

What Makes an Android Application?

Android applications consist of loosely coupled components, bound using a project manifest that describes each component and how they interact.

There are six components that provide the building blocks for your applications:

- ❑ **Activities** Your application's presentation layer. Every screen in your application will be an extension of the Activity class. Activities use Views to form graphical user interfaces that display information and respond to user actions. In terms of desktop development, an Activity is equivalent to a Form. You'll learn more about Activities later in this chapter.
- ❑ **Services** The invisible workers of your application. Service components run invisibly, updating your data sources and visible Activities and triggering Notifications. They're used to perform regular processing that needs to continue even when your application's Activities aren't active or visible. You'll learn how to create Services in Chapter 8.
- ❑ **Content Providers** A shareable data store. Content Providers are used to manage and share application databases. Content Providers are the preferred way of sharing data across application boundaries. This means that you can configure your own Content Providers to permit access from other applications and use Content Providers exposed by others to access their stored data. Android devices include several native Content Providers that expose useful databases like contact information. You'll learn how to create and use Content Providers in Chapter 6.
- ❑ **Intents** A simple message-passing framework. Using Intents, you can broadcast messages system-wide or to a target Activity or Service, stating your intention to have an action performed. The system will then determine the target(s) that will perform any actions as appropriate.
- ❑ **Broadcast Receivers** Intent broadcast consumers. By creating and registering a Broadcast Receiver, your application can listen for broadcast Intents that match specific filter criteria. Broadcast Receivers will automatically start your application to respond to an incoming Intent, making them ideal for event-driven applications.
- ❑ **Notifications** A user notification framework. Notifications let you signal users without stealing focus or interrupting their current Activities. They're the preferred technique for getting a user's attention from within a Service or Broadcast Receiver. For example, when a device receives a text message or an incoming call, it alerts you by flashing lights, making sounds, displaying icons, or showing dialog messages. You can trigger these same events from your own applications using Notifications, as shown in Chapter 8.

By decoupling the dependencies between application components, you can share and interchange individual pieces, such as Content Providers or Services, with other applications — both your own and those of third parties.