



Menu Resource

See also

> [Menus](#)

A menu resource defines an application menu (Options Menu, Context Menu, or submenu) that can be inflated with [MenuInflater](#).

For a guide to using menus, see the [Menus](#) developer guide.

FILE LOCATION:

`res/menu/filename.xml`

The filename will be used as the resource ID.

COMPILED RESOURCE DATATYPE:

Resource pointer to a [Menu](#) (or subclass) resource.

RESOURCE REFERENCE:

In Java: `R.menu.filename`

In XML: `@[package:]menu.filename`

SYNTAX:

```

<?xml version="1.0" encoding="utf-8"?>
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+[package:]id/resource_name"
        android:title="string"
        android:titleCondensed="string"
        android:icon="@[package:]drawable/drawable_resource_name"
        android:onClick="method_name"
        android:showAsAction=["ifRoom" | "never" | "withText" | "always" | "collapseActionView"]
        android:actionLayout="@[package:]layout/layout_resource_name"
        android:actionViewClass="class_name"
        android:actionProviderClass="class_name"
        android:alphabeticShortcut="string"
        android:alphabeticModifiers=["META" | "CTRL" | "ALT" | "SHIFT" | "SYM" | "FUNCTION"]
        android:numericShortcut="string"
        android:numericModifiers=["META" | "CTRL" | "ALT" | "SHIFT" | "SYM" | "FUNCTION"]
        android:checkable=["true" | "false"]
        android:visible=["true" | "false"]
        android:enabled=["true" | "false"]
        android:menuCategory=["container" | "system" | "secondary" | "alternative"]
        android:orderInCategory="integer" />
  <group android:id="@+[package:]id/resource_name"
        android:checkableBehavior=["none" | "all" | "single"]
        android:visible=["true" | "false"]
        android:enabled=["true" | "false"]
        android:menuCategory=["container" | "system" | "secondary" | "alternative"]
        android:orderInCategory="integer" >
    <item />
  </group>
  <item >
    <menu>
      <item />
    </menu>
  </item>
</menu>

```

ELEMENTS:

<menu>

Required. This must be the root node. Contains <item> and/or <group> elements.

attributes:

xmlns:android

XML namespace. **Required.** Defines the XML namespace, which must be "http://schemas.android.com/apk/res/android".

<item>

A menu item. May contain a <menu> element (for a Sub Menu). Must be a child of a <menu> or <group> element.

attributes:

android:id

Resource ID. A unique resource ID. To create a new resource ID for this item, use the form: "@+id/name". The plus symbol indicates that this should be created as a new ID.

android:title

String resource. The menu title as a string resource or raw string.

android:titleCondensed

String resource. A condensed title as a string resource or a raw string. This title is used for situations in which the normal

title is too long.

`android:icon`

Drawable resource. An image to be used as the menu item icon.

`android:onClick`

Method name. The method to call when this menu item is clicked. The method must be declared in the activity as public and accept a [MenuItem](#) as its only parameter, which indicates the item clicked. This method takes precedence over the standard callback to [onOptionsItemSelected\(\)](#). See the example at the bottom.

Warning: If you obfuscate your code using [ProGuard](#) (or a similar tool), be sure to exclude the method you specify in this attribute from renaming, because it can break the functionality.

Introduced in API Level 11.

`android:showAsAction`

Keyword. When and how this item should appear as an action item in the app bar. A menu item can appear as an action item only when the activity includes an app bar. Valid values:

Value	Description
<code>ifRoom</code>	Only place this item in the app bar if there is room for it. If there is not room for all the items marked " <code>ifRoom</code> ", the items with the lowest <code>orderInCategory</code> values are displayed as actions, and the remaining items are displayed in the overflow menu.
<code>withText</code>	Also include the title text (defined by <code>android:title</code>) with the action item. You can include this value along with one of the others as a flag set, by separating them with a pipe <code> </code> .
<code>never</code>	Never place this item in the app bar. Instead, list the item in the app bar's overflow menu.
<code>always</code>	Always place this item in the app bar. Avoid using this unless it's critical that the item always appear in the action bar. Setting multiple items to always appear as action items can result in them overlapping with other UI in the app bar.
<code>collapseActionView</code>	The action view associated with this action item (as declared by <code>android:actionLayout</code> or <code>android:actionViewClass</code>) is collapsible. Introduced in API Level 14.

See the [Adding the App Bar](#) training class for more information.

Introduced in API Level 11.

`android:actionLayout`

Layout resource. A layout to use as the action view.

See [Action Views and Action Providers](#) for more information.

Introduced in API Level 11.

`android:actionViewClass`

Class name. A fully-qualified class name for the [View](#) to use as the action view. For example, "`android.widget.SearchView`" to use [SearchView](#) as an action view.

See [Action Views and Action Providers](#) for more information.

Warning: If you obfuscate your code using [ProGuard](#) (or a similar tool), be sure to exclude the class you specify in this attribute from renaming, because it can break the functionality.

Introduced in API Level 11.

`android:actionProviderClass`

Class name. A fully-qualified class name for the [ActionProvider](#) to use in place of the action item. For example, "`android.widget.ShareActionProvider`" to use [ShareActionProvider](#).

See [Action Views and Action Providers](#) for more information.

Warning: If you obfuscate your code using [ProGuard](#) (or a similar tool), be sure to exclude the class you specify in this attribute from renaming, because it can break the functionality.

Introduced in API Level 14.

`android:alphabeticShortcut`

Char. A character for the alphabetic shortcut key.

`android:numericShortcut`

Integer. A number for the numeric shortcut key.

`android:alphabeticModifiers`

Keyword. A modifier for the menu item's alphabetic shortcut. The default value corresponds to the Control key. Valid values:

Value	Description
<code>META</code>	Corresponds to the Meta meta key
<code>CTRL</code>	Corresponds to the Control meta key
<code>ALT</code>	Corresponds to the Alt meta key
<code>SHIFT</code>	Corresponds to the Shift meta key
<code>SYM</code>	Corresponds to the Sym meta key
<code>FUNCTION</code>	Corresponds to the Function meta key

Note: You can specify multiple keywords in an attribute. For example, `android:alphabeticModifiers="CTRL|SHIFT"` indicates that to trigger the corresponding menu item, the user needs to press both Control and Shift meta keys along with the shortcut.

You can use the `setAlphabeticShortcut()` method to set the attribute values programmatically. For more information about the `alphabeticModifier` attribute, go to [alphabeticModifiers](#).

`android:numericModifiers`

Keyword. A modifier for the menu item's numeric shortcut. The default value corresponds to the Control key. Valid values:

Value	Description
<code>META</code>	Corresponds to the Meta meta key
<code>CTRL</code>	Corresponds to the Control meta key
<code>ALT</code>	Corresponds to the Alt meta key
<code>SHIFT</code>	Corresponds to the Shift meta key
<code>SYM</code>	Corresponds to the Sym meta key
<code>FUNCTION</code>	Corresponds to the Function meta key

Note: You can specify multiple keywords in an attribute. For example, `android:numericModifiers="CTRL|SHIFT"` indicates that to trigger the corresponding menu item, the user needs to press both Control and Shift meta keys along with the shortcut.

You can use the `setNumericShortcut()` method to set the attribute values programmatically. For more information about the `numericModifier` attribute, go to [numericModifiers](#).

`android:checkable`

Boolean. "true" if the item is checkable.

`android:checked`

Boolean. "true" if the item is checked by default.

`android:visible`

Boolean. "true" if the item is visible by default.

`android:enabled`

Boolean. "true" if the item is enabled by default.

`android:menuCategory`

Keyword. Value corresponding to `Menu CATEGORY_*` constants, which define the item's priority. Valid values:

Value	Description
<code>container</code>	For items that are part of a container.
<code>system</code>	For items that are provided by the system.
<code>secondary</code>	For items that are user-supplied secondary (infrequently used) options.
<code>alternative</code>	For items that are alternative actions on the data that is currently displayed.

`android:orderInCategory`

Integer. The order of "importance" of the item, within a group.

`<group>`

A menu group (to create a collection of items that share traits, such as whether they are visible, enabled, or checkable). Contains one or more `<item>` elements. Must be a child of a `<menu>` element.

attributes:

`android:id`

Resource ID. A unique resource ID. To create a new resource ID for this item, use the form: "`@+id/name`". The plus symbol indicates that this should be created as a new ID.

`android:checkableBehavior`

Keyword. The type of checkable behavior for the group. Valid values:

Value	Description
<code>none</code>	Not checkable
<code>all</code>	All items can be checked (use checkboxes)
<code>single</code>	Only one item can be checked (use radio buttons)

`android:visible`

Boolean. "true" if the group is visible.

`android:enabled`

Boolean. "true" if the group is enabled.

`android:menuCategory`

Keyword. Value corresponding to `Menu CATEGORY_*` constants, which define the group's priority. Valid values:

Value	Description
<code>container</code>	For groups that are part of a container.
<code>system</code>	For groups that are provided by the system.
<code>secondary</code>	For groups that are user-supplied secondary (infrequently used) options.
<code>alternative</code>	For groups that are alternative actions on the data that is currently displayed.

`android:orderInCategory`

Integer. The default order of the items within the category.

EXAMPLE:

XML file saved at `res/menu/example_menu.xml`:

```
<menu xmlns:android="http://schemas.android.com/apk/res/android">
  <item android:id="@+id/item1"
        android:title="@string/item1"
        android:icon="@drawable/group_item1_icon"
        android:showAsAction="ifRoom|withText" />
  <group android:id="@+id/group">
    <item android:id="@+id/group_item1"
          android:onClick="onGroupItemClick"
          android:title="@string/group_item1"
          android:icon="@drawable/group_item1_icon" />
    <item android:id="@+id/group_item2"
          android:onClick="onGroupItemClick"
          android:title="@string/group_item2"
          android:icon="@drawable/group_item2_icon" />
  </group>
  <item android:id="@+id/submenu"
        android:title="@string/submenu_title"
        android:showAsAction="ifRoom|withText" >
    <menu>
      <item android:id="@+id/submenu_item1"
            android:title="@string/submenu_item1" />
    </menu>
  </item>
</menu>
```

The following application code inflates the menu from the `onCreateOptionsMenu(Menu)` callback and also declares the on-click callback for two of the items:

```
public boolean onCreateOptionsMenu(Menu menu) {
    MenuInflater inflater = getMenuInflater();
    inflater.inflate(R.menu.example_menu, menu);
    return true;
}

public void onGroupItemClick(MenuItem item) {
    // One of the group items (using the onClick attribute) was clicked
    // The item parameter passed here indicates which item it is
    // All other menu item clicks are handled by onOptionsItemSelected()
}
```