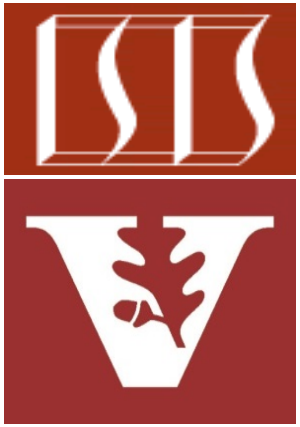
Android Intent Resolution & Filtering (Part 1)



Douglas C. Schmidt
d.schmidt@vanderbilt.edu
www.dre.vanderbilt.edu/~schmidt

Institute for Software
Integrated Systems
Vanderbilt University
Nashville, Tennessee, USA

Learning Objectives in this Part of the Lesson

1. Understand the purpose of the AndroidManifest.xml file

App Manifest

Every application must have an AndroidManifest.xml file (with precisely that name) in its root directory. The manifest file presents essential information about your app to the Android system, information the system must have before it can run any of the app's code. Among other things, the manifest does the following:

- It names the Java package for the application. The package name serves as a unique identifier for the application.
- It describes the components of the application — the activities, services, broadcast receivers, and content providers that the application is composed of. It names the classes that implement each of the components and publishes their capabilities (for example, which `Intent` messages they can handle). These declarations let the Android system know what the components are and under what conditions they can be launched.
- It determines which processes will host application components.
- It declares which permissions the application must have in order to access protected parts of the API and interact with other applications.
- It also declares the permissions that others are required to have in order to interact with the application's components.
- It lists the `Instrumentation` classes that provide profiling and other information as the application is running. These declarations are present in the manifest only while the application is being developed and tested; they're removed before the application is published.
- It declares the minimum level of the Android API that the application requires.

Knowledge of AndroidManifest.xml is necessary to understand intent filtering

Learning Objectives in this Part of the Lesson

1. Understand the purpose of the AndroidManifest.xml file
2. Recognize the structure & functionality of elements in AndroidManifest.xml

```
<manifest> …
  <application>
    <activity>
      <intent-filter>
       <action /> … <data />
      </…> …
    </…>
    <service>
      …
    </…>
    <receiver>
      <intent-filter> … </…>
    </…>
    <provider> … </…>
   …
```

Learning Objectives in this Part of the Lesson

1. Understand the purpose of the AndroidManifest.xml file
2. Recognize the structure & functionality of elements in AndroidManifest.xml

We'll discuss other aspects of this file further in upcoming lessons

```
<manifest> …
  <application>
    <activity>
      <intent-filter>
       <action /> … <data />
      </…> …
    </…>
    <service>
      …
    </…>
    <receiver>
      <intent-filter> … </…>
    </…>
    <provider> … </…>
  …
```

Learning Objectives in this Part of the Lesson

1. Understand the purpose of the AndroidManifest.xml file
2. Recognize the structure & functionality of elements in AndroidManifest.xml

```
<manifest> …
  <application>
    <activity>
      <intent-filter>
       <action /> … <data />
      </…> …
    </…>
    <service>
      …
    </…>
    <receiver>
      <intent-filter> … </…>
    </…>
    <provider> … </…>
   …
```

We'll show how AndroidManifest.xml relates to Intents in Part 2 of this lesson

Purpose of the AndroidManifest.xml File

Purpose of the AndroidManifest.xml File

- Android needs certain info to execute an app

```
<manifest> …
  <application>
    …
  </application>
</manifest>
```

See developer.android.com/training/basics/activity-lifecycle/starting.html

Purpose of the AndroidManifest.xml File

- Android needs certain info to execute an app, e.g.
  - Name of app Java package

```
<manifest …
  package="com.android.email">
  …
```

Purpose of the AndroidManifest.xml File

- Android needs certain info to execute an app, e.g.
  - Name of app Java package
  - App components, classes, & capabilities

```
<manifest> …
  <application>
    <activity>
      <intent-filter>
       <action /> … <data />
      </…> …
    </…>
    <service>
      …
    </…>
    <receiver>
      <intent-filter> … </…> …
    </…>
    <provider> … </…>
  …
```

Purpose of the AndroidManifest.xml File

- Android needs certain info to execute an app, e.g.
  - Name of app Java package
  - App components, classes, & capabilities, e.g.
    - Which components are exported

```
<manifest> …
  <application>
    <activity>
      <intent-filter>
       <action /> … <data />
      </…> …
    </…>
    <service>
      …
    </…>
    <receiver>
      <intent-filter> … </…> …
    </…>
    <provider> … </…>
  …
```

Purpose of the AndroidManifest.xml File

- Android needs certain info to execute an app, e.g.
  - Name of app Java package
  - App components, classes, & capabilities, e.g.
    - Which components are exported
    - Component names & which processes will host the components

```
<manifest> …
  <application>
    <activity>
      <intent-filter>
       <action /> … <data />
      </…> …
    </…>
    <service
     android:name=
       ".model.service.PalantiriService"
     android:process=":palantirService">
     …
    </…>
    …
```

Purpose of the AndroidManifest.xml File

- Android needs certain info to execute an app, e.g.
  - Name of app Java package
  - App components, classes, & capabilities, e.g.
    - Which components are exported
    - Component names & which processes will host the components
    - What intents are handled by components

```
<manifest> …
  <application>
    <activity>
      <intent-filter>
        <action android:name=
         "android.intent.action.MAIN"
        />
        <category android:name=
       "android.intent.category.LAUNCHER"
        />
      </…>
     </…>
   …
```

Purpose of the AndroidManifest.xml File

- Android needs certain info to execute an app, e.g.
  - Name of app Java package
  - App components, classes, & capabilities
  - Which permissions the app must have
    - & permissions other apps need to interact with this app components

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

Later versions of Android have changed so users grant permissions to apps while the app is running, not when they install the app

See developer.android.com/training/permissions/requesting.html

Purpose of the AndroidManifest.xml File

- Android needs certain info to execute an app, e.g.
  - Name of app Java package
  - App components, classes, & capabilities
  - Which permissions the app must have
  - Minimum & target API levels of Android the app requires

```
<manifest> …
   <uses-sdk
    android:minSdkVersion="19"
    android:targetSdkVersion="24"
    />
   …
```

Purpose of the AndroidManifest.xml File

- Android needs certain info to execute an app, e.g.
  - Name of app Java package
  - App components, classes, & capabilities
  - Which permissions the app must have
  - Minimum & target API levels of Android the app requires

```
<manifest> …
    <uses-sdk
     android:minSdkVersion="19"
     android:targetSdkVersion="24"
     />
…
```

```
android {
    ...
    defaultConfig {
        applicationId "…"
        minSdkVersion 24
        targetSdkVersion 24
        versionCode 1
        versionName "1.0"
        ...
```

Gradle can provide this information instead of putting it in AndroidManifest.xml

Purpose of the AndroidManifest.xml File

- Android needs certain info to execute an app, e.g.
  - Name of app Java package
  - App components, classes, & capabilities
  - Which permissions the app must have
  - Minimum & target API levels of Android the app requires
- etc.

```
<manifest> …
  <application>
    …
```

… Yadda Yadda Yadda …

```
  </application>
</manifest>
```

See developer.android.com/guide/topics/manifest/manifest-intro.html

Purpose of the AndroidManifest.xml File

- The AndroidManifest.xml file provides a central place to convey this information

App Manifest

Every application must have an AndroidManifest.xml file (with precisely that name) in its root directory. The manifest file presents essential information about your app to the Android system, information the system must have before it can run any of the app's code. Among other things, the manifest does the following:

- It names the Java package for the application. The package name serves as a unique identifier for the application.
- It describes the components of the application — the activities, services, broadcast receivers, and content providers that the application is composed of. It names the classes that implement each of the components and publishes their capabilities (for example, which `Intent` messages they can handle). These declarations let the Android system know what the components are and under what conditions they can be launched.
- It determines which processes will host application components.
- It declares which permissions the application must have in order to access protected parts of the API and interact with other applications.
- It also declares the permissions that others are required to have in order to interact with the application's components.
- It lists the `Instrumentation` classes that provide profiling and other information as the application is running. These declarations are present in the manifest only while the application is being developed and tested; they're removed before the application is published.
- It declares the minimum level of the Android API that the application requires.

See developer.android.com/guide/topics/manifest/manifest-intro.html

Elements in the AndroidManifest.xml File

Elements in the AndroidManifest.xml File

- <application> is a container that includes tags for specifying the app components

```
<application>
  <activity>
    <intent-filter> … </…>
  </…>
  <service>
    …
  </…>
  <receiver>
    <intent-filter> … </…>
  </…>
  <provider> … </…>
  …
</application>
```

See simpledeveloper.com/android-application-manifest-file

Elements in the AndroidManifest.xml File

- <application> is a container that includes tags for specifying the app components, e.g.
  - **Activities**
    - Implement part of app's visual user interface

```
<application>
  <activity>
    <intent-filter> … </…>
  </…>
  <service>
    …
  </…>
  <receiver>
    <intent-filter> … </…>
  </…>
  <provider> … </…>
  …
</application>
```

See developer.android.com/guide/topics/manifest/activity-element.html

Elements in the AndroidManifest.xml File

- <application> is a container that includes tags for specifying the app components, e.g.
  - **Activities**
    - Implement part of app's visual user interface
    - All activities *must* be represented by<activity> elements in manifest file

```
<application>
  <activity>
    <intent-filter> … </…>
  </…>
  <service>
    …
  </…>
  <receiver>
    <intent-filter> … </…>
  </…>
  <provider> … </…>
  …
</application>
```

See developer.android.com/guide/topics/manifest/activity-element.html

Elements in the AndroidManifest.xml File

- <application> is a container that includes tags for specifying the app components, e.g.
  - Activities
  - **Services**
    - Implement long-duration background operations

```
<application>
  <activity>
    <intent-filter> … </…>
  </…>
  <service>
    …
  </…>
  <receiver>
    <intent-filter> … </…>
  </…>
  <provider> … </…>
  …
</application>
```

See developer.android.com/guide/topics/manifest/service-element.html

Elements in the AndroidManifest.xml File

- <application> is a container that includes tags for specifying the app components, e.g.
  - Activities
  - **Services**
    - Implement long-duration background operations
    - All services *must* be represented by<service> elements in manifest file

```
<application>
  <activity>
    <intent-filter> … </…>
  </…>
  <service>
    …
  </…>
  <receiver>
    <intent-filter> … </…>
  </…>
  <provider> … </…>
  …
</application>
```

See developer.android.com/guide/topics/manifest/service-element.html

Elements in the AndroidManifest.xml File

- <application> is a container that includes tags for specifying the app components, e.g.
  - Activities
  - **Services**
    - Implement long-duration background operations
    - All services *must* be represented by<service> elements in manifest file
    - Services generally should not have intent filters

```
<application>
  <activity>
    <intent-filter> … </…>
  </…>
  <service>
    …
  </…>
  <receiver>
    <intent-filter> … </…>
  </…>
  <provider> … </…>
  …
</application>
```

See developer.android.com/about/versions/android-5.0-changes.html#BindService

Elements in the AndroidManifest.xml File

- <application> is a container that includes tags for specifying the app components, e.g.
  - Activities
  - Services
  - **Broadcast Receivers**
    - Apps can receive broadcast intents even when other app components aren't running

```
<application>
  <activity>
    <intent-filter> … </…>
  </…>
  <service>
    …
  </…>
  <receiver>
    <intent-filter> … </…>
  </…>
  <provider> … </…>
  …
</application>
```

See developer.android.com/guide/topics/manifest/receiver-element.html

Elements in the AndroidManifest.xml File

- <application> is a container that includes tags for specifying the app components, e.g.
  - Activities
  - Services
  - **Broadcast Receivers**
    - Apps can receive broadcast intents even when other app components aren't running
    - Use of manifest file is optional for receivers

```
<application>
  <activity>
    <intent-filter> … </…>
  </…>
  <service>
    …
  </…>
  <receiver>
    <intent-filter> … </…>
  </…>
  <provider> … </…>
  …
</application>
```

See developer.android.com/guide/topics/manifest/receiver-element.html

Elements in the AndroidManifest.xml File

- <application> is a container that includes tags for specifying the app components, e.g.
  - Activities
  - Services
  - Broadcast Receivers
  - **Content Providers**
    - Supply structured access to data managed by app

```
<application>
  <activity>
    <intent-filter> … </…>
  </…>
  <service>
    …
  </…>
  <receiver>
    <intent-filter> … </…>
  </…>
  <provider> … </…>
  …
</application>
```

See developer.android.com/guide/topics/manifest/provider-element.html

Elements in the AndroidManifest.xml File

- <application> is a container that includes tags for specifying the app components, e.g.
  - Activities
  - Services
  - Broadcast Receivers
  - **Content Providers**
    - Supply structured access to data managed by app
    - All app providers must be defined in a <provider> element in the manifest file

```
<application>
  <activity>
    <intent-filter> … </…>
  </…>
  <service>
    …
  </…>
  <receiver>
    <intent-filter> … </…>
  </…>
  <provider> … </…>
  …
</application>
```

See developer.android.com/guide/topics/manifest/provider-element.html

Elements in the AndroidManifest.xml File

- <application> is a container that includes tags for specifying the app components, e.g.
  - Activities
  - Services
  - Broadcast Receivers
  - Content Providers

```
<application>
  <activity>
    <intent-filter> … </…>
  </…>
  <service>
    …
  </…>
  <receiver>
    <intent-filter> … </…>
  </…>
  <provider> … </…>
  …
</application>
```

The next part of lesson focuses on intent filters defined in AndroidManifest.xml

End of Android Intent Resolution & Filtering (Part 1)