



<manifest>

SYNTAX:

```
<manifest xmlns:android="http://schemas.android.com/apk/res/android"
    package="string"
    android:sharedUserId="string"
    android:sharedUserLabel="string resource"
    android:versionCode="integer"
    android:versionName="string"
    android:installLocation=["auto" | "internalOnly" | "preferExternal"] >
    . . .
</manifest>
```

CONTAINED IN:

none

MUST CONTAIN:

[<application>](#)

CAN CONTAIN:

[<compatible-screens>](#)
[<instrumentation>](#)
[<permission>](#)
[<permission-group>](#)
[<permission-tree>](#)
[<supports-gl-texture>](#)
[<supports-screens>](#)
[<uses-configuration>](#)
[<uses-feature>](#)
[<uses-permission>](#)
[<uses-permission-sdk-23>](#)
[<uses-sdk>](#)

DESCRIPTION:

The root element of the AndroidManifest.xml file. It must contain an [<application>](#) element and specify `xmlns:android` and `package` attributes.

ATTRIBUTES:

`xmlns:android`

Defines the Android namespace. This attribute should always be set to "[http://schemas.android.com/apk/res/android](#)".

`package`

A full Java-language-style package name for the Android application. The name should be unique. The name may contain uppercase or lowercase letters ('A' through 'Z'), numbers, and underscores ('_'). However, individual package name parts may only start with letters.

To avoid conflicts with other developers, you should use Internet domain ownership as the basis for your package names (in reverse). For example, applications published by Google start with `com.google`. You should also never use the `com.example` namespace when publishing your applications.

The package name serves as a unique identifier for the application. It's also the default name for the application process (see the `<application>` element's `process` process attribute) and the default task affinity of an activity (see the `<activity>` element's `taskAffinity` attribute).

Caution: Once you publish your application, you **cannot change the package name**. The package name defines your application's identity, so if you change it, then it is considered to be a different application and users of the previous version cannot update to the new version.

`android:sharedUserId`

The name of a Linux user ID that will be shared with other applications. By default, Android assigns each application its own unique user ID. However, if this attribute is set to the same value for two or more applications, they will all share the same ID — provided that their certificate sets are identical. Applications with the same user ID can access each other's data and, if desired, run in the same process.

`android:sharedUserLabel`

A user-readable label for the shared user ID. The label must be set as a reference to a string resource; it cannot be a raw string.

This attribute was introduced in API Level 3. It is meaningful only if the `sharedUserId` attribute is also set.

`android:versionCode`

An internal version number. This number is used only to determine whether one version is more recent than another, with higher numbers indicating more recent versions. This is not the version number shown to users; that number is set by the `versionName` attribute.

The value must be set as an integer, such as "100". You can define it however you want, as long as each successive version has a higher number. For example, it could be a build number. Or you could translate a version number in "x.y" format to an integer by encoding the "x" and "y" separately in the lower and upper 16 bits. Or you could simply increase the number by one each time a new version is released.

`android:versionName`

The version number shown to users. This attribute can be set as a raw string or as a reference to a string resource. The string has no other purpose than to be displayed to users. The `versionCode` attribute holds the significant version number used internally.

`android:installLocation`

The default install location for the application.

The following keyword strings are accepted:

Value	Description
<code>"internalOnly"</code>	The application must be installed on the internal device storage only. If this is set, the application will never be installed on the external storage. If the internal storage is full, then the system will not install the application. This is also the default behavior if you do not define <code>android:installLocation</code> .
<code>"auto"</code>	The application may be installed on the external storage, but the system will install the application on the internal storage by default. If the internal storage is full, then the system will install it on the external storage. Once installed, the user can move the application to either internal or external storage through the system settings.
<code>"preferExternal"</code>	The application prefers to be installed on the external storage (SD card). There is no guarantee that the system will honor this request. The application might be installed on internal storage if the

	external media is unavailable or full. Once installed, the user can move the application to either internal or external storage through the system settings.
--	--

Note: By default, your application will be installed on the internal storage and cannot be installed on the external storage unless you define this attribute to be either `"auto"` or `"preferExternal"`.

When an application is installed on the external storage:

- The `.apk` file is saved to the external storage, but any application data (such as databases) is still saved on the internal device memory.
- The container in which the `.apk` file is saved is encrypted with a key that allows the application to operate only on the device that installed it. (A user cannot transfer the SD card to another device and use applications installed on the card.) Though, multiple SD cards can be used with the same device.
- At the user's request, the application can be moved to the internal storage.

The user may also request to move an application from the internal storage to the external storage. However, the system will not allow the user to move the application to external storage if this attribute is set to `internalOnly`, which is the default setting.

Read [App Install Location](#) for more information about using this attribute (including how to maintain backward compatibility).

Introduced in: API Level 8.

INTRODUCED IN:

API Level 1 for all attributes, unless noted otherwise in the attribute description.

SEE ALSO:

`<application>`