to decide how the output is to be structured. The *build.gradle* file for the app module will make use of the standard *com.android.application* plugin to build the single installable APK file, including in that file the feature modules declared in the dependencies section:

```
apply plugin: 'com.android.application'

android {
    compileSdkVersion 26
    buildToolsVersion "26.0.0"
.
.
dependencies {
    implementation project(':myappbase')
    implementation project(':myappdetail')
```

```
}
```

The *build.gradle* file for the instant app module, on the other hand, will use the *com.android.instantapp* plugin to build separate feature APK files for the features referenced in the dependencies section. Note that feature dependencies are referenced using *implementation project()* declarations:

```
apply plugin: 'com.android.instantapp'

dependencies {
    implementation project(':myappbase')
    implementation project(':myappfeature')
}
```

Each of the non-base feature modules that make up the app will also have a *build.gradle* file that uses the *com.android.feature* plugin, for example:

```
apply plugin: 'com.android.feature'

android {
    compileSdkVersion 26
    buildToolsVersion "26.0.0"
.
.
.
dependencies {
    implementation project(':myappbase')
}
```

The *build.gradle* file for the base feature module is a special case and must include a *baseFeature true* declaration. The file must also use the *feature project()* declaration for any feature module dependencies together with an *application project()* entry referencing the installed app module, for example:

```
apply plugin: 'com.android.feature'

android {

    baseFeature true

    compileSdkVersion 26
    buildToolsVersion "26.0.0"
.
.
.
dependencies {
```

```
    implementation fileTree(dir: 'libs', include: ['*.jar'])
.
.
    application project(':myappapk')
    feature project(':myappfeature')
}
```

74.5 Installing the Instant Apps Development SDK

Before working with Instant Apps in Android Studio, the Instant App must be installed. In preparation for the chapters that follows, launch Android Studio and select the *Configure -> SDK Manager* menu option (or use the *Tools -> Android -> SDK Manager* option if a project is already open).

Within the SDK manager screen, select the *SDK Tools* option and locate and enable the *Instant Apps Development SDK* entry:

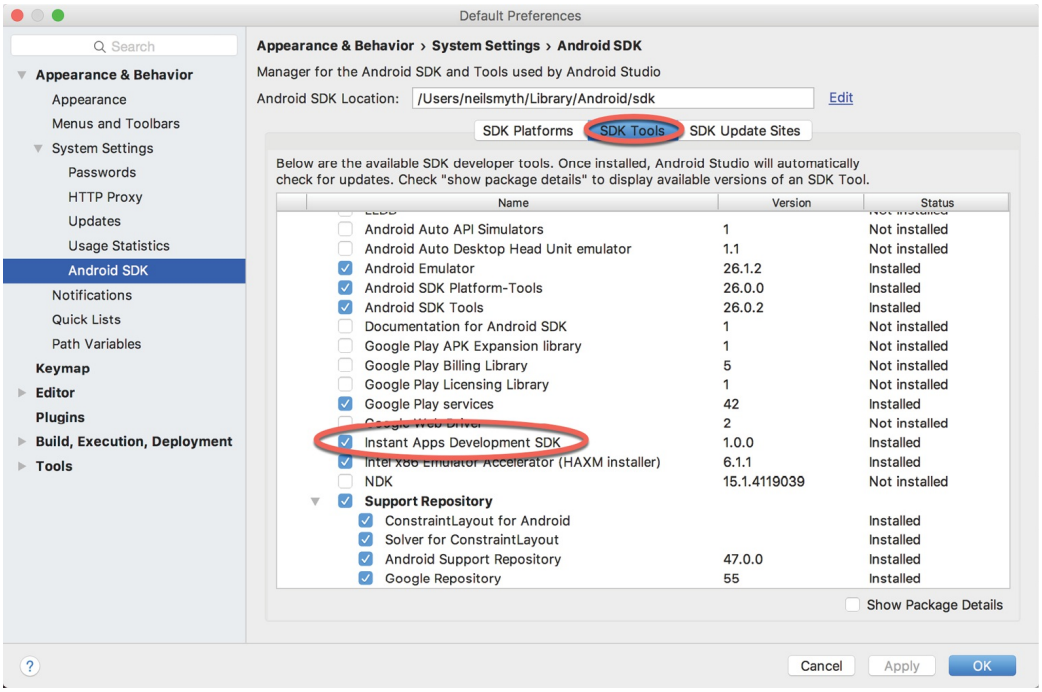

Figure 74-2

With the SDK selected, click on the OK button to perform the installation.

74.6 Summary

Android Instant Apps combine with Android App Links to provide an easy way for users to share and discover the content and features of apps. Instant apps are broken up into separate feature modules which can be launched using app links. When a link is selected, if the app is not already installed on

the user's device, the code for the app feature is downloaded by Google Play onto the device and launched. This allows app features to be run on demand without the need to manually install the entire app through the Google Play app.

Each app project must include an app module to contain the standard installable APK file and an instant app module for generating the separate feature APKs. Both the app and instant app modules serve as containers for the feature modules that make up the app. An app must contain at least one feature module and may also contain additional modules for other features.

With the basics of instant apps covered, the next chapter will explain how to add instant app support to a new Android Studio project.