

Versioning Your Applications

Versioning is a critical component of your application upgrade/maintenance strategy.

- Users need to have specific information about the application version that is installed on their devices and the upgrade versions available for installation.
- Other applications — including other applications that you publish as a suite — need to query the system for your application's version, to determine compatibility and identify dependencies.
- Services through which you will publish your application(s) may also need to query your application for its version, so that they can display the version to users. A publishing service may also need to check the application version to determine compatibility and establish upgrade/downgrade relationships.

The Android system itself *does not ever* check the application version information for an application, such as to enforce restrictions on upgrades, compatibility, and so on. Instead, only users or applications themselves are responsible for enforcing any version restrictions for applications themselves.

The Android system *does* check any system version compatibility expressed by an application in its manifest, in the `minSdkVersion` attribute. This allows an application to specify the minimum system API with which is compatible. For more information see [Specifying Minimum System API Version](#).

Quickview

- Your application *must* be versioned
- You set the version in the application's manifest file
- How you version your applications affects how users upgrade
- Determine your versioning strategy early in the development process, including considerations for future releases.

In this document

[Setting Application Version](#)

[Specifying Your Application's System API Requirements](#)

See also

[Preparing to Publish Your Application](#)

[Publishing On Android Market](#)

[The AndroidManifest.xml File](#)

Setting Application Version

To define the version information for your application, you set attributes in the application's manifest file. Two attributes are available, and you should always define values for both of them:

- `android:versionCode` — An integer value that represents the version of the application code, relative to other versions.

The value is an integer so that other applications can programmatically evaluate it, for example to check an upgrade or downgrade relationship. You can set the value to any integer you want, however you should make sure that each successive release of your application uses a greater value. The system does not enforce this behavior, but increasing the value with successive releases is normative.

Typically, you would release the first version of your application with `versionCode` set to 1, then monotonically increase the value with each release, regardless whether the release constitutes a major or minor release. This means that the `android:versionCode` value does not necessarily have a strong resemblance to the application release version that is visible to the user (see `android:versionName`, below). Applications and publishing services should not display this version value to users.

- `android:versionName` — A string value that represents the release version of the application code, as it should be shown to users.

The value is a string so that you can describe the application version as a `<major>.<minor>.<point>` string, or as any other type of absolute or relative version identifier.

As with `android:versionCode`, the system does not use this value for any internal purpose, other than to enable applications to display it to users. Publishing services may also extract the `android:versionName` value for display to users.

You define both of these version attributes in the `<manifest>` element of the manifest file.

Here's an example manifest that shows the `android:versionCode` and `android:versionName` attributes in the `<manifest>` element.

```
<?xml version="1.0" encoding="utf-8"?>
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="com.example.package.name"
    android:versionCode="2"
    android:versionName="1.1">
    <application android:icon="@drawable/icon" android:label="@string/app_name">
        ...
    </application>
</manifest>
```

In this example, note that `android:versionCode` value indicates that the current .apk contains the second release of the application code, which corresponds to a minor follow-on release, as shown by the `android:versionName` string.

The Android framework provides an API to let applications query the system for version information about your application. To obtain version information, applications use the [getPackageInfo\(java.lang.String, int\)](#) method of [PackageManager](#).

Specifying Your Application's System API Requirements

If your application requires a specific minimum version of the Android platform, or is designed only to support a certain range of Android platform versions, you can specify those version requirements as API Level identifiers in the application's manifest file. Doing so ensures that your application can only be installed on devices that are running a compatible version of the Android system.

To specify API Level requirements, add a `<uses-sdk>` element in the application's manifest, with one or more of these attributes:

- `android:minSdkVersion` — The minimum version of the Android platform on which the application will run, specified by the platform's API Level identifier.
- `android:targetSdkVersion` — Specifies the API Level on which the application is designed to run. In some cases, this allows the application to use manifest elements or behaviors defined in the target API Level, rather than being restricted to using only those defined for the minimum API Level.
- `android:maxSdkVersion` — The maximum version of the Android platform on which the application is designed to run, specified by the platform's API Level identifier. **Important:** Please read the [<uses-sdk>](#) documentation before using this attribute.

When preparing to install your application, the system checks the value of this attribute and compares it to the system version. If the `android:minSdkVersion` value is greater than the system version, the system aborts the installation of the application. Similarly, the system installs your application only if its `android:maxSdkVersion` is compatible with the platform version.

If you do not specify these attributes in your manifest, the system assumes that your application is compatible with all platform versions, with no maximum API Level.

To specify a minimum platform version for your application, add a `<uses-sdk>` element as a child of `<manifest>`, then define the `android:minSdkVersion` as an attribute.

For more information, see the [<uses-sdk>](#) manifest element documentation and the [API Levels](#) document.

Except as noted, this content is licensed under [Apache 2.0](#). For details and restrictions, see the [Content License](#).

[↑ Go to top](#)

Android 3.1 r1 - 17 Jun 2011 10:58

[Site Terms of Service](#) - [Privacy Policy](#) - [Brand Guidelines](#)