

<application>

SYNTAX:

```
<application android:allowTaskReparenting=["true" | "false"]
    android:backupAgent="string"
    android:debuggable=["true" | "false"]
    android:description="string resource"
    android:enabled=["true" | "false"]
    android:hasCode=["true" | "false"]
    android:hardwareAccelerated=["true" | "false"]
    android:icon="drawable resource"
    android:killAfterRestore=["true" | "false"]
    android:label="string resource"
    android:logo="drawable resource"
    android:manageSpaceActivity="string"
    android:name="string"
    android:permission="string"
    android:persistent=["true" | "false"]
    android:process="string"
    android:restoreAnyVersion=["true" | "false"]
    android:taskAffinity="string"
    android:theme="resource or theme" >
    .
    .
    .
</application>
```

CONTAINED IN:

[<manifest>](#)

CAN CONTAIN:

[<activity>](#)
[<activity-alias>](#)
[<service>](#)
[<receiver>](#)
[<provider>](#)
[<uses-library>](#)

DESCRIPTION:

The declaration of the application. This element contains subelements that declare each of the application's components and has attributes that can affect all the components. Many of these attributes (such as `icon`, `label`, `permission`, `process`, `taskAffinity`, and `allowTaskReparenting`) set default values for corresponding attributes of the component elements. Others (such as `debuggable`, `enabled`, `description`, and `allowClearUserData`) set values for the application as a whole and cannot be overridden by the components.

ATTRIBUTES

`android:allowTaskReparenting`

Whether or not activities that the application defines can move from the task that started them to the task they have an affinity for when that task is next brought to the front — "true" if they can move, and "false" if they must remain with the task where they started. The default value is "false".

The `<activity>` element has its own `allowTaskReparenting` attribute that can override the value set here. See that attribute for more information.

`android:backupAgent`

The name of the class that implements the application's backup agent, a subclass of [BackupAgent](#). The attribute value should be a fully qualified class name (such as, `"com.example.project.MyBackupAgent"`). However, as a shorthand, if the first character of the name is a period (for example, `".MyBackupAgent"`), it is appended to the package name specified in the [<manifest>](#) element.

There is no default. The name must be specified.

`android:debuggable`

Whether or not the application can be debugged, even when running on a device in user mode — `"true"` if it can be, and `"false"` if not. The default value is `"false"`.

`android:description`

User-readable text about the application, longer and more descriptive than the application label. The value must be set as a reference to a string resource. Unlike the label, it cannot be a raw string. There is no default value.

`android:enabled`

Whether or not the Android system can instantiate components of the application — `"true"` if it can, and `"false"` if not. If the value is `"true"`, each component's `enabled` attribute determines whether that component is enabled or not. If the value is `"false"`, it overrides the component-specific values; all components are disabled.

The default value is `"true"`.

`android:hasCode`

Whether or not the application contains any code — `"true"` if it does, and `"false"` if not. When the value is `"false"`, the system does not try to load any application code when launching components. The default value is `"true"`.

An application would not have any code of its own only if it's using nothing but built-in component classes, such as an activity that uses the [AliasActivity](#) class, a rare occurrence.

`android:hardwareAccelerated`

Whether or not hardware-accelerated rendering should be enabled for all Activities and Views in this application — `"true"` if it should be enabled, and `"false"` if not. The default value is `"false"`.

Starting from Android 3.0, a hardware-accelerated OpenGL renderer is available to applications, to improve performance for many common 2D graphics operations. When the hardware-accelerated renderer is enabled, most operations in Canvas, Paint, Xfermode, ColorFilter, Shader, and Camera are accelerated. This results in smoother animations, smoother scrolling, and improved responsiveness overall, even for applications that do not explicitly make use of the framework's OpenGL libraries.

Note that not all of the OpenGL 2D operations are accelerated. If you enable the hardware-accelerated renderer, test your application to ensure that it can make use of the renderer without errors.

`android:icon`

An icon for the application as whole, and the default icon for each of the application's components. See the individual `icon` attributes for [<activity>](#), [<activity-alias>](#), [<service>](#), [<receiver>](#), and [<provider>](#) elements.

This attribute must be set as a reference to a drawable resource containing the image (for example `"@drawable/icon"`). There is no default icon.

`android:killAfterRestore`

Whether the application in question should be terminated after its settings have been restored during a full-system restore operation. Single-package restore operations will never cause the application to be shut down. Full-system restore operations typically only occur once, when the phone is first set up. Third-party applications will not normally need to use this attribute.

The default is `true`, which means that after the application has finished processing its data during a full-system restore, it will be terminated.

`android:label`

A user-readable label for the application as a whole, and a default label for each of the application's components. See the individual `label` attributes for [<activity>](#), [<activity-alias>](#), [<service>](#), [<receiver>](#), and [<provider>](#) elements.

The label should be set as a reference to a string resource, so that it can be localized like other strings in the user interface. However, as a convenience while you're developing the application, it can also be set as a raw string.

`android:logo`

A logo for the application as whole, and the default logo for activities.

This attribute must be set as a reference to a drawable resource containing the image (for example `"@drawable/logo"`). There is no default logo.

`android:manageSpaceActivity`

The fully qualified name of an Activity subclass that the system can launch to let users manage the memory occupied by the application on the device. The activity should also be declared with an [<activity>](#) element.

`android:name`

The fully qualified name of an [Application](#) subclass implemented for the application. When the application process is started, this class is instantiated before any of the application's components.

The subclass is optional; most applications won't need one. In the absence of a subclass, Android uses an instance of the base Application class.

`android:permission`

The name of a permission that clients must have in order to interact with the application. This attribute is a convenient way to set a permission that applies to all of the application's components. It can be overwritten by setting the `permission` attributes of individual components.

For more information on permissions, see the [Permissions](#) section in the introduction and another document, [Security and Permissions](#).

`android:persistent`

Whether or not the application should remain running at all times — `"true"` if it should, and `"false"` if not. The default value is `"false"`. Applications should not normally set this flag; persistence mode is intended only for certain system applications.

`android:process`

The name of a process where all components of the application should run. Each component can override this default by setting its own `process` attribute.

By default, Android creates a process for an application when the first of its components needs to run. All components then run in that process. The name of the default process matches the package name set by the [<manifest>](#) element.

By setting this attribute to a process name that's shared with another application, you can arrange for components of both applications to run in the same process — but only if the two applications also share a user ID and be signed with the same certificate.

If the name assigned to this attribute begins with a colon (':'), a new process, private to the application, is created when it's needed. If the process name begins with a lowercase character, a global process of that name is created. A global process can be shared with other applications, reducing resource usage.

`android:restoreAnyVersion`

Indicate that the application is prepared to attempt a restore of any backed-up data set, even if the backup was stored by a newer version of the application than is currently installed on the device. Setting this attribute to `true` will permit the Backup Manager to attempt restore even when a version mismatch suggests that the data are incompatible. *Use with caution!*

The default value of this attribute is `false`.

`android:taskAffinity`

An affinity name that applies to all activities within the application, except for those that set a different affinity with their own [taskAffinity](#) attributes. See that attribute for more information.

By default, all activities within an application share the same affinity. The name of that affinity is the same as the package name set by the [<manifest>](#) element.

`android:theme`

A reference to a style resource defining a default theme for all activities in the application. Individual activities can override the default by setting their own [theme](#) attributes; see that attribute for more information.

INTRODUCED IN:

API Level 1

SEE ALSO:

[<activity>](#)

[<service>](#)

[<receiver>](#)

[<provider>](#)

[← Back to The AndroidManifest.xml File](#)

[↑ Go to top](#)

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

Android 3.1 r1 - 17 Jun 2011 10:58

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