<uses-library>

SYNTAX:

```
<uses-library
  android:name="string"
  android:required=["true" | "false"] />
```

CONTAINED IN:

`<application>`

Google Play Filtering

Google Play uses the <uses-library> elements declared in your app manifest to filter your app from devices that do not meet its library requirements. For more information about filtering, see the topic Google Play filters.

DESCRIPTION:

Specifies a shared library that the application must be linked against. This element tells the system to include the library's code in the class loader for the package.

All of the `android` packages (such as `android.app`, `android.content`, `android.view`, and `android.widget`) are in the default library that all applications are automatically linked against. However, some packages (such as `maps`) are in separate libraries that are not automatically linked. Consult the documentation for the packages you're using to determine which library contains the package code.

This element also affects the installation of the application on a particular device and the availability of the application on Google Play:

Installation

If this element is present and its `android:required` attribute is set to `true`, the `PackageManager` framework won't let the user install the application unless the library is present on the user's device.

The `android:required` attribute is described in detail in the following section.

ATTRIBUTES:

`android:name`

The name of the library. The name is provided by the documentation for the package you are using. An example of this is "`android.test.runner`", a package that contains Android test classes.

`android:required`

Boolean value that indicates whether the application requires the library specified by `android:name`:

- `"true"`: The application does not function without this library. The system will not allow the application on a device that does not have the library.
- `"false"`: The application can use the library if present, but is designed to function without it if necessary. The system will allow the application to be installed, even if the library is not present. If you use `"false"`, you are responsible for checking at runtime that the library is available.

  To check for a library, you can use reflection to determine if a particular class is available.

The default is `"true"`.

Introduced in: API Level 7.

INTRODUCED IN:

API Level 1

SEE ALSO:

- `PackageManager`