In the late 1990s, a wave of viruses spread through the Internet, delivered via e-mail, using contact information culled from Microsoft Outlook. A virus would simply e-mail copies of itself to each of the Outlook contacts that had an e-mail address. This was possible because, at the time, Outlook did not take any steps to protect data from programs using the Outlook API, since that API was designed for ordinary developers, not virus authors.

Nowadays, many applications that hold onto contact data secure that data by requiring that a user explicitly grant rights for other programs to access the contact information. Those rights could be granted on a case-by-case basis or all at once at install time.

Android is no different, in that it requires permissions for applications to read or write contact data. Android’s permission system is useful well beyond contact data, and for content providers and services beyond those supplied by the Android framework.

You, as an Android developer, will frequently need to ensure that your applications have the appropriate permissions to do what you want to do with other applications’ data. You may also elect to require permissions for other applications to use your data or services, if you make those available to other Android components. This chapter covers how to accomplish both these ends.

**Mother, May I?**

Requesting the use of other applications’ data or services requires the uses-permission element to be added to your AndroidManifest.xml file. Your manifest may have zero or more uses-permission elements, all as direct children of the root manifest element.
The uses-permission element takes a single attribute, android:name, which is the name of the permission your application requires:

```xml
<uses-permission
    android:name="android.permission.ACCESS_LOCATION" />
```

All of the stock system permissions begin with android.permission and are listed in the Android SDK documentation for Manifest.permission. Third-party applications may have their own permissions, which, hopefully, they have documented for you. Here are some of the more useful permissions:

- INTERNET, if your application wishes to access the Internet through any means, from raw Java sockets through the WebView widget
- WRITE_EXTERNAL_STORAGE, for writing data to the SD card (or whatever the device has designated as external storage)
- NFC, for performing I/O with the near-field communication (NFC) radio on newer devices
- ACCESS_COARSE_LOCATION and ACCESS_FINE_LOCATION, for determining where the device is located
- CALL_PHONE, to allow the application to place phone calls directly, without user intervention

Permissions are confirmed at the time the application is installed. The user will be prompted to confirm that it is acceptable for your application to do what the permission calls for. Hence, it is important that you ask for as few permissions as possible and justify those you seek, so users do not elect to skip installing your application because you ask for too many unnecessary permissions. This prompt will not appear when loading an application via USB, such as during development.

If you do not have the desired permission and try to do something that needs it, you should get a SecurityException informing you of the missing permission. Note that you will fail on a permission check only if you forgot to ask for the permission—it is impossible for your application to be running and not have been granted your requested permissions.

**Halt! Who Goes There?**

The other side of the coin is to secure your own application. If your application is mostly activities, security may be just an “outbound” thing, where you request the right to use resources of other applications. If, on the other hand, you put content providers or services in your application, you will want to implement “inbound” security to control which applications can do what with the data.

Note that the issue here is less about whether other applications might mess up your data, but rather about privacy of the user’s information or use of services that might incur expense. That is where the stock permissions for built-in Android applications are focused: whether you can read or modify contacts, send SMS messages, and so forth. If