

public static final class  
**R.styleable**  
extends [Object](#)

Summary: [Constants](#) | [Fields](#) | [Ctors](#) | [Inherited Methods](#) | [\[Expand All\]](#)

[java.lang.Object](#)  
↳ android.R.styleable

## Summary

### Constants

int AbsListView_cacheColorHint	Indicates that this list will always be drawn on top of solid, single-color op-
int AbsListView_choiceMode	Defines the choice behavior for the view.
int AbsListView_drawSelectorOnTop	When set to true, the selector will be drawn over the selected item.
int AbsListView_fastScrollAlwaysVisible	When set to true, the list will always show the fast scroll interface.
int AbsListView_fastScrollEnabled	Enables the fast scroll thumb that can be dragged to quickly scroll through the list.
int AbsListView_listSelector	Drawable used to indicate the currently selected item in the list.
int AbsListView_scrollingCache	When set to true, the list uses a drawing cache during scrolling.
int AbsListView_smoothScrollbar	When set to true, the list will use a more refined calculation method based on the items visible on screen.
int AbsListView_stackFromBottom	Used by ListView and GridView to stack their content from the bottom.
int AbsListView_textFilterEnabled	When set to true, the list will filter results as the user types.
int AbsListView_transcriptMode	Sets the transcript mode for the list.
int AbsSpinner_entries	Reference to an array resource that will populate the Spinner.
int AbsoluteLayout_Layout_layout_x	This symbol is the offset where the <code>layout_x</code> ( <a href="#">/reference/android/R.attr.layout_x</a> ) attribute's value can be found in the <code>AbsoluteLayout_Layout</code> ( <a href="#">/reference/android/R.styleable.html#AbsoluteLayout_Layout</a> ) array.
int AbsoluteLayout_Layout_layout_y	This symbol is the offset where the <code>layout_y</code> ( <a href="#">/reference/android/R.attr.layout_y</a> ) attribute's value can be found in the <code>AbsoluteLayout_Layout</code> ( <a href="#">/reference/android/R.styleable.html#AbsoluteLayout_Layout</a> ) array.
int AccelerateInterpolator_factor	This is the amount of deceleration to add when easing in.
int AccessibilityService_accessibilityEventTypes	The event types this service would like to receive as specified in <code>AccessibilityEventTypes</code> ( <a href="#">/reference/android/view/accessibility/AccessibilityEvent.html</a> ).
int AccessibilityService_accessibilityFeedbackType	The feedback types this service provides as specified in <code>AccessibilityFeedbackType</code> ( <a href="#">/reference/android/accessibilityservice/AccessibilityServiceInfo.html</a> ).
int AccessibilityService_accessibilityFlags	Additional flags as specified in <code>AccessibilityServiceInfo</code> ( <a href="#">/reference/android/accessibilityservice/AccessibilityServiceInfo.html</a> ).
int AccessibilityService_canRetrieveWindowContent	Flag whether the accessibility service wants to be able to retrieve the active window's content.
int AccessibilityService_description	Short description of the accessibility service purpose or behavior.

int AccessibilityService_notificationTimeout	The minimal period in milliseconds between two accessibility events of the to this service.
int AccessibilityService_packageNames	Comma separated package names from which this service would like to route for all packages).
int AccessibilityService_settingsActivity	Component name of an activity that allows the user to modify the settings
int AccountAuthenticator_accountPreferences	A preferences.xml file for authenticator-specific settings.
int AccountAuthenticator_accountType	The account type this authenticator handles.
int AccountAuthenticator_customTokens	Account handles its own token storage and permissions.
int AccountAuthenticator_icon	The icon of the authenticator.
int AccountAuthenticator_label	The user-visible name of the authenticator.
int AccountAuthenticator_smallIcon	Smaller icon of the authenticator.
int ActionBar_LayoutParams_layout_gravity	Standard gravity constant that a child supplies to its parent.
int ActionBar_background	Specifies a background drawable for the action bar.
int ActionBar_backgroundSplit	Specifies a background drawable for the bottom component of a split action bar.
int ActionBar_backgroundStacked	Specifies a background drawable for a second stacked row of the action bar.
int ActionBar_customNavigationLayout	Specifies a layout for custom navigation.
int ActionBar_displayOptions	Options affecting how the action bar is displayed.
int ActionBar_divider	Specifies the drawable used for item dividers.
int ActionBar_height	Specifies a fixed height.
int ActionBar_homeLayout	Specifies a layout to use for the "home" section of the action bar.
int ActionBar_icon	Specifies the drawable used for the application icon.
int ActionBar_ineterminateProgressStyle	Specifies a style resource to use for an indeterminate progress spinner.
int ActionBar_itemPadding	Specifies padding that should be applied to the left and right sides of system bar.
int ActionBar_logo	Specifies the drawable used for the application logo.
int ActionBar_navigationMode	The type of navigation to use.
int ActionBar_progressBarPadding	Specifies the horizontal padding on either end for an embedded progress bar.
int ActionBar_progressBarStyle	Specifies a style resource to use for an embedded progress bar.
int ActionBar_subtitle	Specifies subtitle text used for navigationMode="normal" Must be a string value, using '\\\' to escape characters such as '\\n' or '\\ux' character.

int ActionBar_subtitleTextStyle	Specifies a style to use for subtitle text.
int ActionBar_title	Specifies title text used for navigationMode="normal"
int ActionBar_titleTextStyle	Must be a string value, using '\\;' to escape characters such as '\\n' or '\\ux' character.
int ActionMenuItemView_minWidth	This symbol is the offset where the minWidth (/reference/android/R.attr.html#minWidth) attribute's value can be found in the ActionMenuItemView (/reference/android/R.styleable.html#ActionMenuItemView) array.
int ActionMode_background	Specifies a background for the action mode bar.
int ActionMode_backgroundSplit	Specifies a background for the split action mode bar.
int ActionMode_height	Specifies a fixed height for the action mode bar.
int ActionMode_subtitleTextStyle	Specifies a style to use for subtitle text.
int ActionMode_titleTextStyle	Specifies a style to use for title text.
int AdapterViewAnimator_animateFirstView	Defines whether to animate the current View when the ViewAnimation is finished.
int AdapterViewAnimator_inAnimation	Identifier for the animation to use when a view is shown.
int AdapterViewAnimator_loopViews	Defines whether the animator loops to the first view once it has reached the last view.
int AdapterViewAnimator_outAnimation	Identifier for the animation to use when a view is hidden.
int AdapterViewFlipper_autoStart	When true, automatically start animating
	Must be a boolean value, either "true" or "false".
int AdapterViewFlipper_flipInterval	This symbol is the offset where the flipInterval (/reference/android/R.attr.html#flipInterval) attribute's value can be found in the AdapterViewFlipper (/reference/android/R.styleable.html#AdapterViewFlipper) array.
int AlertDialog_bottomBright	This symbol is the offset where the bottomBright (/reference/android/R.attr.html#bottomBright) attribute's value can be found in the AlertDialog (/reference/android/R.styleable.html#AlertDialog) array.
int AlertDialog_bottomDark	This symbol is the offset where the bottomDark (/reference/android/R.attr.html#bottomDark) attribute's value can be found in the AlertDialog (/reference/android/R.styleable.html#AlertDialog) array.
int AlertDialog_bottomMedium	This symbol is the offset where the bottomMedium (/reference/android/R.attr.html#bottomMedium) attribute's value can be found in the AlertDialog (/reference/android/R.styleable.html#AlertDialog) array.
int AlertDialog_centerBright	This symbol is the offset where the centerBright (/reference/android/R.attr.html#centerBright) attribute's value can be found in the AlertDialog (/reference/android/R.styleable.html#AlertDialog) array.
int AlertDialog_centerDark	This symbol is the offset where the centerDark (/reference/android/R.attr.html#centerDark) attribute's value can be found in the AlertDialog (/reference/android/R.styleable.html#AlertDialog) array.
	This symbol is the offset where the centerMedium (/reference/android/R.attr.html#centerMedium) attribute's value can be found in the AlertDialog (/reference/android/R.styleable.html#AlertDialog) array.

int AlertDialog_centerMedium	( <a href="#">/reference/android/R.attr.html#centerMedium</a> ) attribute's value can be found in the <a href="#">AlertDialog</a> ( <a href="#">/reference/android/R.styleable.html#AlertDialog</a> ) array.
int AlertDialog_fullBright	This symbol is the offset where the <b>fullBright</b> ( <a href="#">/reference/android/R.attr.html#fullBright</a> ) attribute's value can be found in the <a href="#">AlertDialog</a> ( <a href="#">/reference/android/R.styleable.html#AlertDialog</a> ) array.
int AlertDialog_fullDark	This symbol is the offset where the <b>fullDark</b> ( <a href="#">/reference/android/R.attr.html#fullDark</a> ) attribute's value can be found in the <a href="#">AlertDialog</a> ( <a href="#">/reference/android/R.styleable.html#AlertDialog</a> ) array.
int AlertDialog_layout	Supply an identifier for the layout resource to inflate when the ViewStub is forced to do so.
int AlertDialog_topBright	This symbol is the offset where the <b>topBright</b> ( <a href="#">/reference/android/R.attr.html#topBright</a> ) attribute's value can be found in the <a href="#">AlertDialog</a> ( <a href="#">/reference/android/R.styleable.html#AlertDialog</a> ) array.
int AlertDialog_topDark	This symbol is the offset where the <b>topDark</b> ( <a href="#">/reference/android/R.attr.html#topDark</a> ) attribute's value can be found in the <a href="#">AlertDialog</a> ( <a href="#">/reference/android/R.styleable.html#AlertDialog</a> ) array.
int AlphaAnimation_fromAlpha	This symbol is the offset where the <b>fromAlpha</b> ( <a href="#">/reference/android/R.attr.html#fromAlpha</a> ) attribute's value can be found in the <a href="#">AlphaAnimation</a> ( <a href="#">/reference/android/R.styleable.html#AlphaAnimation</a> ) array.
int AlphaAnimation_toAlpha	This symbol is the offset where the <b>toAlpha</b> ( <a href="#">/reference/android/R.attr.html#toAlpha</a> ) attribute's value can be found in the <a href="#">AlphaAnimation</a> ( <a href="#">/reference/android/R.styleable.html#AlphaAnimation</a> ) array.
int AnalogClock_dial	This symbol is the offset where the <b>dial</b> ( <a href="#">/reference/android/R.attr.html#dial</a> ) attribute's value can be found in the <a href="#">AnalogClock</a> ( <a href="#">/reference/android/R.styleable.html#AnalogClock</a> ) array.
int AnalogClock_hand_hour	This symbol is the offset where the <b>hand_hour</b> ( <a href="#">/reference/android/R.attr.html#hand_hour</a> ) attribute's value can be found in the <a href="#">AnalogClock</a> ( <a href="#">/reference/android/R.styleable.html#AnalogClock</a> ) array.
int AnalogClock_hand_minute	This symbol is the offset where the <b>hand_minute</b> ( <a href="#">/reference/android/R.attr.html#hand_minute</a> ) attribute's value can be found in the <a href="#">AnalogClock</a> ( <a href="#">/reference/android/R.styleable.html#AnalogClock</a> ) array.
int AndroidManifestAction_name	The name of an action that is handled, using the Java-style naming convention.
int AndroidManifestActivityAlias_description	Descriptive text for the associated data.
int AndroidManifestActivityAlias_enabled	Specify whether the activity-alias is enabled or not (that is, can be instantiated).
int AndroidManifestActivityAlias_exported	Flag indicating whether the given application component is available to other applications.
int AndroidManifestActivityAlias_icon	A Drawable resource providing a graphical representation of its associated item.
int AndroidManifestActivityAlias_label	A user-legible name for the given item.
int AndroidManifestActivityAlias_logo	A Drawable resource providing an extended graphical logo for its associated item.
int AndroidManifestActivityAlias_name	Required name of the class implementing the activity, deriving from <a href="#">Activity</a> ( <a href="#">/reference/android/app/Activity.html</a> ).
int AndroidManifestActivityAlias_parentActivityName	The name of the logical parent of the activity as it appears in the manifest.

int AndroidManifestActivity_permission	Specify a permission that a client is required to have in order to use the activity.
int AndroidManifestActivity_targetActivity	The name of the activity this alias should launch.
int AndroidManifestActivity_allowTaskReparenting	Specify that an activity can be moved out of a task it is in to the task it has appropriate.
int AndroidManifestActivity_alwaysRetainTaskState	Specify whether an activity's task state should always be maintained by the system, or allowed to reset the task to its initial state in certain situations.
int AndroidManifestActivity_clearTaskOnLaunch	Specify whether an activity's task should be cleared when it is re-launched from the home screen.
int AndroidManifestActivity_configChanges	Specify one or more configuration changes that the activity will handle itself.
int AndroidManifestActivity_description	Descriptive text for the associated data.
int AndroidManifestActivity_enabled	Specify whether the activity is enabled or not (that is, can be instantiated by the system).
int AndroidManifestActivity_excludeFromRecents	Indicates that an Activity should be excluded from the list of recently launched activities.
int AndroidManifestActivity_exported	Flag indicating whether the given application component is available to other applications.
int AndroidManifestActivity_finishOnCloseSystemDialogs	Specify whether an activity should be finished when a "close system window" dialog has been made.
int AndroidManifestActivity_finishOnTaskLaunch	Specify whether an activity should be finished when its task is brought to the foreground by relaunching from the home screen.
int AndroidManifestActivity_hardwareAccelerated	Flag indicating whether the application's rendering should be hardware accelerated if possible.
int AndroidManifestActivity_icon	A Drawable resource providing a graphical representation of its associated item.
int AndroidManifestActivity_immersive	Flag declaring this activity to be 'immersive'; immersive activities should not be shown with other activities or notifications.
int AndroidManifestActivity_label	A user-legible name for the given item.
int AndroidManifestActivity_launchMode	Specify how an activity should be launched.
int AndroidManifestActivity_logo	A Drawable resource providing an extended graphical logo for its associated item.
int AndroidManifestActivity_multiprocess	Specify whether a component is allowed to have multiple instances of itself running in different processes.
int AndroidManifestActivity_name	Required name of the class implementing the activity, deriving from <a href="#">Activity</a> . ( <a href="#">/reference/android/app/Activity.html</a> ).
int AndroidManifestActivity_noHistory	Specify whether an activity should be kept in its history stack.
int AndroidManifestActivity_parentActivityName	The name of the logical parent of the activity as it appears in the manifest.
int AndroidManifestActivity_permission	Specify a permission that a client is required to have in order to use the activity.
int AndroidManifestActivity_process	Specify a specific process that the associated code is to run in.
int AndroidManifestActivity_screenOrientation	Specify the orientation an activity should be run in.

int AndroidManifestActivity_showOnLockScreen	Specify that an Activity should be shown over the lock screen and, in a multi-tasking environment, across all users' windows  Must be a boolean value, either "true" or "false".
int AndroidManifestActivity_singleUser	If set to true, a single instance of this component will run for all users.
int AndroidManifestActivity_stateNotNeeded	Indicates that an Activity does not need to have its freeze state (as returned by onSaveInstanceState(Bundle)) ( <a href="#">/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)</a> ) retained in order to be restarted.
int AndroidManifestActivity_taskAffinity	Specify a task name that activities have an "affinity" to.
int AndroidManifestActivity_theme	The overall theme to use for an activity.
int AndroidManifestActivity_uiOptions	Extra options for an activity's UI.
int AndroidManifestActivity_windowSoftInputMode	Specify the default soft-input mode for the main window of this activity.
int AndroidManifestApplication_allowBackup	Whether to allow the application to participate in the backup and restore process.
int AndroidManifestApplication_allowClearUserData	Option to let applications specify that user data can/cannot be cleared.
int AndroidManifestApplication_allowTaskReparenting	Specify that an activity can be moved out of a task it is in to the task it has appropriate.
int AndroidManifestApplication_backupAgent	The name of the class subclassing BackupAgent to manage backup and restore of the application's data on external storage.
int AndroidManifestApplication_debuggable	Flag indicating whether the application can be debugged, even when running in user mode.
int AndroidManifestApplication_description	Descriptive text for the associated data.
int AndroidManifestApplication_enabled	Specify whether the components in this application are enabled or not (they are instantiated by the system).
int AndroidManifestApplication_hardwareAccelerated	Flag indicating whether the application's rendering should be hardware accelerated if possible.
int AndroidManifestApplication_hasCode	Indicate whether this application contains code.
int AndroidManifestApplication_icon	A Drawable resource providing a graphical representation of its associated application.
int AndroidManifestApplication_killAfterRestore	Whether the application in question should be terminated after its settings are restored during a full-system restore operation.
int AndroidManifestApplication_label	A user-legible name for the given item.
int AndroidManifestApplication_largeHeap	Request that your application's processes be created with a large Dalvik heap.
int AndroidManifestApplication_logo	A Drawable resource providing an extended graphical logo for its associated application.
int AndroidManifestApplication_manageSpaceActivity	Name of activity to be launched for managing the application's space on the device.
int AndroidManifestApplication_name	An optional name of a class implementing the overall Application ( <a href="#">/reference/android/app/Application.html</a> ) for this package.
int AndroidManifestApplication_permission	Specify a permission that a client is required to have in order to use the application.

int AndroidManifestApplication_persistent	Flag to control special persistent mode of an application.
int AndroidManifestApplication_process	Specify a specific process that the associated code is to run in.
int AndroidManifestApplication_restoreAnyVersion	Indicate that the application is prepared to attempt a restore of any backup; the backup is apparently from a newer version of the application than is currently on the device.
int AndroidManifestApplication_restoreNeedsApplication	<p><i>This constant is deprecated. This attribute is not used by the Android operating system.</i></p> <p><i>Must be a boolean value, either "true" or "false".</i></p> <p><i>This may also be a reference to a resource (in the form "@[package:]type:[name]" or an attribute (in the form "?[package:]type:[name]" containing a value of this symbol.</i></p> <p><i>This corresponds to the global attribute resource symbol <code>restoreNeedsApplication</code> (see <a href="#">/reference/android/R.attr.html#restoreNeedsApplication</a>).</i></p>
int AndroidManifestApplication_supportsRtl	Declare that your application will be able to deal with RTL (right to left) layout.
int AndroidManifestApplication_taskAffinity	Specify a task name that activities have an "affinity" to.
int AndroidManifestApplication_testOnly	Option to indicate this application is only for testing purposes.
int AndroidManifestApplication_theme	The overall theme to use for an activity.
int AndroidManifestApplication_uiOptions	Extra options for an activity's UI.
int AndroidManifestApplication_vmSafeMode	Flag indicating whether the application requests the VM to operate in the safe mode.
int AndroidManifestCategory_name	The name of category that is handled, using the Java-style naming convention.
int AndroidManifestCompatibleScreensScreen_screenDensity	Specifies a compatible screen density, as per the device configuration screen density.
int AndroidManifestCompatibleScreensScreen_screenSize	Specifies a compatible screen size, as per the device configuration screen size.
int AndroidManifestData_host	Specify a URI authority host that is handled, as per <code>IntentFilter.addDataAuthority(java.lang.String)</code> .
int AndroidManifestData_mimeType	Specify a MIME type that is handled, as per <code>IntentFilter.addDataType(java.lang.String)</code> .
int AndroidManifestData_path	Specify a URI path that must exactly match, as per <code>IntentFilter.addDataPath(java.lang.String)</code> , with <code>PATTERN_LITERAL</code> ( <a href="#">/reference/android/os/PatternMatcher.html#PATTERN_LITERAL</a> ).
int AndroidManifestData_pathPattern	Specify a URI path that matches a simple pattern, as per <code>IntentFilter.addDataAuthority()</code> , ( <a href="#">/reference/android/content/IntentFilter.html#addDataPath(java.lang.String)</a> ) with <code>PATTERN_SIMPLE_GLOB</code> ( <a href="#">/reference/android/os/PatternMatcher.html#PATTERN_SIMPLE_GLOB</a> ).
int AndroidManifestData_pathPrefix	Specify a URI path that must be a prefix to match, as per <code>IntentFilter.addDataAuthority()</code> , ( <a href="#">/reference/android/content/IntentFilter.html#addDataPath(java.lang.String)</a> ) with <code>PATTERN_PREFIX</code> ( <a href="#">/reference/android/os/PatternMatcher.html#PATTERN_PREFIX</a> ).
int AndroidManifestData_port	Specify a URI authority port that is handled, as per <code>IntentFilter.addDataAuthority(java.lang.String)</code> .

int AndroidManifestData_scheme	Specify a URI scheme that is handled, as per <code>IntentFilter.addDataScheme</code> (/reference/android/content/IntentFilter.html#addDataScheme(java.lang.String)).
int AndroidManifestGrantUriPermission_path	Specify a URI path that must exactly match, as per <code>PatternMatcher</code> (/reference/android/os/PatternMatcher.html) with <code>PATTERN_LITERAL</code> (/reference/android/os/PatternMatcher.html#PATTERN_LITERAL).
int AndroidManifestGrantUriPermission_pathPattern	Specify a URI path that matches a simple pattern, as per <code>PatternMatcher</code> (/reference/android/os/PatternMatcher.html) with <code>PATTERN_SIMPLE_GLOB</code> (/reference/android/os/PatternMatcher.html#PATTERN_SIMPLE_GLOB).
int AndroidManifestGrantUriPermission_pathPrefix	Specify a URI path that must be a prefix to match, as per <code>PatternMatcher</code> (/reference/android/os/PatternMatcher.html) with <code>PATTERN_PREFIX</code> (/reference/android/os/PatternMatcher.html#PATTERN_PREFIX).
int AndroidManifestInstrumentation_functionalTest	Flag indicating that an Instrumentation class should be run as a functional test.
int AndroidManifestInstrumentation_handleProfiling	Flag indicating that an Instrumentation class wants to take care of starting itself, rather than relying on the default behavior of profiling the complete application.
int AndroidManifestInstrumentation_icon	A Drawable resource providing a graphical representation of its associated instrumentation.
int AndroidManifestInstrumentation_label	A user-legible name for the given item.
int AndroidManifestInstrumentation_logo	A Drawable resource providing an extended graphical logo for its associated instrumentation.
int AndroidManifestInstrumentation_name	Required name of the class implementing the instrumentation, deriving from <code>Instrumentation</code> (/reference/android/app/Instrumentation.html).
int AndroidManifestInstrumentation_targetPackage	The name of the application package that an <code>Instrumentation</code> object will run in.
int AndroidManifestIntentFilter_icon	A Drawable resource providing a graphical representation of its associated intent filter.
int AndroidManifestIntentFilter_label	A user-legible name for the given item.
int AndroidManifestIntentFilter_logo	A Drawable resource providing an extended graphical logo for its associated intent filter.
int AndroidManifestIntentFilter_priority	Specify the relative importance or ability in handling a particular Intent.
int AndroidManifestMetaData_name	A unique name for the given item.
int AndroidManifestMetaData_resource	Resource identifier to assign to this piece of named meta-data.
int AndroidManifestMetaData_value	Concrete value to assign to this piece of named meta-data.
int AndroidManifestOriginalPackage_name	A unique name for the given item.
int AndroidManifestPackageVerifier_name	Specifies the Java-style package name that defines this package verifier.
int AndroidManifestPackageVerifier_publicKey	The Base64 encoded public key of the package verifier's signature.
int AndroidManifestPathPermission_path	Specify a URI path that must exactly match, as per <code>PatternMatcher</code> (/reference/android/os/PatternMatcher.html) with <code>PATTERN_LITERAL</code> (/reference/android/os/PatternMatcher.html#PATTERN_LITERAL).
int AndroidManifestPathPermission_pathPattern	Specify a URI path that matches a simple pattern, as per <code>PatternMatcher</code> (/reference/android/os/PatternMatcher.html) with <code>PATTERN_SIMPLE_GLOB</code> (/reference/android/os/PatternMatcher.html#PATTERN_SIMPLE_GLOB).

int AndroidManifestPathPermission_pathPrefix	Specify a URI path that must be a prefix to match, as per <a href="#">PatternMatcher</a> (/reference/android/os/PatternMatcher.html) with <b>PATTERN_PREFIX</b> (/reference/android/os/PatternMatcher.html#PATTERN_PREFIX).
int AndroidManifestPathPermission_permission	Specify a permission that a client is required to have in order to use the associated provider.
int AndroidManifestPathPermission_readPermission	A specific permission (/reference/android/R.attr.html#permission) needed to access to a ContentProvider (/reference/android/content/ContentProvider.html#readPermission).
int AndroidManifestPathPermission_writePermission	A specific permission (/reference/android/R.attr.html#permission) needed to write to a ContentProvider (/reference/android/content/ContentProvider.html#writePermission).
int AndroidManifestPermissionGroup_description	Descriptive text for the associated data.
int AndroidManifestPermissionGroup_icon	A Drawable resource providing a graphical representation of its associated group.
int AndroidManifestPermissionGroup_label	A user-legible name for the given item.
int AndroidManifestPermissionGroup_logo	A Drawable resource providing an extended graphical logo for its associated group.
int AndroidManifestPermissionGroup_name	Required public name of the permission group, permissions will use to specify are in.
int AndroidManifestPermissionGroup_permissionGroupFlags	Flags indicating more context for a permission group.
int AndroidManifestPermissionGroup_priority	Specify the relative importance or ability in handling a particular Intent.
int AndroidManifestPermissionTree_icon	A Drawable resource providing a graphical representation of its associated tree.
int AndroidManifestPermissionTree_label	A user-legible name for the given item.
int AndroidManifestPermissionTree_logo	A Drawable resource providing an extended graphical logo for its associated tree.
int AndroidManifestPermissionTree_name	Required public name of the permission tree, which is the base name of all its children.
int AndroidManifestPermission_description	Descriptive text for the associated data.
int AndroidManifestPermission_icon	A Drawable resource providing a graphical representation of its associated permission.
int AndroidManifestPermission_label	A user-legible name for the given item.
int AndroidManifestPermission_logo	A Drawable resource providing an extended graphical logo for its associated permission.
int AndroidManifestPermission_name	Required public name of the permission, which other components and packages referring to this permission.
int AndroidManifestPermission_permissionFlags	Flags indicating more context for a permission.
int AndroidManifestPermission_permissionGroup	Specified the name of a group that this permission is associated with.
int AndroidManifestPermission_protectionLevel	Characterizes the potential risk implied in a permission and indicates the way the system should follow when determining whether to grant the permission to requesting it.
int AndroidManifestProtectedBroadcast_name	A unique name for the given item.

int AndroidManifestProviderAuthorities	Specify the authorities under which this content provider can be found.
int AndroidManifestProviderDescription	Descriptive text for the associated data.
int AndroidManifestProviderEnabled	Specify whether this provider is enabled or not (that is, can be instantiated).
int AndroidManifestProviderExported	Flag indicating whether the given application component is available to other components.
int AndroidManifestProviderGrantUriPermissions	If true, the <code>Context.grantUriPermission</code> ( <a href="#">/reference/android/content/Context.html#grantUriPermission(java.lang.String, android.net.Uri, int)</a> ) or corresponding Intent flags can be used to allow specific URIs in the content provider, even if they do not have an explicit read permission.
int AndroidManifestProviderIcon	A Drawable resource providing a graphical representation of its associated item.
int AndroidManifestProviderInitOrder	Specify the order in which content providers hosted by a process are installed when the process is created.
int AndroidManifestProviderLabel	A user-legible name for the given item.
int AndroidManifestProviderLogo	A Drawable resource providing an extended graphical logo for its associated item.
int AndroidManifestProviderMultiprocess	Specify whether a component is allowed to have multiple instances of itself running in different processes.
int AndroidManifestProviderName	Required name of the class implementing the provider, deriving from <code>ContentProvider</code> ( <a href="#">/reference/android/content/ContentProvider.html</a> ).
int AndroidManifestProviderPermission	Specify a permission that a client is required to have in order to use the associated provider.
int AndroidManifestProviderProcess	Specify a specific process that the associated code is to run in.
int AndroidManifestProviderReadPermission	A specific permission ( <a href="#">/reference/android/R.attr.html#permission</a> ) needed to access to a <code>ContentProvider</code> ( <a href="#">/reference/android/content/ContentProvider.html</a> ).
int AndroidManifestProviderSingleUser	If set to true, a single instance of this component will run for all users.
int AndroidManifestProviderSyncable	Flag indicating whether this content provider would like to participate in distributed synchronization.
int AndroidManifestProviderWritePermission	A specific permission ( <a href="#">/reference/android/R.attr.html#permission</a> ) needed to write to a <code>ContentProvider</code> ( <a href="#">/reference/android/content/ContentProvider.html</a> ).
int AndroidManifestReceiverDescription	Descriptive text for the associated data.
int AndroidManifestReceiverEnabled	Specify whether the receiver is enabled or not (that is, can be instantiated).
int AndroidManifestReceiverExported	Flag indicating whether the given application component is available to other components.
int AndroidManifestReceiverIcon	A Drawable resource providing a graphical representation of its associated item.
int AndroidManifestReceiverLabel	A user-legible name for the given item.
int AndroidManifestReceiverLogo	A Drawable resource providing an extended graphical logo for its associated item.
int AndroidManifestReceiverName	Required name of the class implementing the receiver, deriving from <code>BroadcastReceiver</code> ( <a href="#">/reference/android/content/BroadcastReceiver.html</a> ).

int AndroidManifestReceiver_permission	Specify a permission that a client is required to have in order to use the associated code.
int AndroidManifestReceiver_process	Specify a specific process that the associated code is to run in.
int AndroidManifestReceiver_singleUser	If set to true, a single instance of this component will run for all users.
int AndroidManifestService_description	Descriptive text for the associated data.
int AndroidManifestService_enabled	Specify whether the service is enabled or not (that is, can be instantiated by clients).
int AndroidManifestService_exported	Flag indicating whether the given application component is available to other components.
int AndroidManifestService_icon	A Drawable resource providing a graphical representation of its associated component.
int AndroidManifestService_isolatedProcess	If set to true, this service will run under a special process that is isolated from the system.
int AndroidManifestService_label	A user-legible name for the given item.
int AndroidManifestService_logo	A Drawable resource providing an extended graphical logo for its associated component.
int AndroidManifestService_name	Required name of the class implementing the service, deriving from Service ( <a href="#">/reference/android/app/Service.html</a> ).
int AndroidManifestService_permission	Specify a permission that a client is required to have in order to use the associated code.
int AndroidManifestService_process	Specify a specific process that the associated code is to run in.
int AndroidManifestService_singleUser	If set to true, a single instance of this component will run for all users.
int AndroidManifestService_stopWithTask	If set to true, this service will be automatically stopped when the user removes an activity owned by the application.
int AndroidManifestSupportsScreens_anyDensity	Indicates whether the application can accommodate any screen density.
int AndroidManifestSupportsScreens_compatibleWidthLimitDp	Starting with HONEYCOMB_MR2 ( <a href="#">/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB_MR2</a> ), this specifies the largest screens an application is compatible with.
int AndroidManifestSupportsScreens_largeScreens	Indicates whether the application supports larger screen form-factors.
int AndroidManifestSupportsScreens_largestWidthLimitDp	Starting with HONEYCOMB_MR2 ( <a href="#">/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB_MR2</a> ), this specifies the screens an application is compatible with.
int AndroidManifestSupportsScreens_normalScreens	Indicates whether an application supports the normal screen form-factors.
int AndroidManifestSupportsScreens_requiresSmallestWidthDp	Starting with HONEYCOMB_MR2 ( <a href="#">/reference/android/os/Build.VERSION_CODES.html#HONEYCOMB_MR2</a> ), this specifies the minimum screen size an application is compatible with.
int AndroidManifestSupportsScreens_resizeable	Indicates whether the application can resize itself to newer screen sizes.
int AndroidManifestSupportsScreens_smallScreens	Indicates whether the application supports smaller screen form-factors.
int AndroidManifestSupportsScreens_xlargeScreens	Indicates whether the application supports extra large screen form-factors.
	Application's requirement for five way navigation

int AndroidManifestUsesConfiguration_reqFiveWayNav	Must be a boolean value, either "true" or "false".
int AndroidManifestUsesConfiguration_reqHardKeyboard	Application's requirement for a hard keyboard Must be a boolean value, either "true" or "false".
int AndroidManifestUsesConfiguration_reqKeyboardType	The input method preferred by an application.
int AndroidManifestUsesConfiguration_reqNavigation	The navigation device preferred by an application.
int AndroidManifestUsesConfiguration_reqTouchScreen	The type of touch screen used by an application.
int AndroidManifestUsesFeature_glEsVersion	The GLES driver version number needed by an application.
int AndroidManifestUsesFeature_name	The name of the feature that is being used.
int AndroidManifestUsesFeature_required	Specify whether this feature is required for the application.
int AndroidManifestUsesLibrary_name	Required name of the library you use.
int AndroidManifestUsesLibrary_required	Specify whether this library is required for the application.
int AndroidManifestUsesPermission_name	Required name of the permission you use, as published with the corresponding <code>&lt;permission&gt;</code> tag; often this is one of the standard system permissions ( <a href="#">reference/android/Manifest.permission.html</a> ).
int AndroidManifestUsesSdk_maxSdkVersion	This is the maximum SDK version number that an application works on.
int AndroidManifestUsesSdk_minSdkVersion	This is the minimum SDK version number that the application requires.
int AndroidManifestUsesSdk_targetSdkVersion	This is the SDK version number that the application is targeting.
int AndroidManifest_installLocation	The default install location defined by an application.
int AndroidManifest_sharedUserId	Specify the name of a user ID that will be shared between multiple packages.
int AndroidManifest_sharedUserLabel	Specify a label for the shared user UID of this package.
int AndroidManifest_versionCode	Internal version code.
int AndroidManifest_versionName	The text shown to the user to indicate the version they have.
int AnimatedRotateDrawable_drawable	Reference to a drawable resource to use for the frame.
int AnimatedRotateDrawable_pivotX	This symbol is the offset where the <code>pivotX</code> attribute's value can be found in the <code>AnimatedRotateDrawable</code> array.
int AnimatedRotateDrawable_pivotY	This symbol is the offset where the <code>pivotY</code> attribute's value can be found in the <code>AnimatedRotateDrawable</code> array.
int AnimatedRotateDrawable_visible	Provides initial visibility state of the drawable; the default value is false.
int AnimationDrawableItem_drawable	Reference to a drawable resource to use for the frame.
int AnimationDrawableItem_duration	Amount of time (in milliseconds) to display this frame.

int AnimationDrawable_oneshot	If true, the animation will only run a single time and then stop.
int AnimationDrawable_variablePadding	If true, allows the drawable's padding to change based on the current state.
int AnimationDrawable_visible	Provides initial visibility state of the drawable; the default value is false.
int AnimationSet_duration	Amount of time (in milliseconds) to display this frame.
int AnimationSet_fillAfter	When set to true, the animation transformation is applied after the animation has started.
int AnimationSet_fillBefore	When set to true or when fillEnabled is not set to true, the animation transformation is applied before the animation has started.
int AnimationSet_repeatMode	Defines the animation behavior when it reaches the end and the repeat count is finite or infinite.
int AnimationSet_shareInterpolator	This symbol is the offset where the <code>shareInterpolator</code> ( <a href="#">/reference/android/R.attr.html#shareInterpolator</a> ) attribute's value can be used in an <code>AnimationSet</code> ( <a href="#">/reference/android/R.styleable.html#AnimationSet</a> ) animation.
int AnimationSet_startOffset	Delay in milliseconds before the animation runs, once start time is reached.
int Animation_background	Special background behind animation.
int Animation_detachWallpaper	Special option for window animations: if this window is on top of a wallpaper, it will detach from the wallpaper with it.
int Animation_duration	Amount of time (in milliseconds) for the animation to run.
int Animation_fillAfter	When set to true, the animation transformation is applied after the animation has started.
int Animation_fillBefore	When set to true or when fillEnabled is not set to true, the animation transformation is applied before the animation has started.
int Animation_fillEnabled	When set to true, the value of fillBefore is taken into account.
int Animation_interpolator	Defines the interpolator used to smooth the animation movement in time.
int Animation_repeatCount	Defines how many times the animation should repeat.
int Animation_repeatMode	Defines the animation behavior when it reaches the end and the repeat count is finite or infinite.
int Animation_startOffset	Delay in milliseconds before the animation runs, once start time is reached.
int Animation_zAdjustment	Allows for an adjustment of the Z ordering of the content being animated for the animation.
int AnimatorSet_ordering	Name of the property being animated.
int Animator_duration	Amount of time (in milliseconds) for the animation to run.
int Animator_interpolator	Defines the interpolator used to smooth the animation movement in time.
int Animator_repeatCount	Defines how many times the animation should repeat.
int Animator_repeatMode	Defines the animation behavior when it reaches the end and the repeat count is finite or infinite.

int Animator_startOffset	Delay in milliseconds before the animation runs, once start time is reached.
int Animator_valueFrom	Value the animation starts from.
int Animator_valueTo	Value the animation animates to.
int AnimatorValueType	The type of valueFrom and valueTo.
int AnticipateInterpolator_tension	This is the amount of tension.
int AnticipateOvershootInterpolator_extraTension	This is the amount by which to multiply the tension.
int AnticipateOvershootInterpolator_tension	This is the amount of tension.
int AppWidgetProviderInfo_autoAdvanceViewId	The view id of the AppWidget subview which should be auto-advanced.
int AppWidgetProviderInfo_configure	A class name in the AppWidget's package to be launched to configure.
int AppWidgetProviderInfo_initialKeyguardLayout	A resource id of a layout.
int AppWidgetProviderInfo_initialLayout	A resource id of a layout.
int AppWidgetProviderInfo_minHeight	Minimum height of the AppWidget.
int AppWidgetProviderInfo_minResizeHeight	Minimum height that the AppWidget can be resized to.
int AppWidgetProviderInfo_minResizeWidth	Minimum width that the AppWidget can be resized to.
int AppWidgetProviderInfo_minWidth	Minimum width of the AppWidget.
int AppWidgetProviderInfo_previewImage	A preview of what the AppWidget will look like after it's configured.
int AppWidgetProviderInfo_resizeMode	Optional parameter which indicates if and how this widget can be resized.
int AppWidgetProviderInfo_updatePeriodMillis	Update period in milliseconds, or 0 if the AppWidget will update itself.
int AppWidgetProviderInfo_widgetCategory	Optional parameter which indicates where this widget can be shown, ie.
int AutoCompleteTextView_completionHint	Defines the hint displayed in the drop down menu.
int AutoCompleteTextView_completionHintView	Defines the hint view displayed in the drop down menu.
int AutoCompleteTextView_completionThreshold	Defines the number of characters that the user must type before completion is displayed in a drop down menu.
int AutoCompleteTextView_dropDownAnchor	View to anchor the auto-complete dropdown to.
int AutoCompleteTextView_dropDownHeight	Specifies the basic height of the dropdown.
int AutoCompleteTextView_dropDownHorizontalOffset	Amount of pixels by which the drop down should be offset horizontally.
int AutoCompleteTextView_dropDownSelector	Selector in a drop down list.
int AutoCompleteTextView_dropDownVerticalOffset	Amount of pixels by which the drop down should be offset vertically.
int AutoCompleteTextView_dropDownWidth	Specifies the basic width of the dropdown.

int AutoCompleteTextView_inputType	The type of data being placed in a text field, used to help an input method (IME) know what kind of text to present. See <a href="#">InputType</a> .
int BitmapDrawable_antialias	Enables or disables antialiasing.
int BitmapDrawable_dither	Enables or disables dithering of the bitmap if the bitmap does not have the configuration as the screen (for instance: a ARGB 8888 bitmap with an RGB 565 screen).
int BitmapDrawable_filter	Enables or disables bitmap filtering.
int BitmapDrawable_gravity	Defines the gravity for the bitmap.
int BitmapDrawable_src	Identifier of the bitmap file.
int BitmapDrawable_tileMode	Defines the tile mode.
int CalendarView_dateTextAppearance	The text appearance for the calendar dates.
int CalendarView_firstDayOfWeek	The first day of week according to <a href="#">Calendar</a> ( <a href="#">/reference/java/util/Calendar.html#setFirstDayOfWeek(int)</a> ).
int CalendarView_focusedMonthDateColor	The color for the dates of the focused month.
int CalendarView_maxDate	The minimal date shown by this calendar view in mm/dd/yyyy format.
int CalendarView_minDate	The maximal date shown by this calendar view in mm/dd/yyyy format.
int CalendarView_selectedDateVerticalBar	Drawable for the vertical bar shown at the beginning and at the end of the selected date.
int CalendarView_selectedWeekBackgroundColor	The background color for the selected week.
int CalendarView_showWeekNumber	Whether do show week numbers.
int CalendarView_shownWeekCount	The number of weeks to be shown.
int CalendarView_unfocusedMonthDateColor	The color for the dates of an unfocused month.
int CalendarView_weekDayTextAppearance	The text appearance for the week day abbreviation of the calendar header.
int CalendarView_weekNumberColor	The color for the week numbers.
int CalendarView_weekSeparatorLineColor	The color for the separator line between weeks.
int CheckBoxPreference_disableDependentsState	The state (true for on, or false for off) that causes dependents to be disabled.
int CheckBoxPreference_summaryOff	The summary for the Preference in a PreferenceActivity screen when the CheckBox is unchecked.
int CheckBoxPreference_summaryOn	The summary for the Preference in a PreferenceActivity screen when the CheckBox is checked.
int CheckedTextView_checkMark	Drawable used for the check mark graphic.
int CheckedTextView_checked	Indicates the initial checked state of this text.
int Chronometer_format	Format string: if specified, the Chronometer will display this string, with the current timer value in "MM:SS" or "H:MM:SS" form.

int ClipDrawable_clipOrientation	The orientation for the clip.
int ClipDrawable_drawable	Reference to a drawable resource to draw with the specified scale.
int ClipDrawable_gravity	Specifies where to clip within the drawable.
int ColorDrawable_color	The color to use.
int CompoundButton_button	Drawable used for the button graphic (e.g.
int CompoundButton_checked	Indicates the initial checked state of this button.
int ContactsDataKind_allContactsName	Resource representing the term "All Contacts" (e.g.
int ContactsDataKind_detailColumn	Column in data table that contains details for this data.
int ContactsDataKind_detailSocialSummary	Flag indicating that detail should be built from SocialProvider.
int ContactsDataKind_icon	Icon used to represent data of this kind.
int ContactsDataKind_mimeType	Mime-type handled by this mapping.
int ContactsDataKind_summaryColumn	Column in data table that summarizes this data.
int CycleInterpolator_cycles	This symbol is the offset where the <code>cycles</code> ( <a href="#">/reference/android/R.attr.cycles</a> ) attribute's value can be found in the <code>CycleInterpolator</code> ( <a href="#">/reference/android/R.styleable.html#CycleInterpolator</a> ) array.
int DatePicker_calendarViewShown	Whether the calendar view is shown.
int DatePicker_endYear	The last year (inclusive), for example "2010".
int DatePicker_maxDate	The maximal date shown by this calendar view in mm/dd/yyyy format.
int DatePicker_minDate	The minimal date shown by this calendar view in mm/dd/yyyy format.
int DatePicker_spinnersShown	Whether the spinners are shown.
int DatePicker_startYear	The first year (inclusive), for example "1940".
int DecelerateInterpolator_factor	This is the amount of acceleration to add when easing out.
int DeviceAdmin_visible	Control whether the admin is visible to the user, even when it is not enabled.
int DialogPreference_dialogIcon	The icon for the dialog.
int DialogPreference_dialogLayout	A layout to be used as the content View for the dialog.
int DialogPreference_dialogMessage	The message in the dialog.
int DialogPreference_dialogTitle	The title in the dialog.
int DialogPreference_negativeButtonText	The negative button text for the dialog.
int DialogPreference_positiveButtonText	The positive button text for the dialog.
int DrawableCorners_bottomLeftRadius	Radius of the bottom left corner.

int DrawableCorners_bottomRightRadius	Radius of the bottom right corner.
int DrawableCorners_radius	Defines the radius of the four corners.
int DrawableCorners_topLeftRadius	Radius of the top left corner.
int DrawableCorners_topRightRadius	Radius of the top right corner.
int DrawableStates_state_accelerated	State value for StateListDrawable (/reference/android/graphics/drawable/StateListDrawable.html), indicating the Drawable is in a view that is hardware accelerated.
int DrawableStates_state_activated	State value for StateListDrawable (/reference/android/graphics/drawable/StateListDrawable.html), set when the parent has been "activated" meaning the user has currently marked it as being active.
int DrawableStates_state_active	State value for StateListDrawable (/reference/android/graphics/drawable/StateListDrawable.html).
int DrawableStates_state_checkable	State identifier indicating that the object <i>may</i> display a check mark.
int DrawableStates_state_checked	State identifier indicating that the object is currently checked.
int DrawableStates_state_drag_can_accept	State for StateListDrawable (/reference/android/graphics/drawable/StateListDrawable.html) indicating the Drawable is in a view that is capable of accepting a drop of the content currently manipulated in a drag-and-drop operation.
int DrawableStates_state_drag_hovered	State for StateListDrawable (/reference/android/graphics/drawable/StateListDrawable.html) indicating the operation (for which the Drawable's view is a valid recipient) is currently performed on the Drawable.
int DrawableStates_state_enabled	State value for StateListDrawable (/reference/android/graphics/drawable/StateListDrawable.html), set when enabled.
int DrawableStates_state_first	State value for StateListDrawable (/reference/android/graphics/drawable/StateListDrawable.html).
int DrawableStates_state_focused	State value for StateListDrawable (/reference/android/graphics/drawable/StateListDrawable.html), set when input focus.
int DrawableStates_state_hovered	State value for StateListDrawable (/reference/android/graphics/drawable/StateListDrawable.html), set when hovering over the view.
int DrawableStates_state_last	State value for StateListDrawable (/reference/android/graphics/drawable/StateListDrawable.html).
int DrawableStates_state_middle	State value for StateListDrawable (/reference/android/graphics/drawable/StateListDrawable.html).
int DrawableStates_state_pressed	State value for StateListDrawable (/reference/android/graphics/drawable/StateListDrawable.html), set when pressing down in a view.
int DrawableStates_state_selected	State value for StateListDrawable (/reference/android/graphics/drawable/StateListDrawable.html), set when one of its parents) is currently selected.

int DrawableStates_state_single	State value for StateListDrawable (/reference/android/graphics/drawable/StateListDrawable.html).
int DrawableStates_state_window_focused	State value for StateListDrawable (/reference/android/graphics/drawable/StateListDrawable.html), set when window has input focus.
int Drawable_visible	Provides initial visibility state of the drawable; the default value is false.
int Dream_settingsActivity	Component name of an activity that allows the user to modify the settings
int ExpandableListChildIndicatorState_state_last	State identifier indicating the child is the last child within its group.
int ExpandableListGroupIndicatorState_state_empty	State identifier indicating the group is empty (has no children).
int ExpandableListGroupIndicatorState_state_expanded	State identifier indicating the group is expanded.
int ExpandableListView_childDivider	Drawable or color that is used as a divider for children.
int ExpandableListView_childIndicator	Indicator shown beside the child View.
int ExpandableListView_childIndicatorLeft	The left bound for a child's indicator.
int ExpandableListView_childIndicatorRight	The right bound for a child's indicator.
int ExpandableListView_groupIndicator	Indicator shown beside the group View.
int ExpandableListView_indicatorLeft	The left bound for an item's indicator.
int ExpandableListView_indicatorRight	The right bound for an item's indicator.
int Extra_name	Required name of the extra data.
int Extra_value	Concrete value to put for this named extra data.
int FragmentAnimation_fragmentCloseEnterAnimation	This symbol is the offset where the fragmentCloseEnterAnimation (/reference/android/R.attr.html#fragmentCloseEnterAnimation) attribute found in the FragmentAnimation (/reference/android/R.styleable.html#FragmentAnimation) array.
int FragmentAnimation_fragmentCloseExitAnimation	This symbol is the offset where the fragmentCloseExitAnimation (/reference/android/R.attr.html#fragmentCloseExitAnimation) attribute found in the FragmentAnimation (/reference/android/R.styleable.html#FragmentAnimation) array.
int FragmentAnimation_fragmentFadeEnterAnimation	This symbol is the offset where the fragmentFadeEnterAnimation (/reference/android/R.attr.html#fragmentFadeEnterAnimation) attribute found in the FragmentAnimation (/reference/android/R.styleable.html#FragmentAnimation) array.
int FragmentAnimation_fragmentFadeExitAnimation	This symbol is the offset where the fragmentFadeExitAnimation (/reference/android/R.attr.html#fragmentFadeExitAnimation) attribute found in the FragmentAnimation (/reference/android/R.styleable.html#FragmentAnimation) array.
int FragmentAnimation_fragmentOpenEnterAnimation	This symbol is the offset where the fragmentOpenEnterAnimation (/reference/android/R.attr.html#fragmentOpenEnterAnimation) attribute found in the FragmentAnimation (/reference/android/R.styleable.html#FragmentAnimation) array.

<code>int FragmentAnimation_fragmentOpenExitAnimation</code>	This symbol is the offset where the <code>fragmentOpenExitAnimation</code> ( <code>/reference/android/R.attr.html#fragmentOpenExitAnimation</code> ) attribute found in the <code>FragmentAnimation</code> ( <code>/reference/android/R.styleable.html#FragmentAnimation</code> ) array.
<code>int FragmentBreadCrumbs_gravity</code>	Specifies how an object should position its content, on both the X and Y axis, relative to its bounds.
<code>int Fragment_id</code>	Supply an identifier name for the top-level view, to later retrieve it with <code>View.findViewById()</code> ( <code>/reference/android/view/View.html#findViewById</code> ) or <code>Activity.findViewById()</code> ( <code>/reference/android/app/Activity.html#findViewById</code> ).
<code>int Fragment_name</code>	Supply the name of the fragment class to instantiate.
<code>int Fragment_tag</code>	Supply a tag for the top-level view containing a String, to be retrieved later with <code>View.getTag()</code> ( <code>/reference/android/view/View.html#getTag</code> ) or <code>View.findViewWithTag()</code> ( <code>/reference/android/view/View.html#findViewWithTag</code> ( <code>java.lang.Object</code> )).
<code>int FrameLayout_Layout_layout_gravity</code>	Standard gravity constant that a child supplies to its parent.
<code>int FrameLayout_foreground</code>	Defines the drawable to draw over the content.
<code>int FrameLayout_foregroundGravity</code>	Defines the gravity to apply to the foreground drawable.
<code>int FrameLayout_measureAllChildren</code>	Determines whether to measure all children or just those in the VISIBLE or GONE state when measuring.
<code>int Gallery_animationDuration</code>	Sets how long a transition animation should run (in milliseconds) when launching a transition.
<code>int Gallery_gravity</code>	Specifies how an object should position its content, on both the X and Y axis, relative to its bounds.
<code>int Gallery_spacing</code>	This symbol is the offset where the <code>spacing</code> ( <code>/reference/android/R.attr.html#spacing</code> ) attribute's value can be found in the <code>Gallery</code> ( <code>/reference/android/R.styleable.html#Gallery</code> ) array.
<code>int Gallery_unselectedAlpha</code>	Sets the alpha on the items that are not selected.
<code>int GestureOverlayView_eventsInterceptionEnabled</code>	Defines whether the overlay should intercept the motion events when a gesture is detected.
<code>int GestureOverlayView_fadeDuration</code>	Duration, in milliseconds, of the fade out effect after the user is done drawing a gesture.
<code>int GestureOverlayView_fadeEnabled</code>	Defines whether the gesture will automatically fade out after being recognized.
<code>int GestureOverlayView_fadeOffset</code>	Time, in milliseconds, to wait before the gesture fades out after the user is done drawing a gesture.
<code>int GestureOverlayView_gestureColor</code>	Color used to draw a gesture.
<code>int GestureOverlayView_gestureStrokeAngleThreshold</code>	Minimum curve angle a stroke must contain before it is recognized as a gesture.
<code>int GestureOverlayView_gestureStrokeLengthThreshold</code>	Minimum length of a stroke before it is recognized as a gesture.
<code>int GestureOverlayView_gestureStrokeSquarenessThreshold</code>	Squareness threshold of a stroke before it is recognized as a gesture.
<code>int GestureOverlayView_gestureStrokeType</code>	Defines the type of strokes that define a gesture.

int GestureOverlayView_gestureStrokeWidth	Width of the stroke used to draw the gesture.
int GestureOverlayView_orientation	Indicates whether horizontal (when the orientation is vertical) or vertical ('horizontal) strokes automatically define a gesture.
int GestureOverlayView_uncertainGestureColor	Color used to draw the user's strokes until we are sure it's a gesture.
int GlowPadView_directionDescriptions	Reference to an array resource that be used to announce the directions with circle.
int GlowPadView_gravity	Specifies how an object should position its content, on both the X and Y axis, relative to its bounds.
int GlowPadView_innerRadius	Inner radius of glow area.
int GlowPadView_targetDescriptions	Reference to an array resource that be used as description for the targets associated with the glow area.
int GradientDrawableGradient_angle	Angle of the gradient.
int GradientDrawableGradient_centerColor	Optional center color.
int GradientDrawableGradient_centerX	X coordinate of the origin of the gradient within the shape.
int GradientDrawableGradient_centerY	Y coordinate of the origin of the gradient within the shape.
int GradientDrawableGradient_endColor	End color of the gradient.
int GradientDrawableGradient_gradientRadius	Radius of the gradient, used only with radial gradient.
int GradientDrawableGradient_startColor	Start color of the gradient.
int GradientDrawableGradient_type	Type of gradient.
int GradientDrawableGradient_useLevel	This symbol is the offset where the <code>useLevel</code> ( <a href="#">/reference/android/R.attr.useLevel</a> ) attribute's value can be found in the <code>GradientDrawableGradient</code> ( <a href="#">/reference/android/R.styleable.html#GradientDrawableGradient</a> ) array.
int GradientDrawablePadding_bottom	Amount of bottom padding inside the gradient shape.
int GradientDrawablePadding_left	Amount of left padding inside the gradient shape.
int GradientDrawablePadding_right	Amount of right padding inside the gradient shape.
int GradientDrawablePadding_top	Amount of top padding inside the gradient shape.
int GradientDrawableSize_height	Height of the gradient shape.
int GradientDrawableSize_width	Width of the gradient shape.
int GradientDrawableSolid_color	Solid color for the gradient shape.
int GradientDrawableStroke_color	Color of the gradient shape's stroke.
int GradientDrawableStroke_dashGap	Gap between dashes in the stroke.
int GradientDrawableStroke_dashWidth	Length of a dash in the stroke.
int GradientDrawableStroke_width	Width of the gradient shape's stroke.

int GradientDrawable_dither	Enables or disables dithering.
int GradientDrawable_innerRadius	Inner radius of the ring.
int GradientDrawable_innerRadiusRatio	Inner radius of the ring expressed as a ratio of the ring's width.
int GradientDrawable_shape	Indicates what shape to fill with a gradient.
int GradientDrawable_thickness	Thickness of the ring.
int GradientDrawable_thicknessRatio	Thickness of the ring expressed as a ratio of the ring's width.
int GradientDrawable_useLevel	Indicates whether the drawable's level affects the way the gradient is drawn.
int GradientDrawable_visible	Indicates whether the drawable should initially be visible.
int GridLayoutAnimation_columnDelay	Fraction of the animation duration used to delay the beginning of the animation for a column.
int GridLayoutAnimation_direction	Direction of the animation in the grid.
int GridLayoutAnimation_directionPriority	Priority of the rows and columns.
int GridLayoutAnimation_rowDelay	Fraction of the animation duration used to delay the beginning of the animation for a row.
int GridLayout_Layout_layout_column	The column boundary delimiting the left of the group of cells occupied by this view.
int GridLayout_Layout_layout_columnSpan	The column span: the difference between the right and left boundaries defining the group of cells occupied by this view.
int GridLayout_Layout_layout_gravity	Gravity specifies how a component should be placed in its group of cells.
int GridLayout_Layout_layout_row	The row boundary delimiting the top of the group of cells occupied by this view.
int GridLayout_Layout_layout_rowSpan	The row span: the difference between the bottom and top boundaries defining the group of cells occupied by this view.
int GridLayout_alignmentMode	When set to alignMargins, causes alignment to take place between the outer margins of the view, as defined by its margins.
int GridLayout_columnCount	The maximum number of columns to create when automatically positioning the view.
int GridLayout_columnOrderPreserved	When set to true, forces column boundaries to appear in the same order as specified in the layout parameters.
int GridLayout_orientation	The orientation property is not used during layout.
int GridLayoutRowCount	The maximum number of rows to create when automatically positioning the view.
int GridLayout_rowOrderPreserved	When set to true, forces row boundaries to appear in the same order as specified in the layout parameters.
int GridLayout_useDefaultMargins	When set to true, tells GridLayout to use default margins when none are specified in the layout parameters.
int GridView_columnWidth	Specifies the fixed width for each column.
int GridView_gravity	Specifies the gravity within each cell.

int GridView_horizontalSpacing	Defines the default horizontal spacing between columns.
int GridView_numColumns	Defines how many columns to show.
int GridView_stretchMode	Defines how columns should stretch to fill the available empty space, if any.
int GridView_verticalSpacing	Defines the default vertical spacing between rows.
int HorizontalScrollView_fillViewport	Defines whether the scrollview should stretch its content to fill the viewport.
int IconDefault_icon	A Drawable resource providing a graphical representation of its associated icon.
int IconMenuView_maxItemsPerRow	Defines the maximum number of items per row.
int IconMenuView_maxRows	Defines the maximum number of rows displayed.
int IconMenuView_moreIcon	'More' icon.
int IconMenuView_rowHeight	Defines the height of each row.
int Icon_icon	A Drawable resource providing a graphical representation of its associated icon.
int Icon_mimeType	Specify a MIME type that is handled, as per <code>IntentFilter.addDataType()</code> <a href="#">reference/android/content/IntentFilter.html#addDataType(java.lang.String)</a>
int ImageView_adjustViewBounds	Set this to true if you want the ImageView to adjust its bounds to preserve its drawable.
int ImageView_baseline	The offset of the baseline within this view.
int ImageView_baselineAlignBottom	If true, the image view will be baseline aligned with based on its bottom edge.
int ImageView_cropToPadding	If true, the image will be cropped to fit within its padding.
int ImageView_maxHeight	An optional argument to supply a maximum height for this view.
int ImageView_maxWidth	An optional argument to supply a maximum width for this view.
int ImageView_scaleType	Controls how the image should be resized or moved to match the size of the view.
int ImageView_src	Sets a drawable as the content of this ImageView.
int ImageView_tint	Set a tinting color for the image.
int InputMethodService_imeExtractEnterAnimation	Animation to use when showing the fullscreen extract UI after it had previously been hidden.
int InputMethodService_imeExtractExitAnimation	Animation to use when hiding the fullscreen extract UI after it had previously been shown.
int InputMethodService_imeFullscreenBackground	Background to use for entire input method when it is being shown in fullscreen extract view, to ensure that it completely covers the application.
int InputMethod_Subtype_icon	The icon of the subtype.
int InputMethod_Subtype_imeSubtypeExtraValue	The extra value of the subtype.
int InputMethod_Subtype_imeSubtypeLocale	The locale of the subtype.

int InputMethod_Subtype_imeSubtypeMode	The mode of the subtype.
int InputMethod_Subtype_isAuxiliary	Set true if the subtype is auxiliary.
int InputMethod_Subtype_label	The name of the subtype.
int InputMethod_Subtype_overridesImplicitlyEnabledSubtype	Set true when this subtype should be selected by default if no other subtype explicitly.
int InputMethod_Subtype_subtypeId	The unique id for the subtype.
int InputMethod_isDefault	Set to true in all of the configurations for which this input method should be the option as the default.
int InputMethod_settingsActivity	Component name of an activity that allows the user to modify the settings.
int InsetDrawable_drawable	Reference to a drawable resource to use for the frame.
int InsetDrawable_insetBottom	This symbol is the offset where the <code>insetBottom</code> ( <a href="#">/reference/android/R.attr.html#insetBottom</a> ) attribute's value can be found in the <code>InsetDrawable</code> ( <a href="#">/reference/android/R.styleable.html#InsetDrawable</a> ).
int InsetDrawable_insetLeft	This symbol is the offset where the <code>insetLeft</code> ( <a href="#">/reference/android/R.attr.html#insetLeft</a> ) attribute's value can be found in the <code>InsetDrawable</code> ( <a href="#">/reference/android/R.styleable.html#InsetDrawable</a> ).
int InsetDrawable_insetRight	This symbol is the offset where the <code>insetRight</code> ( <a href="#">/reference/android/R.attr.html#insetRight</a> ) attribute's value can be found in the <code>InsetDrawable</code> ( <a href="#">/reference/android/R.styleable.html#InsetDrawable</a> ).
int InsetDrawable_insetTop	This symbol is the offset where the <code>insetTop</code> ( <a href="#">/reference/android/R.attr.html#insetTop</a> ) attribute's value can be found in the <code>InsetDrawable</code> ( <a href="#">/reference/android/R.styleable.html#InsetDrawable</a> ) array.
int InsetDrawable_visible	Provides initial visibility state of the drawable; the default value is false.
int IntentCategory_name	Required name of the category.
int Intent_action	The action name to assign to the Intent, as per <code>Intent.setAction()</code> ( <a href="#">/reference/android/content/Intent.html#setAction(java.lang.String)</a> ).
int Intent_data	The data URI to assign to the Intent, as per <code>Intent.setData()</code> ( <a href="#">/reference/android/content/Intent.html#setData(android.net.Uri)</a> ).
int Intent_mimeType	The MIME type name to assign to the Intent, as per <code>Intent.setType()</code> ( <a href="#">/reference/android/content/Intent.html#setType(java.lang.String)</a> ).
int Intent_targetClass	The class part of the ComponentName to assign to the Intent, as per <code>Intent.setComponent()</code> ( <a href="#">/reference/android/content/Intent.html#setComponent(android.content.ComponentName)</a> ).
int Intent_targetPackage	The package part of the ComponentName to assign to the Intent, as per <code>Intent.setComponent()</code> ( <a href="#">/reference/android/content/Intent.html#setComponent(android.content.ComponentName)</a> ).
int KeyboardLayout_keyboardLayout	The key character map file resource.
int KeyboardLayout_label	The display label of the keyboard layout.

int KeyboardLayout_name	The name of the keyboard layout, must be unique in the receiver.
int KeyboardViewPreviewState_state_long_pressable	State for KeyboardView (/reference/android/inputmethodservice/Keyb preview background).
int KeyboardView_keyBackground	Image for the key.
int KeyboardView_keyPreviewHeight	Height of the key press feedback popup.
int KeyboardView_keyPreviewLayout	Layout resource for key press feedback.
int KeyboardView_keyPreviewOffset	Vertical offset of the key press feedback from the key.
int KeyboardView_keyTextColor	Color to use for the label in a key.
int KeyboardView_keyTextSize	Size of the text for character keys.
int KeyboardView_labelTextSize	Size of the text for custom keys with some text and no icon.
int KeyboardView_popupLayout	Layout resource for popup keyboards.
int KeyboardView_shadowColor	Place a shadow of the specified color behind the text.
int KeyboardView_shadowRadius	Radius of the shadow.
int KeyboardView_verticalCorrection	Amount to offset the touch Y coordinate by, for bias correction.
int Keyboard_Key_codes	The unicode value or comma-separated values that this key outputs.
int Keyboard_Key_iconPreview	The icon to show in the popup preview.
int Keyboard_Key_isModifier	Whether this is a modifier key such as Alt or Shift.
int Keyboard_Key_isRepeatable	Whether long-pressing on this key will make it repeat.
int Keyboard_Key_isSticky	Whether this is a toggle key.
int Keyboard_Key_keyEdgeFlags	Key edge flags.
int Keyboard_Key_keyIcon	The icon to display on the key instead of the label.
int Keyboard_Key_keyLabel	The label to display on the key.
int Keyboard_Key_keyOutputText	The string of characters to output when this key is pressed.
int Keyboard_Key_keyboardMode	Mode of the keyboard.
int Keyboard_Key_popupCharacters	The characters to display in the popup keyboard.
int Keyboard_Key_popupKeyboard	The XML keyboard layout of any popup keyboard.
int Keyboard_Row_keyboardMode	Mode of the keyboard.
int Keyboard_Row_rowEdgeFlags	Row edge flags.
int Keyboard_horizontalGap	Default horizontal gap between keys.

int Keyboard_keyHeight	Default height of a key, in pixels or percentage of display width.
int Keyboard_keyWidth	Default width of a key, in pixels or percentage of display width.
int Keyboard_verticalGap	Default vertical gap between rows of keys.
int LayerDrawableItem_bottom	Bottom coordinate of the layer.
int LayerDrawableItem_drawable	Drawable used to render the layer.
int LayerDrawableItem_id	Identifier of the layer.
int LayerDrawableItem_left	Left coordinate of the layer.
int LayerDrawableItem_right	Right coordinate of the layer.
int LayerDrawableItem_top	Top coordinate of the layer.
int LayerDrawable_opacity	Indicates the opacity of the layer.
int LayoutAnimation_animation	Animation to use on each child.
int LayoutAnimation_animationOrder	The order in which the animations will be started.
int LayoutAnimation_delay	Fraction of the animation duration used to delay the beginning of the animation.
int LayoutAnimation_interpolator	Interpolator used to interpolate the delay between the start of each animation.
int LevelListDrawableItem_drawable	Reference to a drawable resource to use for the frame.
int LevelListDrawableItem_maxLevel	The maximum level allowed for this item.
int LevelListDrawableItem_minLevel	The minimum level allowed for this item.
int LinearLayout_Layout_layout_gravity	Standard gravity constant that a child supplies to its parent.
int LinearLayout_Layout_layout_height	Specifies the basic height of the view.
int LinearLayout_Layout_layout_weight	This symbol is the offset where the <code>layout_weight</code> attribute's value can be applied to the <code>LinearLayout_Layout</code> array.
int LinearLayout_Layout_layout_width	Specifies the basic width of the view.
int LinearLayout_baselineAligned	When set to false, prevents the layout from aligning its children's baselines.
int LinearLayout_baselineAlignedChildIndex	When a linear layout is part of another layout that is baseline aligned, it causes its children to baseline align to (that is, which child <code>TextView</code> ).
int LinearLayout_divider	Drawable to use as a vertical divider between buttons.
int LinearLayout_dividerPadding	Size of padding on either end of a divider.
int LinearLayout_gravity	Specifies how an object should position its content, on both the X and Y axes.
int LinearLayout_measureWithLargestChild	When set to true, all children with a weight will be considered having the same largest child.

int LinearLayout_orientation	Should the layout be a column or a row? Use "horizontal" for a row, "vertical" for a column.
int LinearLayout_showDividers	Setting for which dividers to show.
int LinearLayout_weightSum	Defines the maximum weight sum.
int ListPreference_entries	The human-readable array to present as a list.
int ListPreference_entryValues	The array to find the value to save for a preference when an entry from entries is selected.
int ListView_divider	Drawable or color to draw between list items.
int ListView_dividerHeight	Height of the divider.
int ListView_entries	Reference to an array resource that will populate the ListView.
int ListView_footerDividersEnabled	When set to false, the ListView will not draw the divider before each footer item.
int ListView_headerDividersEnabled	When set to false, the ListView will not draw the divider after each header item.
int ListView_overScrollFooter	Drawable to draw below list content.
int ListView_overScrollHeader	Drawable to draw above list content.
int MapView_apiKey	Value is a string that specifies the Maps API Key to use.
int MediaRouteButton_mediaRouteTypes	The types of media routes the button and its resulting chooser will filter by.
int MediaRouteButton_minHeight	This symbol is the offset where the <code>minHeight</code> attribute's value can be found in the <code>MediaRouteButton</code> array. <small>(/reference/android/R.styleable.html#MediaRouteButton_minHeight)</small>
int MediaRouteButton_minWidth	This symbol is the offset where the <code>minWidth</code> attribute's value can be found in the <code>MediaRouteButton</code> array. <small>(/reference/android/R.styleable.html#MediaRouteButton_minWidth)</small>
int MenuGroup_checkableBehavior	Whether the items are capable of displaying a check mark.
int MenuGroup_enabled	Whether the items are enabled.
int MenuGroup_id	The ID of the group.
int MenuGroup_menuCategory	The category applied to all items within this group.
int MenuGroup_orderInCategory	The order within the category applied to all items within this group.
int MenuGroup_visible	Whether the items are shown/visible.
int MenuItemCheckedFocusedState_state_checkable	State identifier indicating that the object <i>may</i> display a check mark.
int MenuItemCheckedFocusedState_state_checked	State identifier indicating that the object is currently checked.
int MenuItemCheckedFocusedState_state_focused	State value for <code>StateListDrawable</code> . <small>(/reference/android/graphics/drawable/StateListDrawable.html), set when input focus.</small>

int MenuItemCheckedState_state_checkable	State identifier indicating that the object <i>may</i> display a check mark.
int MenuItemCheckedState_state_checked	State identifier indicating that the object is currently checked.
int MenuItemUncheckedFocusedState_state_checkable	State identifier indicating that the object <i>may</i> display a check mark.
int MenuItemUncheckedFocusedState_state_focused	State value for StateListDrawable (/reference/android/graphics/drawable/StateListDrawable.html), set when input focus.
int MenuItemUncheckedState_state_checkable	State identifier indicating that the object <i>may</i> display a check mark.
int MenuItem_actionLayout	An optional layout to be used as an action view.
int MenuItem_actionProviderClass	The name of an optional ActionProvider class to instantiate an action view operations such as default action for that menu item.
int MenuItem_actionViewClass	The name of an optional View class to instantiate and use as an action view.
int MenuItem_alphabeticShortcut	The alphabetic shortcut key.
int MenuItem_checkable	Whether the item is capable of displaying a check mark.
int MenuItem_checked	Whether the item is checked.
int MenuItem_enabled	Whether the item is enabled.
int MenuItem_icon	The icon associated with this item.
int MenuItem_id	The ID of the item.
int MenuItem_menuCategory	The category applied to the item.
int MenuItem_numericShortcut	The numeric shortcut key.
int MenuItem_onClick	Name of a method on the Context used to inflate the menu that will be called when clicked.
int MenuItem_orderInCategory	The order within the category applied to the item.
int MenuItem_showAsAction	How this item should display in the Action Bar, if present.
int MenuItem_title	The title associated with the item.
int MenuItem_titleCondensed	The condensed title associated with the item.
int MenuItem_visible	Whether the item is shown/visible.
int MenuView_headerBackground	Default background for the menu header.
int MenuView_horizontalDivider	Default horizontal divider between rows of menu items.
int MenuView_itemBackground	Default background for each menu item.
int MenuView_itemIconDisabledAlpha	Default disabled icon alpha for each menu item that shows an icon.
int MenuView_itemTextAppearance	Default appearance of menu item text.

int MenuView_verticalDivider	Default vertical divider between menu items.
int MenuView_windowAnimationStyle	Default animations for the menu.
int MipmapDrawableItem_drawable	Reference to a drawable resource to use for the frame.
int MultiPaneChallengeLayout_Layout_layout_gravity	Standard gravity constant that a child supplies to its parent.
int MultiPaneChallengeLayout_orientation	Influences how layout_centerWithinArea behaves Must be one of the following constant values.
int MultiSelectListPreference_entries	The human-readable array to present as a list.
int MultiSelectListPreference_entryValues	The array to find the value to save for a preference when an entry from entr
int MultiWaveView_directionDescriptions	Reference to an array resource that be used to announce the directions wit circle.
int MultiWaveView_targetDescriptions	Reference to an array resource that be used as description for the targets a
int NinePatchDrawable_dither	Enables or disables dithering of the bitmap if the bitmap does not have the configuration as the screen (for instance: a ARGB 8888 bitmap with an RG
int NinePatchDrawable_src	Identifier of the bitmap file.
int OvershootInterpolator_tension	This is the amount of tension.
int PopupWindowBackgroundState_state_above_anchor	State identifier indicating the popup will be above the anchor.
int PopupWindow_popupAnimationStyle	This symbol is the offset where the popupAnimationStyle (/reference/android/R.attr.html#popupAnimationStyle) attribute's value car PopupWindow (/reference/android/R.styleable.html#PopupWindow) arra
int PopupWindow_popupBackground	This symbol is the offset where the popupBackground (/reference/android/R.attr.html#popupBackground) attribute's value car PopupWindow (/reference/android/R.styleable.html#PopupWindow) arra
int PreferenceGroup_orderingFromXml	Whether to order the Preference under this group as they appear in the XML
int PreferenceHeader_breadCrumbShortTitle	The short title for the bread crumb of this item.
int PreferenceHeader_breadCrumbTitle	The title for the bread crumb of this item.
int PreferenceHeader_fragment	The fragment that is displayed when the user selects this item.
int PreferenceHeader_icon	An icon for the item.
int PreferenceHeader_id	Identifier value for the header.
int PreferenceHeader_summary	The summary for the item.
int PreferenceHeader_title	The title of the item that is shown to the user.
int Preference_defaultValue	The default value for the preference, which will be set either if persistence i
int Preference_dependency	The key of another Preference that this Preference will depend on.

int Preference_enabled	Whether the Preference is enabled.
int Preference_fragment	When used inside of a modern PreferenceActivity, this declares a new Preference to be shown when the user selects this item.
int Preference_icon	The optional icon for the preference Must be a reference to another resource, in the form "@[+][ package: ] type [ theme attribute in the form "?[ package: ] [ type: ] name".
int Preference_key	The key to store the Preference value.
int Preference_layout	The layout for the Preference in a PreferenceActivity screen.
int Preference_order	The order for the Preference (lower values are to be ordered first).
int Preference_persistent	Whether the Preference stores its value to the shared preferences.
int Preference_selectable	Whether the Preference is selectable.
int Preference_shouldDisableView	Whether the view of this Preference should be disabled when this Preference is selected.
int Preference_summary	The summary for the Preference in a PreferenceActivity screen.
int Preference_title	The title for the Preference in a PreferenceActivity screen.
int Preference_widgetLayout	The layout for the controllable widget portion of a Preference.
int ProgressBar_animationResolution	Timeout between frames of animation in milliseconds Must be an integer value, such as "100".
int ProgressBar_ineterminate	Allows to enable the indeterminate mode.
int ProgressBar_ineterminateBehavior	Defines how the indeterminate mode should behave when the progress reaches the maximum value.
int ProgressBar_ineterminateDrawable	Drawable used for the indeterminate mode.
int ProgressBar_ineterminateDuration	Duration of the indeterminate animation.
int ProgressBar_ineterminateOnly	Restricts to ONLY indeterminate mode (state-keeping progress mode will not be used).
int ProgressBar_interpolator	This symbol is the offset where the <code>interpolator</code> attribute's value can be found in the <code>ProgressBar</code> ( <code>/reference/android/R.styleable.html#ProgressBar</code> ) array.
int ProgressBar_max	Defines the maximum value the progress can take.
int ProgressBar_maxHeight	An optional argument to supply a maximum height for this view.
int ProgressBar_maxWidth	An optional argument to supply a maximum width for this view.
int ProgressBar_minHeight	This symbol is the offset where the <code>minHeight</code> attribute's value can be found in the <code>ProgressBar</code> ( <code>/reference/android/R.styleable.html#ProgressBar</code> ) array.
int ProgressBar_minWidth	This symbol is the offset where the <code>minWidth</code> attribute's value can be found in the <code>ProgressBar</code> ( <code>/reference/android/R.styleable.html#ProgressBar</code> ) array.

	( <a href="#">/reference/android/R.styleable.html#ProgressBar</a> ) array.
int ProgressBar_progress	Defines the default progress value, between 0 and max.
int ProgressBar_progressDrawable	Drawable used for the progress mode.
int ProgressBar_secondaryProgress	Defines the secondary progress value, between 0 and max.
int PropertyAnimator_propertyName	Name of the property being animated.
int RadioGroup_checkedButton	The id of the child radio button that should be checked by default within the radio group.
int RadioGroup_orientation	Should the radio group be a column or a row? Use "horizontal" for a row, "vertical" for a column.
int RatingBar_isIndicator	Whether this rating bar is an indicator (and non-changeable by the user).
int RatingBar_numStars	The number of stars (or rating items) to show.
int RatingBar_rating	The rating to set by default.
int RatingBar_stepSize	The step size of the rating.
int RecognitionService_settingsActivity	Component name of an activity that allows the user to modify the settings.
int RelativeLayout_Layout_layout_above	Positions the bottom edge of this view above the given anchor view ID.
int RelativeLayout_Layout_layout_alignBaseline	Positions the baseline of this view on the baseline of the given anchor view.
int RelativeLayout_Layout_layout_alignBottom	Makes the bottom edge of this view match the bottom edge of the given anchor view.
int RelativeLayout_Layout_layout_alignEnd	Makes the end edge of this view match the end edge of the given anchor view.
int RelativeLayout_Layout_layout_alignLeft	Makes the left edge of this view match the left edge of the given anchor view.
int RelativeLayout_Layout_layout_alignParentBottom	If true, makes the bottom edge of this view match the bottom edge of the parent.
int RelativeLayout_Layout_layout_alignParentEnd	If true, makes the end edge of this view match the end edge of the parent.
int RelativeLayout_Layout_layout_alignParentLeft	If true, makes the left edge of this view match the left edge of the parent.
int RelativeLayout_Layout_layout_alignParentRight	If true, makes the right edge of this view match the right edge of the parent.
int RelativeLayout_Layout_layout_alignParentStart	If true, makes the start edge of this view match the start edge of the parent.
int RelativeLayout_Layout_layout_alignParentTop	If true, makes the top edge of this view match the top edge of the parent.
int RelativeLayout_Layout_layout_alignRight	Makes the right edge of this view match the right edge of the given anchor view.
int RelativeLayout_Layout_layout_alignStart	Makes the start edge of this view match the start edge of the given anchor view.
int RelativeLayout_Layout_layout_alignTop	Makes the top edge of this view match the top edge of the given anchor view.
int RelativeLayout_Layout_layout_alignWithParentIfMissing	If set to true, the parent will be used as the anchor when the anchor cannot be positioned using layout_toLeftOf, layout_toRightOf, etc.
int RelativeLayout_Layout_layout_below	Positions the top edge of this view below the given anchor view ID.

int RelativeLayout_Layout_centerHorizontal	If true, centers this child horizontally within its parent.
int RelativeLayout_Layout_centerInParent	If true, centers this child horizontally and vertically within its parent.
int RelativeLayout_Layout_centerVertical	If true, centers this child vertically within its parent.
int RelativeLayout_Layout_toEndOf	Positions the start edge of this view to the end of the given anchor view ID.
int RelativeLayout_Layout_toLeftOf	Positions the right edge of this view to the left of the given anchor view ID.
int RelativeLayout_Layout_toRightOf	Positions the left edge of this view to the right of the given anchor view ID.
int RelativeLayout_Layout_toStartOf	Positions the end edge of this view to the start of the given anchor view ID.
int RelativeLayout_gravity	Specifies how an object should position its content, on both the X and Y axis bounds.
int RelativeLayout_ignoreGravity	Indicates what view should not be affected by gravity.
int RingtonePreference_ringtoneType	Which ringtone type(s) to show in the picker.
int RingtonePreference_showDefault	Whether to show an item for a default sound.
int RingtonePreference_showSilent	Whether to show an item for 'Silent'.
int RotarySelector_orientation	Use "horizontal" or "vertical".
int RotateAnimation_fromDegrees	This symbol is the offset where the fromDegrees (/reference/android/R.attr.html#fromDegrees) attribute's value can be found in the RotateAnimation (/reference/android/R.styleable.html#RotateAnimation) array.
int RotateAnimation_pivotX	This symbol is the offset where the pivotX (/reference/android/R.attr. attribute's value can be found in the RotateAnimation (/reference/android/R.styleable.html#RotateAnimation) array.
int RotateAnimation_pivotY	This symbol is the offset where the pivotY (/reference/android/R.attr. attribute's value can be found in the RotateAnimation (/reference/android/R.styleable.html#RotateAnimation) array.
int RotateAnimation_toDegrees	This symbol is the offset where the toDegrees (/reference/android/R.attr.html#toDegrees) attribute's value can be found in the RotateAnimation (/reference/android/R.styleable.html#RotateAnimation) array.
int RotateDrawable_drawable	Reference to a drawable resource to use for the frame.
int RotateDrawable_fromDegrees	This symbol is the offset where the fromDegrees (/reference/android/R.attr.html#fromDegrees) attribute's value can be found in the RotateDrawable (/reference/android/R.styleable.html#RotateDrawable) array.
int RotateDrawable_pivotX	This symbol is the offset where the pivotX (/reference/android/R.attr. attribute's value can be found in the RotateDrawable (/reference/android/R.styleable.html#RotateDrawable) array.
int RotateDrawable_pivotY	This symbol is the offset where the pivotY (/reference/android/R.attr. attribute's value can be found in the RotateDrawable (/reference/android/R.styleable.html#RotateDrawable) array.
int RotateDrawable_toDegrees	This symbol is the offset where the toDegrees (/reference/android/R.attr.html#toDegrees) attribute's value can be found in the RotateDrawable (/reference/android/R.styleable.html#RotateDrawable) array.

int RotateDrawable_visible	Provides initial visibility state of the drawable; the default value is false.
int ScaleAnimation_fromXScale	This symbol is the offset where the <code>fromXScale</code> ( <a href="#">/reference/android/R.attr.html#fromXScale</a> ) attribute's value can be found in the <code>ScaleAnimation</code> ( <a href="#">/reference/android/R.styleable.html#ScaleAnimation</a> ) array.
int ScaleAnimation_fromYScale	This symbol is the offset where the <code>fromYScale</code> ( <a href="#">/reference/android/R.attr.html#fromYScale</a> ) attribute's value can be found in the <code>ScaleAnimation</code> ( <a href="#">/reference/android/R.styleable.html#ScaleAnimation</a> ) array.
int ScaleAnimation_pivotX	This symbol is the offset where the <code>pivotX</code> ( <a href="#">/reference/android/R.attr.html#pivotX</a> ) attribute's value can be found in the <code>ScaleAnimation</code> ( <a href="#">/reference/android/R.styleable.html#ScaleAnimation</a> ) array.
int ScaleAnimation_pivotY	This symbol is the offset where the <code>pivotY</code> ( <a href="#">/reference/android/R.attr.html#pivotY</a> ) attribute's value can be found in the <code>ScaleAnimation</code> ( <a href="#">/reference/android/R.styleable.html#ScaleAnimation</a> ) array.
int ScaleAnimation_toXScale	This symbol is the offset where the <code>toXscale</code> ( <a href="#">/reference/android/R.attr.html#toXscale</a> ) attribute's value can be found in the <code>ScaleAnimation</code> ( <a href="#">/reference/android/R.styleable.html#ScaleAnimation</a> ) array.
int ScaleAnimation_toYScale	This symbol is the offset where the <code>toYScale</code> ( <a href="#">/reference/android/R.attr.html#toYScale</a> ) attribute's value can be found in the <code>ScaleAnimation</code> ( <a href="#">/reference/android/R.styleable.html#ScaleAnimation</a> ) array.
int ScaleDrawable_drawable	Reference to a drawable resource to draw with the specified scale.
int ScaleDrawable_scaleGravity	Specifies where the drawable is positioned after scaling.
int ScaleDrawable_scaleHeight	Scale height, expressed as a percentage of the drawable's bound.
int ScaleDrawable_scaleWidth	Scale width, expressed as a percentage of the drawable's bound.
int ScaleDrawable_useIntrinsicSizeAsMinimum	Use the drawable's intrinsic width and height as minimum size values.
int ScrollView_fillViewport	Defines whether the scrollview should stretch its content to fill the viewport.
int SearchView_iconifiedByDefault	The default state of the <code>SearchView</code> .
int SearchView_imeOptions	The IME options to set on the query text field.
int SearchView_inputType	The input type to set on the query text field.
int SearchView_maxWidth	An optional maximum width of the <code>SearchView</code> .
int SearchView_queryHint	An optional query hint string to be displayed in the empty query field.
int SearchableActionKey_keycode	This attribute denotes the action key you wish to respond to.
int SearchableActionKey_queryActionMsg	If you wish to handle an action key during normal search query entry, you run an action string here.
int SearchableActionKey_suggestActionMsg	If you wish to handle an action key while a suggestion is being displayed a two ways to handle this.
int SearchableActionKey_suggestActionMsgColumn	If you wish to handle an action key while a suggestion is being displayed a do not wish to enable this action key for every suggestion, then you can us control it on a suggestion-by-suggestion basis.

int Searchable_autoUrlDetect	If provided and true, URLs entered in the search dialog while searching will be detected and treated as URLs (show a 'go' button in the keyboard browser directly when user launches the URL instead of passing the URL to the search engine).
int Searchable_hint	If supplied, this string will be displayed as a hint to the user.
int Searchable_icon	<b>This is deprecated.</b>
int Searchable_imeOptions	Additional features you can enable in an IME associated with an editor to integrate with your application.
int Searchable_includeInGlobalSearch	If provided and true, this searchable activity will be included in any global targets.
int Searchable_inputType	The type of data being placed in a text field, used to help an input method let the user enter text.
int Searchable_label	This is the user-displayed name of the searchable activity.
int Searchable_queryAfterZeroResults	If provided and true, this searchable activity will be invoked for all queries in a session.
int Searchable_searchButtonText	If supplied, this string will be displayed as the text of the "Search" button.
int Searchable_searchMode	Additional features are controlled by mode bits in this field.
int Searchable_searchSettingsDescription	If provided, this string will be used to describe the searchable item in the system settings within system search settings.
int Searchable_searchSuggestAuthority	If provided, this is the trigger indicating that the searchable activity provides suggestions.
int Searchable_searchSuggestIntentAction	If provided, and not overridden by an action in the selected suggestion, this is placed in the action field of the Intent ( <a href="#">/reference/android/content/Intent.html#ACTION_SEARCH</a> ) when the user clicks a suggestion.
int Searchable_searchSuggestIntentData	If provided, and not overridden by an action in the selected suggestion, this is placed in the data field of the Intent ( <a href="#">/reference/android/content/Intent.html#ACTION_SEARCH</a> ) when the user clicks a suggestion.
int Searchable_searchSuggestPath	If provided, this will be inserted in the suggestions query Uri, after the authority provided but before the standard suggestions path.
int Searchable_searchSuggestSelection	If provided, suggestion queries will be passed into your query function as the selection parameter.
int Searchable_searchSuggestThreshold	If provided, this is the minimum number of characters needed to trigger suggestions.
int Searchable_voiceLanguage	If provided, this specifies the spoken language to be expected, and that it will be one set in the getDefault() ( <a href="#">/reference/java/util/Locale.html#getDefault()</a> ) Locale.
int Searchable_voiceLanguageModel	If provided, this specifies the language model that should be used by the voice system.
int Searchable_voiceMaxResults	If provided, enforces the maximum number of results to return, including the primary query which will always be provided as the SEARCH intent's primary query.
int Searchable_voicePromptText	If provided, this specifies a prompt that will be displayed during voice input.

int Searchable_voiceSearchMode	Voice search features are controlled by mode bits in this field.
int SeekBar_thumb	Draws the thumb on a seekbar.
int SeekBar_thumbOffset	An offset for the thumb that allows it to extend out of the range of the track.
int SelectionModeDrawables_actionModeCopyDrawable	Drawable to use for the Copy action button in Contextual Action Bar Must be a reference to another resource, in the form "@[+][ package: ] type: [ theme: ] name".
int SelectionModeDrawables_actionModeCutDrawable	Drawable to use for the Cut action button in Contextual Action Bar Must be a reference to another resource, in the form "@[+][ package: ] type: [ theme: ] name".
int SelectionModeDrawables_actionModePasteDrawable	Drawable to use for the Paste action button in Contextual Action Bar Must be a reference to another resource, in the form "@[+][ package: ] type: [ theme: ] name".
int SelectionModeDrawables_actionModeSelectAllDrawable	Drawable to use for the Select all action button in Contextual Action Bar Must be a reference to another resource, in the form "@[+][ package: ] type: [ theme: ] name".
int ShapeDrawablePadding_bottom	Bottom padding.
int ShapeDrawablePadding_left	Left padding.
int ShapeDrawablePadding_right	Right padding.
int ShapeDrawablePadding_top	Top padding.
int ShapeDrawable_color	Defines the color of the shape.
int ShapeDrawable_dither	Enables or disables dithering.
int ShapeDrawable_height	Defines the height of the shape.
int ShapeDrawable_width	Defines the width of the shape.
int SlidingDrawer_allowSingleTap	Indicates whether the drawer can be opened/closed by a single tap on the handle.
int SlidingDrawer_animateOnClick	Indicates whether the drawer should be opened/closed with an animation on the handle.
int SlidingDrawer_bottomOffset	Extra offset for the handle at the bottom of the SlidingDrawer.
int SlidingDrawer_content	Identifier for the child that represents the drawer's content.
int SlidingDrawer_handle	Identifier for the child that represents the drawer's handle.
int SlidingDrawer_orientation	Orientation of the SlidingDrawer.
int SlidingDrawer_topOffset	Extra offset for the handle at the top of the SlidingDrawer.
int SlidingTab_orientation	Use "horizontal" for a row, "vertical" for a column.
int SpellChecker_Subtype_label	The name of the subtype.

int SpellChecker_Subtype_subtypeExtraValue	The extra value of the subtype.
int SpellChecker_Subtype_subtypeLocale	The locale of the subtype.
int SpellChecker_Label	The name of the spell checker.
int SpellChecker_settingsActivity	Component name of an activity that allows the user to modify the settings
int Spinner_dropDownHorizontalOffset	Horizontal offset from the spinner widget for positioning the dropdown in spinnerMode="dropdown".
int Spinner_dropDownSelector	List selector to use for spinnerMode="dropdown" display.
int Spinner_dropDownVerticalOffset	Vertical offset from the spinner widget for positioning the dropdown in spinnerMode="dropdown".
int Spinner_dropDownWidth	Width of the dropdown in spinnerMode="dropdown".
int Spinner_gravity	Gravity setting for positioning the currently selected item.
int Spinner_popupBackground	Background drawable to use for the dropdown in spinnerMode="dropdown".
int Spinner_prompt	The prompt to display when the spinner's dialog is shown.
int Spinner_spinnerMode	Display mode for spinner options.
int StateListDrawable_constantSize	If true, the drawable's reported internal size will remain constant as the state changes, even if it is larger than the maximum of all of the states.
int StateListDrawable_dither	Enables or disables dithering of the bitmap if the bitmap does not have the same configuration as the screen (for instance: a ARGB 8888 bitmap with an RGB 565 screen).
int StateListDrawable_enterFadeDuration	Amount of time (in milliseconds) to fade in a new state drawable.
int StateListDrawable_exitFadeDuration	Amount of time (in milliseconds) to fade out an old state drawable.
int StateListDrawable_variablePadding	If true, allows the drawable's padding to change based on the current state.
int StateListDrawable_visible	Indicates whether the drawable should be initially visible.
int SwitchPreference_disableDependentsState	The state (true for on, or false for off) that causes dependents to be disabled.
int SwitchPreference_summaryOff	The summary for the Preference in a PreferenceActivity screen when the Switch preference is unchecked.
int SwitchPreference_summaryOn	The summary for the Preference in a PreferenceActivity screen when the Switch preference is checked.
int SwitchPreference_switchTextOff	The text used on the switch itself when in the "off" state.
int SwitchPreference_switchTextOn	The text used on the switch itself when in the "on" state.
int Switch_switchMinWidth	Minimum width for the switch component
int Switch_switchMinWidth	Must be a dimension value, which is a floating point number appended with a unit suffix such as "14.5sp".
int Switch_switchMinWidth	Minimum space between the switch and caption text

int Switch_switchPadding	Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".
int Switch_switchTextAppearance	TextAppearance style for text displayed on the switch thumb.
int Switch_textOff	Text to use when the switch is in the unchecked/"off" state.
int Switch_textOn	Text to use when the switch is in the checked/"on" state.
int Switch_thumb	Drawable to use as the "thumb" that switches back and forth.
int Switch_thumbTextPadding	Amount of padding on either side of text within the switch thumb.
int Switch_track	Drawable to use as the "track" that the switch thumb slides within.
int SyncAdapter_accountType	The account type this authenticator handles.
int SyncAdapter_allowParallelSyncs	Set to true to tell the SyncManager that this SyncAdapter supports multiple syncs for the same account type and authority.
int SyncAdapter_contentAuthority	the authority of a content provider.
int SyncAdapter_isAlwaysSyncable	Set to true to tell the SyncManager to automatically call setIsSyncable(..., true) on this SyncAdapter instead of issuing an initialization sync to the SyncAdapter.
int SyncAdapter_settingsActivity	If provided, specifies the action of the settings activity for this SyncAdapter.
int SyncAdapter_supportsUploading	This symbol is the offset where the supportsUploading (/reference/android/R.styleable.html#SyncAdapter的支持上传) attribute's value can be found. SyncAdapter (/reference/android/R.styleable.html#SyncAdapter) array.
int SyncAdapter_userVisible	This symbol is the offset where the userVisible (/reference/android/R.styleable.html#SyncAdapter的用户可见) attribute's value can be found. SyncAdapter (/reference/android/R.styleable.html#SyncAdapter) array.
int TabWidget_divider	Drawable used to draw the divider between tabs.
int TabWidget_tabStripEnabled	Determines whether the strip under the tab indicators is drawn or not.
int TabWidget_tabStripLeft	Drawable used to draw the left part of the strip underneath the tabs.
int TabWidget_tabStripRight	Drawable used to draw the right part of the strip underneath the tabs.
int TableLayoutCollapseColumns	The zero-based index of the columns to collapse.
int TableLayout_shrinkColumns	The zero-based index of the columns to shrink.
int TableLayout_stretchColumns	The zero-based index of the columns to stretch.
int TableRow_Cell_layout_column	The index of the column in which this child should be.
int TableRow_Cell_layout_span	Defines how many columns this child should span.
int TextAppearance_fontFamily	Font family (named by string) for the text.
int TextAppearance_textAllCaps	Present the text in ALL CAPS.
int TextAppearance_textColor	Text color.

int TextAppearance_textColorHighlight	Color of the text selection highlight.
int TextAppearance_textColorHint	Color of the hint text.
int TextAppearance_textColorLink	Color of the links.
int TextAppearance textSize	Size of the text.
int TextAppearance textStyle	Style (bold, italic, bolditalic) for the text.
int TextAppearance typeface	Typeface (normal, sans, serif, monospace) for the text.
int TextClock_format12Hour	Specifies the formatting pattern used to show the time and/or date in 12-h
int TextClock_format24Hour	Specifies the formatting pattern used to show the time and/or date in 24-h
int TextClock_timeZone	Specifies the time zone to use.
int TextToSpeechEngine_settingsActivity	Component name of an activity that allows the user to modify the settings
int TextViewAppearance_textAppearance	Base text color, typeface, size, and style.
int TextViewMultiLineBackgroundState_state_multiline	State identifier indicating a TextView has a multi-line layout.
int TextView_autoLink	Controls whether links such as urls and email addresses are automatically converted to clickable links.
int TextView_autoText	If set, specifies that this TextView has a textual input method and automatically correct common spelling errors.
int TextView_bufferType	Determines the minimum type that getText() will return.
int TextView_capitalize	If set, specifies that this TextView has a textual input method and should automatically capitalize what the user types.
int TextView_cursorVisible	Makes the cursor visible (the default) or invisible.
int TextView_digits	If set, specifies that this TextView has a numeric input method and that the characters are the ones that it will accept.
int TextView_drawableBottom	The drawable to be drawn below the text.
int TextView_drawableEnd	The drawable to be drawn to the end of the text.
int TextView_drawableLeft	The drawable to be drawn to the left of the text.
int TextView_drawablePadding	The padding between the drawables and the text.
int TextView_drawableRight	The drawable to be drawn to the right of the text.
int TextView_drawableStart	The drawable to be drawn to the start of the text.
int TextView_drawableTop	The drawable to be drawn above the text.
int TextView_editable	If set, specifies that this TextView has an input method.
int TextView_editorExtras	Reference to an <input-extras> (/reference/android/R.styleable.htm) XML resource containing additional data to supply to an input method, which

	implementation of the input method.
int TextView_ellipsize	If set, causes words that are longer than the view is wide to be ellipsized in the middle.
int TextView_ems	Makes the TextView be exactly this many ems wide.
int TextView_enabled	Specifies whether the widget is enabled.
int TextView_fontFamily	Font family (named by string) for the text.
int TextView_freezesText	If set, the text view will include its current complete text inside of its frozen meta-data such as the current cursor position.
int TextView_gravity	Specifies how to align the text by the view's x- and/or y-axis when the text is viewed.
int TextView_height	Makes the TextView be exactly this many pixels tall.
int TextView_hint	Hint text to display when the text is empty.
int TextView imeActionId	Supply a value for EditorInfo.actionId ( <a href="#">/reference/android/view/inputmethod/EditorInfo.html#actionId</a> ) used when the method is connected to the text view.
int TextView imeActionButton	Supply a value for EditorInfo.actionLabel ( <a href="#">/reference/android/view/inputmethod/EditorInfo.html#actionLabel</a> ) used when the method is connected to the text view.
int TextView imeOptions	Additional features you can enable in an IME associated with an editor to facilitate integration with your application.
int TextView_includeFontPadding	Leave enough room for ascenders and descenders instead of using the font strictly.
int TextView_inputMethod	If set, specifies that this TextView should use the specified input method (as a qualified class name).
int TextView_inputType	The type of data being placed in a text field, used to help an input method decide how to enter text.
int TextView_lineSpacingExtra	Extra spacing between lines of text.
int TextView_lineSpacingMultiplier	Extra spacing between lines of text, as a multiplier.
int TextView_lines	Makes the TextView be exactly this many lines tall.
int TextView_linksClickable	If set to false, keeps the movement method from being set to the link movement mode. autoLink causes links to be found.
int TextView_marqueeRepeatLimit	The number of times to repeat the marquee animation.
int TextView_maxEms	Makes the TextView be at most this many ems wide.
int TextView_maxHeight	Makes the TextView be at most this many pixels tall.
int TextView_maxLength	Set an input filter to constrain the text length to the specified number.

int TextView_maxLines	Makes the TextView be at most this many lines tall.
int TextView_maxWidth	Makes the TextView be at most this many pixels wide.
int TextView_minEms	Makes the TextView be at least this many ems wide.
int TextView_minHeight	Makes the TextView be at least this many pixels tall.
int TextView_minLines	Makes the TextView be at least this many lines tall.
int TextView_minWidth	Makes the TextView be at least this many pixels wide.
int TextView_numeric	If set, specifies that this TextView has a numeric input method.
int TextView_password	Whether the characters of the field are displayed as password dots instead of plain text.
int TextView_phoneNumber	If set, specifies that this TextView has a phone number input method.
int TextView_privateImeOptions	An additional content type description to supply to the input method attached which is private to the implementation of the input method.
int TextView_scrollHorizontally	Whether the text is allowed to be wider than the view (and therefore can be scrollable horizontally).
int TextView_selectAllOnFocus	If the text is selectable, select it all when the view takes focus.
int TextView_shadowColor	Place a shadow of the specified color behind the text.
int TextView_shadowDx	Horizontal offset of the shadow.
int TextView_shadowDy	Vertical offset of the shadow.
int TextView_shadowRadius	Radius of the shadow.
int TextView_singleLine	Constrains the text to a single horizontally scrolling line instead of letting multiple lines, and advances focus instead of inserting a newline when you press the enter key.
int TextView_text	Text to display.
int TextView_textAllCaps	Present the text in ALL CAPS.
int TextView_textAppearance	Base text color, typeface, size, and style.
int TextView_textColor	Text color.
int TextView_textColorHighlight	Color of the text selection highlight.
int TextView_textColorHint	Color of the hint text.
int TextView_textColorLink	Text color for links.
int TextView_textCursorDrawable	Reference to a drawable that will be drawn under the insertion cursor.
int TextView_textEditNoPasteWindowLayout	Variation of textEditPasteWindowLayout displayed when the clipboard is empty.
int TextView_textEditPasteWindowLayout	The layout of the view that is displayed on top of the cursor to paste inside.
int TextView_textEditSideNoPasteWindowLayout	Variation of textEditSidePasteWindowLayout displayed when the clipboard is empty.

int TextView_textEditSidePasteWindowLayout	Used instead of <code>textEditPasteWindowLayout</code> when the window is moved or insertion cursor because it would be clipped if it were positioned on top.
int TextView_textEditSuggestionItemLayout	Layout of the <code>TextView</code> item that will populate the suggestion popup window.
int TextView_textIsSelectable	Indicates that the content of a non-editable text can be selected.
int TextView_textScaleX	Sets the horizontal scaling factor for the text.
int TextView_textSelectHandle	Reference to a drawable that will be used to display a text selection anchor cursor within text.
int TextView_textSelectHandleLeft	Reference to a drawable that will be used to display a text selection anchor for the left side of a selection region.
int TextView_textSelectHandleRight	Reference to a drawable that will be used to display a text selection anchor for the right side of a selection region.
int TextView textSize	Size of the text.
int TextView textStyle	Style (bold, italic, bolditalic) for the text.
int TextView_typeface	Typeface (normal, sans, serif, monospace) for the text.
int TextView_width	Makes the <code>TextView</code> be exactly this many pixels wide.
int Theme_absListViewStyle	Default <code>AbsListView</code> style.
int Theme_actionBarDivider	Custom divider drawable to use for elements in the action bar.
int Theme_actionBarItemBackground	Custom item state list drawable background for action bar items.
int Theme_actionBarSize	Size of the Action Bar, including the contextual bar used to present Action Bar items.
int Theme_actionBarSplitStyle	Reference to a style for the split Action Bar.
int Theme_actionBarStyle	Reference to a style for the Action Bar
int Theme_actionBarTabBarStyle	Must be a reference to another resource, in the form " <code>@[+][ package: ] type: [ name ]</code> ". The <code>theme</code> attribute in the form " <code>?[ package: ] [ type: ] name</code> ".  This symbol is the offset where the <code>actionBarTabBarStyle</code> ( <code>/reference/android/R.attr.html#actionBarTabBarStyle</code> ) attribute's value is defined in the <code>Theme</code> ( <code>/reference/android/R.styleable.html#Theme</code> ) array.
int Theme_actionBarTabStyle	Default style for tabs within an action bar
int Theme_actionBarWidgetTheme	Must be a reference to another resource, in the form " <code>@[+][ package: ] type: [ name ]</code> ".  This symbol is the offset where the <code>actionBarTabTextStyle</code> ( <code>/reference/android/R.attr.html#actionBarTabTextStyle</code> ) attribute's value is defined in the <code>Theme</code> ( <code>/reference/android/R.styleable.html#Theme</code> ) array.
int ThemeActionButtonStyle	Default action button style.
int ThemeActionDropDownStyle	Default ActionBar dropdown style.

int Theme_actionMenuTextAppearance	TextAppearance style that will be applied to text that appears within action menu items.
int Theme_actionMenuTextColor	Color for text that appears within action menu items.
int Theme_actionModeBackground	Background drawable to use for action mode UI
	Must be a reference to another resource, in the form "@[+][ package: ] type: [ theme: ]".
int Theme_actionModeCloseButtonStyle	This symbol is the offset where the <code>actionModeCloseButtonStyle</code> ( <code>/reference/android/R.attr.html#actionModeCloseButtonStyle</code> ) attribute's value can be found in the <code>Theme</code> ( <code>/reference/android/R.styleable.html#Theme</code> ) array
	Drawable to use for the close action mode button
int Theme_actionModeCloseDrawable	Must be a reference to another resource, in the form "@[+][ package: ] type: [ theme: ]".
int Theme_actionModeCopyDrawable	Drawable to use for the Copy action button in Contextual Action Bar
	Must be a reference to another resource, in the form "@[+][ package: ] type: [ theme: ]".
int Theme_actionModeCutDrawable	Drawable to use for the Cut action button in Contextual Action Bar
	Must be a reference to another resource, in the form "@[+][ package: ] type: [ theme: ]".
int Theme_actionModePasteDrawable	Drawable to use for the Paste action button in Contextual Action Bar
	Must be a reference to another resource, in the form "@[+][ package: ] type: [ theme: ]".
int Theme_actionModeSelectAllDrawable	Drawable to use for the Select all action button in Contextual Action Bar
	Must be a reference to another resource, in the form "@[+][ package: ] type: [ theme: ]".
int Theme_actionModeSplitBackground	Background drawable to use for action mode UI in the lower split bar
	Must be a reference to another resource, in the form "@[+][ package: ] type: [ theme: ]".
int Theme_actionModeStyle	This symbol is the offset where the <code>actionModeStyle</code> ( <code>/reference/android/R.attr.html#actionModeStyle</code> ) attribute's value can be found in the <code>Theme</code> ( <code>/reference/android/R.styleable.html#Theme</code> ) array.
int Theme_actionOverflowButtonStyle	This symbol is the offset where the <code>actionOverflowButtonStyle</code> ( <code>/reference/android/R.attr.html#actionOverflowButtonStyle</code> ) attribute's value can be found in the <code>Theme</code> ( <code>/reference/android/R.styleable.html#Theme</code> ) array
int Theme_activatedBackgroundIndicator	Drawable used as a background for activated items.
int Theme_alertDialogIcon	Icon drawable to use for alerts
	Must be a reference to another resource, in the form "@[+][ package: ] type: [ theme: ]".
int Theme_alertDialogStyle	This symbol is the offset where the <code>alertDialogStyle</code> ( <code>/reference/android/R.attr.html#alertDialogStyle</code> ) attribute's value can be found in the <code>Theme</code> ( <code>/reference/android/R.styleable.html#Theme</code> ) array.
int Theme_alertDialogTheme	Theme to use for alert dialogs spawned from this theme.

int Theme_autoCompleteTextViewStyle	Default AutoCompleteTextView style.
int Theme_backgroundDimAmount	Default background dim amount when a menu, dialog, or something similar is displayed.
int Theme_backgroundDimEnabled	Control whether dimming behind the window is enabled.
int Theme_borderlessButtonStyle	Style for buttons without an explicit border, often used in groups.
int Theme_buttonBarButtonStyle	Style for buttons within button bars  Must be a reference to another resource, in the form "@[+][ package: ] type[ theme attribute in the form "?[ package: ][ type: ]name".
int Theme_buttonBarStyle	Style for button bars  Must be a reference to another resource, in the form "@[+][ package: ] type[ theme attribute in the form "?[ package: ][ type: ]name".
int Theme_buttonStyle	Normal Button style.
int Theme_buttonStyleInset	Button style to inset into an EditText.
int Theme_buttonStyleSmall	Small Button style.
int Theme_buttonStyleToggle	ToggleButton style.
int Theme_calendarViewStyle	The CalendarView style.
int Theme_candidatesTextStyleSpans	A styled string, specifying the style to be used for showing inline candidate composing with an input method.
int Theme_checkBoxPreferenceStyle	Default style for CheckBoxPreference.
int Theme_checkboxStyle	Default Checkbox style.
int Theme_checkedTextViewStyle	Default CheckedTextView style.
int Theme_colorActivatedHighlight	Default highlight color for items that are activated.
int Theme_colorBackground	Color that matches (as closely as possible) the window background.
int Theme_colorBackgroundCacheHint	This is a hint for a solid color that can be used for caching rendered views.
int Theme_colorFocusedHighlight	Default highlight color for items that are focused.
int Theme_colorForeground	Default color of foreground imagery.
int Theme_colorForegroundInverse	Default color of foreground imagery on an inverted background.
int Theme_colorLongPressedHighlight	Default highlight color for items that are long-pressed.
int Theme_colorMultiSelectHighlight	Default highlight color for items in multiple selection mode.
int Theme_colorPressedHighlight	Default highlight color for items that are pressed.
int Theme_datePickerStyle	The DatePicker style.
int Theme_detailsElementBackground	Background that can be used behind parts of a UI that provide details on demand.

	selecting.
int Theme_dialogPreferenceStyle	Default style for DialogPreference.
int Theme_dialogTheme	Theme to use for dialogs spawned from this theme.
int Theme_disabledAlpha	Default disabled alpha for widgets that set enabled/disabled alpha programmatically.
int Theme_dividerHorizontal	Drawable to use for generic horizontal dividers.
int Theme_dividerVertical	Drawable to use for generic vertical dividers.
int Theme_dropDownHintAppearance	Default style for drop down hints.
int Theme_dropDownItemStyle	Default style for drop down items.
int Theme_dropDownListviewStyle	Default ListView style for drop downs.
int Theme_dropDownSpinnerStyle	Default dropdown Spinner style.
int Theme_editTextBackground	EditText background drawable.
int Theme_editTextColor	EditText text foreground color.
int Theme_editTextPreferenceStyle	Default style for EditTextPreference.
int Theme_editTextStyle	Default EditText style.
int Theme_expandableListPreferredChildIndicatorLeft	The preferred left bound for an expandable list child's indicator.
int Theme_expandableListPreferredChildIndicatorRight	The preferred right bound for an expandable list child's indicator.
int Theme_expandableListPreferredChildPaddingLeft	The preferred left padding for an expandable list item that is a child.
int Theme_expandableListPreferredItemIndicatorLeft	The preferred left bound for an expandable list item's indicator.
int Theme_expandableListPreferredItemIndicatorRight	The preferred right bound for an expandable list item's indicator.
int Theme_expandableListPreferredItemPaddingLeft	The preferred left padding for an expandable list item (for child-specific layout). (This is the same value as expandableListPreferredChildPaddingLeft).
int Theme_expandableListViewStyle	Default ExpandableListView style.
int Theme_expandableListViewWhiteStyle	ExpandableListView with white background.
int Theme_fastScrollOverlayPosition	Position of the fast scroll index overlay window.
int Theme_fastScrollPreviewBackgroundLeft	Drawable to use as the fast scroll index preview window background where the scroll bar is on the left.
int Theme_fastScrollPreviewBackgroundRight	Drawable to use as the fast scroll index preview window background where the scroll bar is on the right.
int Theme_fastScrollTextColor	Text color for the fast scroll index overlay.
int Theme_fastScrollThumbDrawable	Drawable to use as the fast scroll thumb.
int Theme_fastScrollTrackDrawable	Drawable to use as the track for the fast scroll thumb.

int Theme_galleryItemBackground	The preferred background for gallery items.
int Theme_galleryStyle	Default Gallery style.
int Theme_gridViewStyle	Default GridView style.
int Theme_homeAsUpIndicator	Specifies a drawable to use for the 'home as up' indicator.
int Theme_horizontalScrollViewStyle	Default HorizontalScrollView style.
int Theme_imageButtonStyle	The style resource to use for an ImageButton.
int Theme_imageWellStyle	The style resource to use for an ImageButton that is an image well.
int Theme_listChoiceBackgroundIndicator	Drawable used as a background for selected list items.
int Theme_listChoiceIndicatorMultiple	Drawable to use for multiple choice indicators.
int Theme_listChoiceIndicatorSingle	Drawable to use for single choice indicators.
int Theme_listDivider	The drawable for the list divider.
int Theme_listDividerAlertDialog	The list divider used in alert dialogs.
int Theme_listPopupWindowStyle	Default ListPopupWindow style.
int Theme_listPreferredItemHeight	The preferred list item height.
int Theme_listPreferredItemHeightLarge	A larger, more robust list item height.
int Theme_listPreferredItemHeightSmall	A smaller, sleeker list item height.
int Theme_listPreferredItemPaddingEnd	The preferred padding along the end edge of list items.
int Theme_listPreferredItemPaddingLeft	The preferred padding along the left edge of list items.
int Theme_listPreferredItemPaddingRight	The preferred padding along the right edge of list items.
int Theme_listPreferredItemPaddingStart	The preferred padding along the start edge of list items.
int Theme_listSeparatorTextViewStyle	TextView style for list separators.
int Theme_listViewStyle	Default ListView style.
int Theme_listViewWhiteStyle	ListView with white background.
int Theme_mapViewStyle	Default MapView style.
int Theme_mediaRouteButtonStyle	Default style for the MediaRouteButton widget.
int Theme_panelBackground	The background of a panel when it is inset from the left and right edges of
int Theme_panelColorBackground	Color that matches (as closely as possible) the panel background.
int Theme_panelColorForeground	Default color of foreground panel imagery.
int Theme_panelFullBackground	The background of a panel when it extends to the left and right edges of th

int Theme_panelTextAppearance	Default appearance of panel text.
int Theme_popupMenuStyle	Default PopupMenu style.
int Theme_popupWindowStyle	Default PopupWindow style.
int Theme_preferenceCategoryStyle	Default style for PreferenceCategory.
int Theme_preferenceInformationStyle	Default style for informational Preference.
int Theme_preferenceLayoutChild	The preference layout that has the child/tabbed effect.
int Theme_preferenceScreenStyle	Default style for PreferenceScreen.
int Theme_preferenceStyle	Default style for Preference.
int Theme_presentationTheme	Theme to use for presentations spawned from this theme.
int Theme_progressBarStyle	Default ProgressBar style.
int Theme_progressBarStyleHorizontal	Horizontal ProgressBar style.
int Theme_progressBarStyleInverse	Inverse ProgressBar style.
int Theme_progressBarStyleLarge	Large ProgressBar style.
int Theme_progressBarStyleLargeInverse	Large inverse ProgressBar style.
int Theme_progressBarStyleSmall	Small ProgressBar style.
int Theme_progressBarStyleSmallInverse	Small inverse ProgressBar style.
int Theme_progressBarStyleSmallTitle	Small ProgressBar in title style.
int Theme_quickContactBadgeStyleSmallWindowLarge	Default quickcontact badge style with large quickcontact window.
int Theme_quickContactBadgeStyleSmallWindowMedium	Default quickcontact badge style with medium quickcontact window.
int Theme_quickContactBadgeStyleSmallWindowSmall	Default quickcontact badge style with small quickcontact window.
int Theme_quickContactBadgeStyleWindowLarge	Default quickcontact badge style with large quickcontact window.
int Theme_quickContactBadgeStyleWindowMedium	Default quickcontact badge style with medium quickcontact window.
int Theme_quickContactBadgeStyleWindowSmall	Default quickcontact badge style with small quickcontact window.
int Theme_radioButtonStyle	Default RadioButton style.
int Theme_ratingBarStyle	Default RatingBar style.
int Theme_ratingBarStyleIndicator	Indicator RatingBar style.
int Theme_ratingBarStyleSmall	Small indicator RatingBar style.
int Theme_ringtonePreferenceStyle	Default style for RingtonePreference.

int Theme_scrollViewStyle	Default ScrollView style.
int Theme_seekBarStyle	Default SeekBar style.
int Theme_segmentedButtonStyle	Style for segmented buttons - a container that houses several buttons with a singel button broken into segments.
int Theme_selectableItemBackground	Background drawable for standalone items that need focus/pressed state.
int Theme_spinnerDropDownItemStyle	Default style for spinner drop down items.
int Theme_spinnerItemStyle	Default spinner item style.
int Theme_spinnerStyle	Default Spinner style.
int Theme_starStyle	Default Star style.
int Theme_switchPreferenceStyle	Default style for switch preferences.
int Theme_tabWidgetStyle	Default TabWidget style.
int Theme_textAppearance	Default appearance of text: color, typeface, size, and style.
int Theme_textAppearanceButton	Text color, typeface, size, and style for the text inside of a button.
int Theme_textAppearanceInverse	Default appearance of text against an inverted background: color, typeface, size, and style.
int Theme_textAppearanceLarge	Text color, typeface, size, and style for "large" text.
int Theme_textAppearanceLargeInverse	Text color, typeface, size, and style for "large" inverse text.
int Theme_textAppearanceLargePopupMenu	Text color, typeface, size, and style for the text inside of a popup menu.
int Theme_textAppearanceListItem	The preferred TextAppearance for the primary text of list items.
int Theme_textAppearanceListItemSmall	The preferred TextAppearance for the primary text of small list items.
int Theme_textAppearanceMedium	Text color, typeface, size, and style for "medium" text.
int Theme_textAppearanceMediumInverse	Text color, typeface, size, and style for "medium" inverse text.
int Theme_textAppearanceSearchResultSubtitle	Text color, typeface, size, and style for system search result subtitle.
int Theme_textAppearanceSearchResultTitle	Text color, typeface, size, and style for system search result title.
int Theme_textAppearanceSmall	Text color, typeface, size, and style for "small" text.
int Theme_textAppearanceSmallInverse	Text color, typeface, size, and style for "small" inverse text.
int Theme_textAppearanceSmallPopupMenu	Text color, typeface, size, and style for small text inside of a popup menu.
int Theme_textCheckMark	Drawable to use for check marks.
int Theme_textCheckMarkInverse	This symbol is the offset where the <code>textCheckMarkInverse</code> ( <code>/reference/android/R.attr.html#textCheckMarkInverse</code> ) attribute's value is defined in the <code>Theme</code> ( <code>/reference/android/R.styleable.html#Theme</code> ) array.

int Theme_textColorAlertDialogListItem	Color of list item text in alert dialogs.
int Theme_textColorHighlightInverse	Color of highlighted text, when used in a light theme.
int Theme_textColorHintInverse	Inverse hint text color.
int Theme_textColorLinkInverse	Color of link text (URLs), when used in a light theme.
int Theme_textColorPrimary	The most prominent text color.
int Theme_textColorPrimaryDisableOnly	Bright text color.
int Theme_textColorPrimaryInverse	Primary inverse text color, useful for inverted backgrounds.
int Theme_textColorPrimaryInverseDisableOnly	Bright inverse text color.
int Theme_textColorPrimaryInverseNoDisable	Bright inverse text color.
int Theme_textColorPrimaryNoDisable	Bright text color.
int Theme_textColorSecondary	Secondary text color.
int Theme_textColorSecondaryInverse	Secondary inverse text color, useful for inverted backgrounds.
int Theme_textColorSecondaryInverseNoDisable	Dim inverse text color.
int Theme_textColorSecondaryNoDisable	Dim text color.
int Theme_textColorTertiary	Tertiary text color.
int Theme_textColorTertiaryInverse	Tertiary inverse text color, useful for inverted backgrounds.
int ThemeTextEditNoPasteWindowLayout	Variation of textEditPasteWindowLayout displayed when the clipboard is
int ThemeTextEditPasteWindowLayout	The layout of the view that is displayed on top of the cursor to paste inside
int ThemeTextEditSideNoPasteWindowLayout	Variation of textEditSidePasteWindowLayout displayed when the clipboard is
int ThemeTextEditSidePasteWindowLayout	Used instead of textEditPasteWindowLayout when the window is moved or insertion cursor because it would be clipped if it were positioned on top.
int ThemeTextEditSuggestionItemLayout	Layout of the TextView item that will populate the suggestion popup window.
int ThemeTextSelectHandle	Reference to a drawable that will be used to display a text selection anchor cursor within text.
int ThemeTextSelectHandleLeft	Reference to a drawable that will be used to display a text selection anchor for a selection region.
int ThemeTextSelectHandleRight	Reference to a drawable that will be used to display a text selection anchor for a selection region.
int ThemeTextSelectHandleWindowStyle	Reference to a style that will be used for the window containing a text selection.
int ThemeTextSuggestionsWindowStyle	Reference to a style that will be used for the window containing a list of suggestions in an EditText.
int ThemeTextViewStyle	Default TextView style.

int Theme_webTextViewStyle	Default WebTextView style.
int Theme_webViewStyle	Default WebView style.
int Theme_windowActionBar	Flag indicating whether this window should have an Action Bar in place of its title bar.
int Theme_windowActionBarOverlay	Flag indicating whether this window's Action Bar should overlay application content.
int Theme_windowActionModeOverlay	Flag indicating whether action modes should overlay window content when reserved space for their UI (such as an Action Bar).
int Theme_windowAnimationStyle	Reference to a style resource holding the set of window animations to use, the attributes defined by WindowAnimation ( <a href="#">/reference/android/R.styleable.html#WindowAnimation</a> ).
int Theme_windowBackground	Drawable to use as the overall window background.
int Theme_windowCloseOnTouchOutside	Control whether a container should automatically close itself if the user touches outside it.
int Theme_windowContentOverlay	This Drawable is overlaid over the foreground of the Window's content area and has a shadow below the title.
int Theme_windowDisablePreview	Flag allowing you to disable the preview animation for a window.
int Theme_windowEnableSplitTouch	Flag indicating that this window should allow touches to be split across one screen and also support split touch.
int Theme_windowFrame	Drawable to use as a frame around the window.
int Theme_windowFullscreen	Flag indicating whether this window should fill the entire screen.
int Theme_windowIsFloating	Flag indicating whether this is a floating window.
int Theme_windowIsTranslucent	Flag indicating whether this is a translucent window.
int Theme_windowNoDisplay	Flag indicating that this window should not be displayed at all.
int Theme_windowNoTitle	Flag indicating whether there should be no title on this window.
int Theme_windowShowWallpaper	Flag indicating that this window's background should be the user's current wallpaper.
int Theme_windowSoftInputMode	Defines the default soft input state that this window would like when it is created.
int Theme_windowTitleBackgroundStyle	The style resource to use for a window's title area.
int Theme_windowTitleSize	The style resource to use for a window's title bar height.
int Theme_windowTitleStyle	The style resource to use for a window's title text.
int Theme_yesNoPreferenceStyle	Default style for YesNoPreference.
int ToggleButton_disabledAlpha	The alpha to apply to the indicator when disabled.
int ToggleButton_textOff	The text for the button when it is not checked.
int ToggleButton_textOn	The text for the button when it is checked.

int TranslateAnimation_fromXDelta	This symbol is the offset where the <code>fromXDelta</code> attribute's value can be found in the <code>TranslateAnimation</code> array.
int TranslateAnimation_fromYDelta	This symbol is the offset where the <code>fromYDelta</code> attribute's value can be found in the <code>TranslateAnimation</code> array.
int TranslateAnimation_toXDelta	This symbol is the offset where the <code>toXDelta</code> attribute's value can be found in the <code>TranslateAnimation</code> array.
int TranslateAnimation_toYDelta	This symbol is the offset where the <code>toYDelta</code> attribute's value can be found in the <code>TranslateAnimation</code> array.
int TwoLineListItem_mode	This symbol is the offset where the <code>mode</code> value can be found in the <code>TwoLineListItem</code> array.
int VerticalSlider_Layout_layout_scale	This symbol is the offset where the <code>layout_scale</code> attribute's value can be found in the <code>VerticalSlider_Layout</code> array.
int ViewAnimator_animateFirstView	Defines whether to animate the current View when the ViewAnimation is fired.
int ViewAnimator_inAnimation	Identifier for the animation to use when a view is shown.
int ViewAnimator_outAnimation	Identifier for the animation to use when a view is hidden.
int ViewDrawableStates_state_accelerated	State value for <code>StateListDrawable</code> ( <a href="#">/reference/android/graphics/drawable/StateListDrawable.html</a> ), indicating the Drawable is in a view that is hardware accelerated.
int ViewDrawableStates_state_activated	State value for <code>StateListDrawable</code> ( <a href="#">/reference/android/graphics/drawable/StateListDrawable.html</a> ), set when the parent has been "activated" meaning the user has currently marked it as being active.
int ViewDrawableStates_state_drag_can_accept	State for <code>StateListDrawable</code> ( <a href="#">/reference/android/graphics/drawable/StateListDrawable.html</a> ) indicating the Drawable is in a view that is capable of accepting a drop of the content currently being manipulated in a drag-and-drop operation.
int ViewDrawableStates_state_drag_hovered	State for <code>StateListDrawable</code> ( <a href="#">/reference/android/graphics/drawable/StateListDrawable.html</a> ) indicating the operation (for which the Drawable's view is a valid recipient) is currently performed on the Drawable.
int ViewDrawableStates_state_enabled	State value for <code>StateListDrawable</code> ( <a href="#">/reference/android/graphics/drawable/StateListDrawable.html</a> ), set when the view is enabled.
int ViewDrawableStates_state_focused	State value for <code>StateListDrawable</code> ( <a href="#">/reference/android/graphics/drawable/StateListDrawable.html</a> ), set when the view has input focus.
int ViewDrawableStates_state_hovered	State value for <code>StateListDrawable</code> ( <a href="#">/reference/android/graphics/drawable/StateListDrawable.html</a> ), set when the view is hovering over the view.
	State value for <code>StateListDrawable</code>

int ViewDrawableStates_state_pressed	( <a href="#">/reference/android/graphics/drawable/StateListDrawable.html</a> ), set when pressing down in a view.
int ViewDrawableStates_state_selected	State value for StateListDrawable ( <a href="#">/reference/android/graphics/drawable/StateListDrawable.html</a> ), set when one of its parents) is currently selected.
int ViewDrawableStates_state_window_focused	State value for StateListDrawable ( <a href="#">/reference/android/graphics/drawable/StateListDrawable.html</a> ), set when the window has input focus.
int ViewFlipper_autoStart	When true, automatically start animating Must be a boolean value, either "true" or "false".
int ViewFlipper_flipInterval	This symbol is the offset where the flipInterval ( <a href="#">/reference/android/R.styleable.html#flipInterval</a> ) attribute's value can be ViewFlipper ( <a href="#">/reference/android/R.styleable.html#ViewFlipper</a> ) attribute's value can be
int ViewGroup_Layout_layout_height	Specifies the basic height of the view.
int ViewGroup_Layout_layout_width	Specifies the basic width of the view.
int ViewGroup_MarginLayout_layout_height	Specifies the basic height of the view.
int ViewGroup_MarginLayout_layout_margin	Specifies extra space on the left, top, right and bottom sides of this view.
int ViewGroup_MarginLayout_layout_marginBottom	Specifies extra space on the bottom side of this view.
int ViewGroup_MarginLayout_layout_marginEnd	Specifies extra space on the end side of this view.
int ViewGroup_MarginLayout_layout_marginLeft	Specifies extra space on the left side of this view.
int ViewGroup_MarginLayout_layout_marginRight	Specifies extra space on the right side of this view.
int ViewGroup_MarginLayout_layout_marginStart	Specifies extra space on the start side of this view.
int ViewGroup_MarginLayout_layout_marginTop	Specifies extra space on the top side of this view.
int ViewGroup_MarginLayout_layout_width	Specifies the basic width of the view.
int ViewGroup_addStatesFromChildren	Sets whether this ViewGroup's drawable states also include its children's states.
int ViewGroup_alwaysDrawnWithCache	Defines whether the ViewGroup should always draw its children using their drawing cache or not.
int ViewGroup_animateLayoutChanges	Defines whether changes in layout (caused by adding and removing items) will trigger a LayoutTransition to run.
int ViewGroup_animationCache	Defines whether layout animations should create a drawing cache for their children.
int ViewGroup_clipChildren	Defines whether a child is limited to draw inside of its bounds or not.
int ViewGroup_clipToPadding	Defines whether the ViewGroup will clip its drawing surface so as to exclude padding.
int ViewGroup_descendantFocusability	Defines the relationship between the ViewGroup and its descendants when it receives a key event to take focus.
int ViewGroup_layoutAnimation	Defines the layout animation to use the first time the ViewGroup is laid out.

int ViewGroup_persistentDrawingCache	Defines the persistence of the drawing cache.
int ViewGroup_splitMotionEvents	Sets whether this ViewGroup should split MotionEvents to separate child's event dispatch.
int ViewStub_inflatedId	Overrides the id of the inflated View with this value.
int ViewStub_layout	Supply an identifier for the layout resource to inflate when the ViewStub is forced to do so.
int View_alpha	alpha property of the view, as a value between 0 (completely transparent) and 1 (opaque).
int View_background	A drawable to use as the background.
int View_clickable	Defines whether this view reacts to click events.
int View_contentDescription	Defines text that briefly describes content of the view.
int View_drawingCacheQuality	Defines the quality of translucent drawing caches.
int View_duplicateParentState	When this attribute is set to true, the view gets its drawable state (focused or not) from its direct parent rather than from itself.
int View_fadeScrollbars	Defines whether to fade out scrollbars when they are not in use.
int View_fadingEdge	This attribute is deprecated and will be ignored as of API level 14 (ICE_CREAM_SANDWICH and higher). (/reference/android/os/Build.VERSION_CODES.html#ICE_CREAM_SANDWICH)
int View_fadingEdgeLength	Defines the length of the fading edges.
int View_filterTouchesWhenObscured	Specifies whether to filter touches when the view's window is obscured by another window.
int View.fitsSystemWindows	Boolean internal attribute to adjust view layout based on system windows bar.
int View_focusable	Boolean that controls whether a view can take focus.
int View_focusableInTouchMode	Boolean that controls whether a view can take focus while in touch mode.
int View_hapticFeedbackEnabled	Boolean that controls whether a view should have haptic feedback enabled for long presses.
int View_id	Supply an identifier name for this view, to later retrieve it with View.findViewById(). (/reference/android/view/View.html#findViewById(int)) or Activity.findViewById(). (/reference/android/app/Activity.html#findViewById(int)).
int View_importantForAccessibility	Controls how this View is important for accessibility which is if it fires accessibility events if it is reported to accessibility services that query the screen.
int View_isScrollContainer	Set this if the view will serve as a scrolling container, meaning that it can be a scrollable window so that there will be space for an input method.
int View_keepScreenOn	Controls whether the view's window should keep the screen on while visible.
int View_labelFor	Specifies the id of a view for which this view serves as a label for accessibility.
int View_layerType	Specifies the type of layer backing this view.

int View_layoutDirection	Defines the direction of layout drawing.
int View_longClickable	Defines whether this view reacts to long click events.
int View_minHeight	Defines the minimum height of the view.
int View_minWidth	Defines the minimum width of the view.
int View_nextFocusDown	Defines the next view to give focus to when the next focus is FOCUS_DOWN (/reference/android/view/View.html#FOCUS_DOWN) If the reference refers to not exist or is part of a hierarchy that is invisible, a RuntimeException (/reference/java/lang/RuntimeException.html) will result when the refer
int View_nextFocusForward	Defines the next view to give focus to when the next focus is FOCUS_FORWARD (/reference/android/view/View.html#FOCUS_FORWARD) If the reference ref does not exist or is part of a hierarchy that is invisible, a RuntimeException (/reference/java/lang/RuntimeException.html) will result when the refer
int View_nextFocusLeft	Defines the next view to give focus to when the next focus is FOCUS_LEFT (/reference/android/view/View.html#FOCUS_LEFT).
int View_nextFocusRight	Defines the next view to give focus to when the next focus is FOCUS_RIGHT (/reference/android/view/View.html#FOCUS_RIGHT) If the reference refers to not exist or is part of a hierarchy that is invisible, a RuntimeException (/reference/java/lang/RuntimeException.html) will result when the refer
int View_nextFocusUp	Defines the next view to give focus to when the next focus is FOCUS_UP (/reference/android/view/View.html#FOCUS_UP) If the reference refers to exist or is part of a hierarchy that is invisible, a RuntimeException (/reference/java/lang/RuntimeException.html) will result when the refer
int View_onClick	Name of the method in this View's context to invoke when the view is clicked.
int View_overScrollMode	Defines over-scrolling behavior.
int View_padding	Sets the padding, in pixels, of all four edges.
int View_paddingBottom	Sets the padding, in pixels, of the bottom edge; see padding (/reference/android/R.attr.html#padding).
int View_paddingEnd	Sets the padding, in pixels, of the end edge; see padding (/reference/android/R.attr.html#padding).
int View_paddingLeft	Sets the padding, in pixels, of the left edge; see padding (/reference/android/R.attr.html#padding).
int View_paddingRight	Sets the padding, in pixels, of the right edge; see padding (/reference/android/R.attr.html#padding).
int View_paddingStart	Sets the padding, in pixels, of the start edge; see padding (/reference/android/R.attr.html#padding).
int View_paddingTop	Sets the padding, in pixels, of the top edge; see padding (/reference/android/R.attr.html#padding).
int View_requiresFadingEdge	Defines which edges should be faded on scrolling.
int View_rotation	rotation of the view, in degrees.
int View_rotationX	rotation of the view around the x axis, in degrees.

int View_rotationY	rotation of the view around the y axis, in degrees.
int View_saveEnabled	If unset, no state will be saved for this view when it is being frozen.
int View_scaleX	scale of the view in the x direction.
int View_scaleY	scale of the view in the y direction.
int View_scrollX	The initial horizontal scroll offset, in pixels.
int View_scrollY	The initial vertical scroll offset, in pixels.
int View_scrollbarAlwaysDrawHorizontalTrack	Defines whether the horizontal scrollbar track should always be drawn.
int View_scrollbarAlwaysDrawVerticalTrack	Defines whether the vertical scrollbar track should always be drawn.
int View_scrollbarDefaultDelayBeforeFade	Defines the delay in milliseconds that a scrollbar waits before fade out.
int View_scrollbarFadeDuration	Defines the delay in milliseconds that a scrollbar takes to fade out.
int View_scrollbarSize	Sets the width of vertical scrollbars and height of horizontal scrollbars.
int View_scrollbarStyle	Controls the scrollbar style and position.
int View_scrollbarThumbHorizontal	Defines the horizontal scrollbar thumb drawable.
int View_scrollbarThumbVertical	Defines the vertical scrollbar thumb drawable.
int View_scrollbarTrackHorizontal	Defines the horizontal scrollbar track drawable.
int View_scrollbarTrackVertical	Defines the vertical scrollbar track drawable.
int View_scrollbars	Defines which scrollbars should be displayed on scrolling or not.
int View_soundEffectsEnabled	Boolean that controls whether a view should have sound effects enabled for clicking and touching.
int View_tag	Supply a tag for this view containing a String, to be retrieved later with <code>View.getTag()</code> or searched for with <code>View.findViewWithTag()</code> (/reference/android/view/View.html#findViewWithTag(java.lang.Object))
int View_textAlignment	Defines the alignment of the text.
int View_textDirection	Defines the direction of the text.
int View_transformPivotX	x location of the pivot point around which the view will rotate and scale.
int View_transformPivotY	y location of the pivot point around which the view will rotate and scale.
int View_translationX	translation in x of the view.
int View_translationY	translation in y of the view.
int View_verticalScrollbarPosition	Determines which side the vertical scroll bar should be placed on.
int View_visibility	Controls the initial visibility of the view.

int VolumePreference_streamType	Different audio stream types.
int WallpaperPreviewInfo_staticWallpaperPreview	A resource id of a static drawable.
int Wallpaper_author	Name of the author of this component, e.g.
int Wallpaper_description	Short description of the component's purpose or behavior.
int Wallpaper_settingsActivity	Component name of an activity that allows the user to modify the settings
int Wallpaper_thumbnail	Reference to a the wallpaper's thumbnail bitmap.
int WindowAnimation_activityCloseEnterAnimation	When closing the current activity, this is the animation that is run on the next activity (which is entering the screen).
int WindowAnimation_activityCloseExitAnimation	When closing the current activity, this is the animation that is run on the current activity (which is exiting the screen).
int WindowAnimation_activityOpenEnterAnimation	When opening a new activity, this is the animation that is run on the next activity (which is entering the screen).
int WindowAnimation_activityOpenExitAnimation	When opening a new activity, this is the animation that is run on the previous activity (which is exiting the screen).
int WindowAnimation_taskCloseEnterAnimation	When closing the last activity of a task, this is the animation that is run on the next task (which is entering the screen).
int WindowAnimation_taskCloseExitAnimation	When opening an activity in a new task, this is the animation that is run on the old task (which is exiting the screen).
int WindowAnimation_taskOpenEnterAnimation	When opening an activity in a new task, this is the animation that is run on the new task (which is entering the screen).
int WindowAnimation_taskOpenExitAnimation	When opening an activity in a new task, this is the animation that is run on the old task (which is exiting the screen).
int WindowAnimation_taskToBackEnterAnimation	When sending the current task to the background, this is the animation that is run on the activity of the task behind it (which is entering the screen).
int WindowAnimation_taskToBackExitAnimation	When sending the current task to the background, this is the animation that is run on the activity of the current task (which is exiting the screen).
int WindowAnimation_taskToFrontEnterAnimation	When bringing an existing task to the foreground, this is the animation that is run on the activity of the task being brought to the foreground (which is entering the screen).
int WindowAnimation_taskToFrontExitAnimation	When bringing an existing task to the foreground, this is the animation that is run on the current foreground activity (which is exiting the screen).
int WindowAnimation_wallpaperCloseEnterAnimation	When opening a new activity that hides the wallpaper, while currently showing this is the animation that is run on the new activity (which is entering the screen).
int WindowAnimation_wallpaperCloseExitAnimation	When opening a new activity that hides the wallpaper, while currently showing this is the animation that is run on the old wallpaper activity (which is exiting the screen).
int WindowAnimation_wallpaperIntraCloseEnterAnimation	When closing a foreground activity that is on top of the wallpaper when the wallpaper is also on top of the wallpaper, this is the animation that is run on the previous activity (which is entering the screen).
	When closing a foreground activity that is on top of the wallpaper when the wallpaper is also on top of the wallpaper, this is the animation that is run on the previous activity (which is exiting the screen).

int WindowAnimation_wallpaperIntraCloseExitAnimation	also on top of the wallpaper, this is the animation that is run on the current activity when exiting the screen).
int WindowAnimation_wallpaperIntraOpenEnterAnimation	When opening a new activity that is on top of the wallpaper when the current activity is on top of the wallpaper, this is the animation that is run on the new activity (which is on the screen).
int WindowAnimation_wallpaperIntraOpenExitAnimation	When opening a new activity that is on top of the wallpaper when the current activity is on top of the wallpaper, this is the animation that is run on the current activity (which is on the screen).
int WindowAnimation_wallpaperOpenEnterAnimation	When opening a new activity that shows the wallpaper, while currently not showing the wallpaper, this is the animation that is run on the new wallpaper activity (which is on the screen).
int WindowAnimation_wallpaperOpenExitAnimation	When opening a new activity that shows the wallpaper, while currently not showing the wallpaper, this is the animation that is run on the current activity (which is on the screen).
int WindowAnimation_windowEnterAnimation	The animation used when a window is being added.
int WindowAnimation_windowExitAnimation	The animation used when a window is being removed.
int WindowAnimation_windowHideAnimation	The animation used when a window is going from VISIBLE to INVISIBLE.
int WindowAnimation_windowShowAnimation	The animation used when a window is going from INVISIBLE to VISIBLE.
int Window_backgroundDimAmount	Default background dim amount when a menu, dialog, or something similar is displayed.
int Window_backgroundDimEnabled	Control whether dimming behind the window is enabled.
int Window_textColor	Color of text (usually same as colorForeground).
int Window_windowActionBar	Flag indicating whether this window should have an Action Bar in place of its own.
int Window_windowActionBarOverlay	Flag indicating whether this window's Action Bar should overlay application content.
int Window_windowActionModeOverlay	Flag indicating whether action modes should overlay window content when reserved space for their UI (such as an Action Bar).
int Window_windowAnimationStyle	Reference to a style resource holding the set of window animations to use, the attributes defined by <code>WindowAnimation</code> ( <a href="#">/reference/android/R.styleable.html#WindowAnimation</a> ).
int Window_windowBackground	Drawable to use as the overall window background.
int Window_windowCloseOnTouchOutside	Control whether a container should automatically close itself if the user touches outside it.
int Window_windowContentOverlay	This Drawable is overlaid over the foreground of the Window's content area, including the shadow below the title.
int Window_windowDisablePreview	Flag allowing you to disable the preview animation for a window.
int Window_windowEnableSplitTouch	Flag indicating that this window should allow touches to be split across multiple windows (such as when the window also supports split touch).
int Window_windowFrame	Drawable to use as a frame around the window.
int Window_windowFullscreen	Flag indicating whether this window should fill the entire screen.
int Window_windowIsFloating	Flag indicating whether this is a floating window.

int Window_windowIsTranslucent	Flag indicating whether this is a translucent window.
int Window_windowMinWidthMajor	The minimum width the window is allowed to be, along the major axis of its parent.
int Window_windowMinWidthMinor	The minimum width the window is allowed to be, along the minor axis of its parent.
int Window_windowNoDisplay	Flag indicating that this window should not be displayed at all.
int Window_windowNoTitle	Flag indicating whether there should be no title on this window.
int Window_windowShowWallpaper	Flag indicating that this window's background should be the user's current wallpaper.
int Window_windowSoftInputMode	Defines the default soft input state that this window would like when it is created.
<b>Fields</b>	
public static final int[] AbsListView	Attributes that can be used with a AbsListView.
public static final int[] AbsSpinner	Attributes that can be used with a AbsSpinner.
public static final int[] AbsoluteLayout_Layout	Attributes that can be used with a AbsoluteLayout_Layout.
public static final int[] AccelerateInterpolator	Attributes that can be used with a AccelerateInterpolator.
public static final int[] AccessibilityService	Use accessibility-service as the root tag of the XML resource that describes an AccessibilityService service, which is referenced from its SERVICE_META_DATA meta-data entry.
public static final int[] AccountAuthenticator	Use account-authenticator as the root tag of the XML resource that describes an account authenticator.
public static final int[] ActionBar	Attributes used to style the Action Bar.
public static final int[] ActionBar_LayoutParams	Attributes that can be used with a ActionBar_LayoutParams.
public static final int[] ActionMenuItemView	Attributes that can be used with a ActionMenuItemView.
public static final int[] ActionMode	Attributes that can be used with a ActionMode.
public static final int[] ActivityChooserView	Attributes for a ActivityChooserView.
public static final int[] AdapterViewAnimator	Attributes that can be used with a AdapterViewAnimator.
public static final int[] AdapterViewFlipper	Attributes that can be used with a AdapterViewFlipper.
public static final int[] AlertDialog	The set of attributes that describe a AlertDialog's theme.
public static final int[] AlphaAnimation	Attributes that can be used with a AlphaAnimation.
public static final int[] AnalogClock	Attributes that can be used with a AnalogClock.
public static final int[] AndroidManifest	The manifest tag is the root of an AndroidManifest.xml file, describing the contents of an Android package (.apk) file.
public static final int[] AndroidManifestAction	Attributes that can be supplied in an AndroidManifest.xml action tag, a child of the intent-filter tag.
public static final int[] AndroidManifestActivity	The activity tag declares an Activity class that is available as part of the package's application components, implementing a part of the application's user interface.
public static final int[] AndroidManifestActivityAlias	The activity-alias tag declares a new name for an existing activity tag.
public static final int[] AndroidManifestApplication	The application tag describes application-level components contained in the package, as well as general application-wide settings.

public static final int[] AndroidManifestCategory	application attributes. Attributes that can be supplied in an <code>AndroidManifest.xml</code> category tag, a child of the <code>intent-filter</code> tag.
public static final int[] AndroidManifestCompatibleScreensScreen	Attributes that can be supplied in an <code>AndroidManifest.xml</code> screen tag, a child of <code>compatible-screens</code> , which is itself a child of the root <code>manifest</code> tag.
public static final int[] AndroidManifestData	Attributes that can be supplied in an <code>AndroidManifest.xml</code> data tag, a child of the <code>intent-filter</code> tag, describing the types of data that match.
public static final int[] AndroidManifestGrantUriPermission	Attributes that can be supplied in an <code>AndroidManifest.xml</code> grant-uri-permission tag, a child of the provider tag, describing a specific URI path that can be granted as a permission.
public static final int[] AndroidManifestInstrumentation	Attributes that can be supplied in an <code>AndroidManifest.xml</code> instrumentation tag, a child of the root <code>manifest</code> tag.
public static final int[] AndroidManifestIntentFilter	The <code>intent-filter</code> tag is used to construct an <code>IntentFilter</code> object that will be used to determine which component can handle a particular Intent that has been given to the system.
public static final int[] AndroidManifestMetaData	The <code>meta-data</code> tag is used to attach additional arbitrary data to an application component.
public static final int[] AndroidManifestOriginalPackage	Private tag to declare the original package name that this package is based on.
public static final int[] AndroidManifestPackageVerifier	Attributes relating to a package verifier.
public static final int[] AndroidManifestPathPermission	Attributes that can be supplied in an <code>AndroidManifest.xml</code> path-permission tag, a child of the provider tag, describing a permission that allows access to a specific path in the provider.
public static final int[] AndroidManifestPermission	The <code>permission</code> tag declares a security permission that can be used to control access from other packages to specific components or features in your package (or other packages).
public static final int[] AndroidManifestPermissionGroup	The <code>permission-group</code> tag declares a logical grouping of related permissions.
public static final int[] AndroidManifestPermissionTree	The <code>permission-tree</code> tag declares the base of a tree of permission values: it declares that this package has ownership of the given permission name, as well as all names underneath it (separated by ':').
public static final int[] AndroidManifestProtectedBroadcast	Private tag to declare system protected broadcast actions.
public static final int[] AndroidManifestProvider	The <code>provider</code> tag declares a <code>ContentProvider</code> class that is available as part of the package's application components, supplying structured access to data managed by the application.
public static final int[] AndroidManifestReceiver	The <code>receiver</code> tag declares a <code>BroadcastReceiver</code> class that is available as part of the package's application components, allowing the application to receive actions or data broadcast by other applications even if it is not currently running.
public static final int[] AndroidManifestService	The <code>service</code> tag declares a <code>Service</code> class that is available as part of the package's application components, implementing long-running background operations or a rich communication API that can be called by other packages.
	The <code>supports-screens</code> specifies the

public static final int[] AndroidManifestSupportsScreens	screen dimensions an application supports.
public static final int[] AndroidManifestUsesConfiguration	The <code>uses-configuration</code> tag specifies a specific hardware configuration value used by the application.
public static final int[] AndroidManifestUsesFeature	The <code>uses-feature</code> tag specifies a specific feature used by the application.
public static final int[] AndroidManifestUsesLibrary	The <code>uses-libraries</code> specifies a shared library that this package requires to be linked against.
public static final int[] AndroidManifestUsesPermission	The <code>uses-permission</code> tag requests a <code>&lt;permission&gt;</code> that the containing package must be granted in order for it to operate correctly.
public static final int[] AndroidManifestUsesSdk	The <code>uses-sdk</code> tag describes the SDK features that the containing package must be running on to operate correctly.
public static final int[] AnimatedRotateDrawable	Attributes that can be used with a <code>AnimatedRotateDrawable</code> .
public static final int[] Animation	Attributes that can be used with a <code>Animation</code> .
public static final int[] AnimationDrawable	Drawable used to render several animated frames.
public static final int[] AnimationDrawableItem	Represents a single frame inside an <code>AnimationDrawable</code> .
public static final int[] AnimationSet	Attributes that can be used with a <code>AnimationSet</code> .
public static final int[] Animator	Attributes that can be used with a <code>Animator</code> .
public static final int[] AnimatorSet	Attributes that can be used with a <code>AnimatorSet</code> .
public static final int[] AnticipateInterpolator	Attributes that can be used with a <code>AnticipateInterpolator</code> .
public static final int[] AnticipateOvershootInterpolator	Attributes that can be used with a <code>AnticipateOvershootInterpolator</code> .
public static final int[] AppWidgetProviderInfo	Use <code>appwidget-provider</code> as the root tag of the XML resource that describes an <code>AppWidget</code> provider.
public static final int[] AutoCompleteTextView	Attributes that can be used with a <code>AutoCompleteTextView</code> .
public static final int[] BitmapDrawable	Drawable used to draw bitmaps.
public static final int[] Button	Attributes that can be used with a <code>Button</code> .
public static final int[] CalendarView	Attributes that can be used with a <code>CalendarView</code> .
public static final int[] CheckBoxPreference	Base attributes available to <code>CheckBoxPreference</code> .
public static final int[] CheckedTextView	Attributes that can be used with a <code>CheckedTextView</code> .
public static final int[] Chronometer	Attributes that can be used with a <code>Chronometer</code> .
public static final int[] ClipDrawable	Attributes that can be used with a <code>ClipDrawable</code> .
public static final int[] ColorDrawable	Drawable used to draw a single color.
public static final int[] CompoundButton	Attributes that can be used with a <code>CompoundButton</code> .
public static final int[] ContactsDataKind	Maps a specific contact data MIME-type to styling information.
public static final int[] CycleInterpolator	Attributes that can be used with a <code>CycleInterpolator</code> .
public static final int[] DatePicker	Attributes that can be used with a <code>DatePicker</code> .
public static final int[] DecelerateInterpolator	Attributes that can be used with a <code>DecelerateInterpolator</code> .
public static final int[] DeviceAdmin	Use <code>device-admin</code> as the root tag of the XML resource that describes a <code>DeviceAdminReceiver</code> , which is referenced from its <code>DEVICE_ADMIN_META_DATA</code> meta-data

public static final int[] DialogPreference	entry.
public static final int[] Drawable	Base attributes available to DialogPreference.
public static final int[] DrawableCorners	Base attributes that are available to all Drawable objects.
public static final int[] DrawableStates	Describes the corners for the rectangle shape of a GradientDrawable.
public static final int[] Dream	Drawable states.
public static final int[] EditText	Use dream as the root tag of the XML resource that describes an DreamService, which is referenced from its DREAM_META_DATA meta-data entry.
public static final int[] ExpandableListChildIndicatorState	Attributes that can be used with a EditText.
public static final int[] ExpandableListGroupIndicatorState	State array representing an expandable list child's indicator.
public static final int[] ExpandableListView	State array representing an expandable list group's indicator.
public static final int[] Extra	Attributes that can be used with a ExpandableListView.
public static final int[] Fragment	An extra data value to place into a an extra/name value pair held in a Bundle, as per Bundle.
public static final int[] FragmentAnimation	Attributes that can be used with <fragment> tags inside of the layout of an Activity.
public static final int[] FragmentBreadCrumbs	Fragment animation class attributes.
public static final int[] FrameLayout	Attributes that can be used with <FragmentBreadCrumbs> tags.
public static final int[] FrameLayout_Layout	Attributes that can be used with a FrameLayout.
public static final int[] Gallery	Attributes that can be used with a FrameLayout_Layout.
public static final int[] GestureOverlayView	Attributes that can be used with a Gallery.
public static final int[] GlowPadView	GestureOverlayView specific attributes.
public static final int[] GradientDrawable	Attributes that can be used with a GlowPadView.
public static final int[] GradientDrawableGradient	Drawable used to render a geometric shape, with a gradient or a solid color.
public static final int[] GradientDrawablePadding	Used to describe the gradient used to fill the shape of a GradientDrawable.
public static final int[] GradientDrawableSize	Used to specify the optional padding of a GradientDrawable.
public static final int[] GradientDrawableSolid	Used to specify the size of the shape for GradientDrawable.
public static final int[] GradientDrawableStroke	Used to fill the shape of GradientDrawable with a solid color.
public static final int[] GridLayout	Used to describe the optional stroke of a GradientDrawable.
public static final int[] GridLayoutAnimation	Attributes that can be used with a GridLayout.
public static final int[] GridLayout_Layout	Attributes that can be used with a GridLayoutAnimation.
public static final int[] GridView	Attributes that can be used with a GridLayout_Layout.
public static final int[] HorizontalScrollView	Attributes that can be used with a GridView.
public static final int[] Icon	Attributes that can be used with a HorizontalScrollView.
public static final int[] IconDefault	Attributes that can be used with a Icon.
public static final int[] IconMenuView	Attributes that can be used with a IconDefault.
public static final int[] ImageSwitcher	Attributes that can be used with a IconMenuView.
public static final int[] ImageView	Attributes that can be used with a ImageSwitcher.

<code>public static final int[] InputExtras</code>	<code>ImageView</code> . An <code>input-extras</code> is a container for extra data to supply to an input method.
<code>public static final int[] InputMethod</code>	Use <code>input-method</code> as the root tag of the XML resource that describes an <code>InputMethod</code> service, which is referenced from its <code>SERVICE_META_DATA</code> meta-data entry.
<code>public static final int[] InputMethodService</code>	Attributes that can be used with a <code>InputMethodService</code> .
<code>public static final int[] InputMethod_Subtype</code>	This is the subtype of <code>InputMethod</code> .
<code>public static final int[] InsetDrawable</code>	Attributes that can be used with a <code>InsetDrawable</code> .
<code>public static final int[] Intent</code>	Declaration of an <code>Intent</code> object in XML.
<code>public static final int[] IntentCategory</code>	A category to add to an <code>Intent</code> , as per <code>Intent.addCategory()</code> .
<code>public static final int[] Keyboard</code>	Attributes that can be used with a <code>Keyboard</code> .
<code>public static final int[] KeyboardLayout</code>	Use <code>keyboard-layouts</code> as the root tag of the XML resource that describes a collection of keyboard layouts provided by an application.
<code>public static final int[] KeyboardView</code>	Attributes that can be used with a <code>KeyboardView</code> .
<code>public static final int[] KeyboardViewPreviewState</code>	Attributes that can be used with a <code>KeyboardViewPreviewState</code> .
<code>public static final int[] Keyboard_Key</code>	Attributes that can be used with a <code>Keyboard_Key</code> .
<code>public static final int[] Keyboard_Row</code>	Attributes that can be used with a <code>Keyboard_Row</code> .
<code>public static final int[] KeyguardGlowStripView</code>	Attributes that can be used with a <code>KeyguardGlowStripView</code> .
<code>public static final int[] KeyguardSecurityViewFlipper_Layout</code>	Attributes that can be used with a <code>KeyguardSecurityViewFlipper_Layout</code> .
<code>public static final int[] LayerDrawable</code>	Drawable used to render several drawables stacked on top of each other.
<code>public static final int[] LayerDrawableItem</code>	Describes an item (or child) of a <code>LayerDrawable</code> .
<code>public static final int[] LayoutAnimation</code>	Attributes that can be used with a <code>LayoutAnimation</code> .
<code>public static final int[] LevelListDrawableItem</code>	Attributes that can be used with a <code>LevelListDrawableItem</code> .
<code>public static final int[] LinearLayout</code>	Attributes that can be used with a <code>LinearLayout</code> .
<code>public static final int[] LinearLayout_Layout</code>	Attributes that can be used with a <code>LinearLayout_Layout</code> .
<code>public static final int[] ListPreference</code>	Base attributes available to <code>ListPreference</code> .
<code>public static final int[] ListView</code>	Attributes that can be used with a <code>ListView</code> .
<code>public static final int[] LockPatternView</code>	Attributes that can be used with a <code>LockPatternView</code> .
<code>public static final int[] MapView</code>	The set of attributes for a <code>MapView</code> .
<code>public static final int[] MediaRouteButton</code>	Attributes that can be used with a <code>MediaRouteButton</code> .
<code>public static final int[] Menu</code>	Base attributes that are available to all <code>Menu</code> objects.
<code>public static final int[] MenuGroup</code>	Base attributes that are available to all groups.
<code>public static final int[] MenuItem</code>	Base attributes that are available to all <code>Item</code> objects.
<code>public static final int[] MenuItemCheckedFocusedState</code>	State array representing a menu item that is currently focused and checked.
<code>public static final int[] MenuItemCheckedState</code>	State array representing a menu item that is currently checked.
<code>public static final int[] MenuItemUncheckedFocusedState</code>	State array representing a menu item that is focused and checkable but is not currently checked.
<code>public static final int[] MenuItemUncheckableState</code>	State array representing a menu item

public static final int[] MenuItemUncheckedState	that is checkable but is not currently checked.
public static final int[] MenuView	Attributes that can be used with a MenuView.
public static final int[] MipmapDrawableItem	Attributes that can be used with a MipmapDrawableItem.
public static final int[] MultiPaneChallengeLayout	Attributes that can be used with a MultiPaneChallengeLayout.
public static final int[] MultiPaneChallengeLayout_Layout	Attributes that can be used with a MultiPaneChallengeLayout_Layout.
public static final int[] MultiSelectListPreference	Attributes that can be used with a MultiSelectListPreference.
public static final int[] MultiWaveView	Attributes that can be used with a MultiWaveView.
public static final int[] NinePatchDrawable	Drawable used to draw 9-patches.
public static final int[] NumPadKey	Attributes that can be used with a NumPadKey.
public static final int[] OvershootInterpolator	Attributes that can be used with a OvershootInterpolator.
public static final int[] PagedView	PagedView specific attributes.
public static final int[] Pointer	Attributes that can be used with a Pointer.
public static final int[] PointerIcon	Attributes that can be used with a PointerIcon.
public static final int[] PopupWindow	Attributes that can be used with a PopupWindow.
public static final int[] PopupWindowBackgroundState	Attributes that can be used with a PopupWindowBackgroundState.
public static final int[] Preference	Base attributes available to Preference.
public static final int[] PreferenceFrameLayout	Attributes that can be used with a PreferenceFrameLayout.
public static final int[] PreferenceFrameLayout_Layout	Attributes that can be used with a PreferenceFrameLayout_Layout.
public static final int[] PreferenceGroup	Base attributes available to PreferenceGroup.
public static final int[] PreferenceHeader	Attribute for a header describing the item shown in the top-level list from which the selects the set of preference to dig in to. <i>This field is deprecated. Not used by the framework.</i>
public static final int[] ProgressBar	Attributes that can be used with a PropertyAnimator.
public static final int[] PropertyAnimator	Attributes that can be used with a QuickContactBadge.
public static final int[] QuickContactBadge	Attributes that can be used with a RadioGroup.
public static final int[] RadioGroup	Attributes that can be used with a RatingBar.
public static final int[] RatingBar	Use <code>recognition-service</code> as the root tag of the XML resource that describes a <code>RecognitionService</code> , which is reference from its <code>SERVICE_META_DATA</code> meta-data entry.
public static final int[] RecognitionService	Attributes that can be used with a RelativeLayout.
public static final int[] RelativeLayout	Attributes that can be used with a RelativeLayout_Layout.
public static final int[] RelativeLayout_Layout	Base attributes available to RingtonePreference.
public static final int[] RingtonePreference	Attributes that can be used with a RotateAnimation.
public static final int[] RotateAnimation	Drawable used to rotate another drawable.
public static final int[] RotateDrawable	Attributes that can be used with a ScaleAnimation.
public static final int[] ScaleAnimation	Attributes that can be used with a ScaleDrawable.
public static final int[] ScaleDrawable	Attributes that can be used with a ScrollView.
public static final int[] ScrollView	

public static final int[] SearchView	Attributes that can be used with a SearchView.
public static final int[] Searchable	Searchable activities and applications must provide search configuration information in an XML file, typically called searchable.xml.
public static final int[] SearchableActionKey	In order to process special action keys during search, you must define them using one or more "ActionKey" elements in your Searchable metadata.
public static final int[] SeekBar	Attributes that can be used with a SeekBar.
public static final int[] SelectionModeDrawables	Attributes that can be used with a SelectionModeDrawables.
public static final int[] ShapeDrawable	Drawable used to draw shapes.
public static final int[] ShapeDrawablePadding	Defines the padding of a ShapeDrawable.
public static final int[] SizeAdaptiveLayout	Attributes that can be used with a SizeAdaptiveLayout.
public static final int[] SizeAdaptiveLayout_Layout	Attributes that can be used with a SizeAdaptiveLayout_Layout.
public static final int[] SlidingChallengeLayout_Layout	Attributes that can be used with a SlidingChallengeLayout_Layout.
public static final int[] SlidingDrawer	SlidingDrawer specific attributes.
public static final int[] SlidingTab	Attributes that can be used with a SlidingTab.
public static final int[] SpellChecker	Use <code>spell-checker</code> as the root tag of the XML resource that describes an SpellCheckerService service, which is referenced from its <code>SERVICE_META_DATA</code> meta-data entry.
public static final int[] SpellChecker_Subtype	This is the subtype of the spell checker.
public static final int[] Spinner	Attributes that can be used with a Spinner.
public static final int[] StackView	Attributes that can be used with a StackView.
public static final int[] StateListDrawable	Drawable used to render several states.
public static final int[] Storage	Attributes that can be used with a Storage.
public static final int[] SuggestionSpan	Attributes that can be used with a SuggestionSpan.
public static final int[] Switch	Attributes that can be used with a Switch.
public static final int[] SwitchPreference	Attributes that can be used with a SwitchPreference.
public static final int[] SyncAdapter	Use <code>account-authenticator</code> as the root tag of the XML resource that describes an account authenticator.
public static final int[] TabWidget	Attributes that can be used with a TabWidget.
public static final int[] TableLayout	Attributes that can be used with a TableLayout.
public static final int[] TableRow	Attributes that can be used with a TableRow.
public static final int[] TableRow_Cell	Attributes that can be used with a TableRow_Cell.
public static final int[] TextAppearance	Attributes that can be used with a TextAppearance.
public static final int[] TextClock	Attributes that can be used with a TextClock.
public static final int[] TextSwitcher	Attributes that can be used with a TextSwitcher.
public static final int[] TextToSpeechEngine	Use <code>tts-engine</code> as the root tag of the XML resource that describes a text to speech engine implemented as a subclass of TextToSpeechService.
public static final int[] TextView	Attributes that can be used with a TextView.
public static final int[] TextViewAppearance	Attributes that can be used with a TextViewAppearance.

<code>public static final int[] TextViewMultiLineBackgroundState</code>	Attributes that can be used with a <code>TextViewMultiLineBackgroundState</code> . These are the standard attributes that make up a complete theme.
<code>public static final int[] Theme</code>	Attributes that can be used with a <code>TimePicker</code> .
<code>public static final int[] TimePicker</code>	Attributes that can be used with a <code>ToggleButton</code> .
<code>public static final int[] ToggleButton</code>	Attributes that can be used with a <code>TranslateAnimation</code> .
<code>public static final int[] TranslateAnimation</code>	Attributes that can be used with a <code>TwoLineListItem</code> .
<code>public static final int[] TwoLineListItem</code>	Attributes that can be used with a <code>VerticalSlider_Layout</code> .
<code>public static final int[] VerticalSlider_Layout</code>	Attributes that can be used with a <code>View</code> or any of its subclasses.
<code>public static final int[] View</code>	Attributes that can be used with a <code>ViewAnimator</code> .
<code>public static final int[] ViewAnimator</code>	Attributes that can be used with a <code>ViewDrawableStates</code> .
<code>public static final int[] ViewDrawableStates</code>	Attributes that can be used with a <code>ViewFlipper</code> .
<code>public static final int[] ViewFlipper</code>	Attributes that can be used with a <code>ViewGroup</code> or any of its subclasses.
<code>public static final int[] ViewGroup</code>	This is the basic set of layout attributes that are common to all layout managers.
<code>public static final int[] ViewGroup_Layout</code>	This is the basic set of layout attributes for layout managers that wish to place margins around their child views.
<code>public static final int[] ViewGroup.MarginLayout</code>	A <code>ViewStub</code> lets you lazily include other XML layouts inside your application at runtime.
<code>public static final int[] ViewStub</code>	Attributes that can be used with a <code>ViewSwitcher</code> .
<code>public static final int[] ViewSwitcher</code>	Base attributes available to <code>VolumePreference</code> .
<code>public static final int[] VolumePreference</code>	Use <code>wallpaper</code> as the root tag of the XML resource that describes an <code>WallpaperService</code> , which is referenced from its <code>SERVICE_META_DATA</code> meta-data entry.
<code>public static final int[] Wallpaper</code>	Use <code>wallpaper-preview</code> as the root tag of the XML resource that describes a <code>wallpaper preview</code> .
<code>public static final int[] WallpaperPreviewInfo</code>	The set of attributes that describe a Windows's theme.
<code>public static final int[] Window</code>	Window animation class attributes.
<code>public static final int[] WindowAnimation</code>	

#### Public Constructors

`R.styleable()`  
**Inherited Methods** [Expand]  
 ► From class `java.lang.Object`

## Constants

---

### `public static final int AbsListView_cacheColorHint`

Indicates that this list will always be drawn on top of solid, single-color opaque background. This allows the list to optimize drawing.

Must be a color value, in the form "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [cacheColorHint](#)  
[\(/reference/android/R.attr.html#cacheColorHint\)](#).

Constant Value: 6 (0x00000006)

## `public static final int AbsListView_choiceMode`

Defines the choice behavior for the view. By default, lists do not have any choice behavior. By setting the choiceMode to singleChoice, the list allows up to one item to be in a chosen state. By setting the choiceMode to multipleChoice, the list allows any number of items to be chosen. Finally, by setting the choiceMode to multipleChoiceModal the list allows any number of items to be chosen in a special selection mode. The application will supply a [AbsListView.MultiChoiceModeListener](#) ([/reference/android/widget/AbsListView.MultiChoiceModeListener.html](#)) using `setMultiChoiceModeListener(AbsListView.MultiChoiceModeListener)` ([/reference/android/widget/AbsListView.html#setMultiChoiceModeListener\(android.widget.AbsListView.MultiChoiceModeListener\)](#)) to control the selection mode. This uses the [ActionMode](#) ([/reference/android/view/ActionMode.html](#)) API.

Must be one of the following constant values.

Constant	Value	Description
none	0	Normal list that does not indicate choices.
singleChoice	1	The list allows up to one choice.
multipleChoice	2	The list allows multiple choices.
multipleChoiceModal	3	The list allows multiple choices in a custom selection mode.

This corresponds to the global attribute resource symbol [choiceMode](#) ([/reference/android/R.attr.html#choiceMode](#)).

Constant Value: 7 (0x00000007)

## `public static final int AbsListView_drawSelectorOnTop`

When set to true, the selector will be drawn over the selected item. Otherwise the selector is drawn behind the selected item. The default value is false.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [drawSelectorOnTop](#) ([/reference/android/R.attr.html#drawSelectorOnTop](#)).

Constant Value: 1 (0x00000001)

## `public static final int AbsListView_fastScrollAlwaysVisible`

When set to true, the list will always show the fast scroll interface. This setting implies fastScrollEnabled.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [fastScrollAlwaysVisible](#) ([/reference/android/R.attr.html#fastScrollAlwaysVisible](#)).

Constant Value: 10 (0x0000000a)

## `public static final int AbsListView_fastScrollEnabled`

Enables the fast scroll thumb that can be dragged to quickly scroll through the list.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [fastScrollEnabled](#) ([/reference/android/R.attr.html#fastScrollEnabled](#)).

Constant Value: 8 (0x00000008)

## `public static final int AbsListView_listSelector`

Drawable used to indicate the currently selected item in the list.

May be a reference to another resource, in the form "@+[ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [listSelector](#)  
(/reference/android/R.attr.html#listSelector).

Constant Value: 0 (0x00000000)

#### public static final int **AbsListView\_scrollingCache**

When set to true, the list uses a drawing cache during scrolling. This makes the rendering faster but uses more memory. The default value is true.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [scrollingCache](#)  
(/reference/android/R.attr.html#scrollingCache).

Constant Value: 3 (0x00000003)

#### public static final int **AbsListView\_smoothScrollbar**

When set to true, the list will use a more refined calculation method based on the pixels height of the items visible on screen. This property is set to true by default but should be set to false if your adapter will display items of varying heights. When this property is set to true and your adapter displays items of varying heights, the scrollbar thumb will change size as the user scrolls through the list. When set to false, the list will use only the number of items in the adapter and the number of items visible on screen to determine the scrollbar's properties.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [smoothScrollbar](#)  
(/reference/android/R.attr.html#smoothScrollbar).

Constant Value: 9 (0x00000009)

#### public static final int **AbsListView\_stackFromBottom**

Used by ListView and GridView to stack their content from the bottom.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [stackFromBottom](#)  
(/reference/android/R.attr.html#stackFromBottom).

Constant Value: 2 (0x00000002)

#### public static final int **AbsListView\_textFilterEnabled**

When set to true, the list will filter results as the user types. The List's adapter must support the Filterable interface for this to work.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [textFilterEnabled](#)  
(/reference/android/R.attr.html#textFilterEnabled).

Constant Value: 4 (0x00000004)

#### public static final int **AbsListView\_transcriptMode**

Sets the transcript mode for the list. In transcript mode, the list scrolls to the bottom to make new items visible when they are added.

Must be one of the following constant values.

Constant	Value	Description
disabled	0	Disables transcript mode. This is the default value. The list will automatically scroll to the bottom when a data set change

normal 1 notification is received and only if the last item is already visible on screen.

alwaysScroll 2 The list will automatically scroll to the bottom, no matter what items are currently visible.

This corresponds to the global attribute resource symbol [transcriptMode](#)  
[\(/reference/android/R.attr.html#transcriptMode\)](#).

Constant Value: 5 (0x00000005)

#### **public static final int AbsSpinner\_entries**

Reference to an array resource that will populate the Spinner. For static content, this is simpler than populating the Spinner programmatically.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [entries](#)  
[\(/reference/android/R.attr.html#entries\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int AbsoluteLayout\_Layout\_layout\_x**

This symbol is the offset where the [layout\\_x](#) ([\(/reference/android/R.attr.html#layout\\_x\)](#)) attribute's value can be found in the [AbsoluteLayout\\_Layout](#)  
[\(/reference/android/R.styleable.html#AbsoluteLayout\\_Layout\)](#) array.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

Constant Value: 0 (0x00000000)

#### **public static final int AbsoluteLayout\_Layout\_layout\_y**

This symbol is the offset where the [layout\\_y](#) ([\(/reference/android/R.attr.html#layout\\_y\)](#)) attribute's value can be found in the [AbsoluteLayout\\_Layout](#)  
[\(/reference/android/R.styleable.html#AbsoluteLayout\\_Layout\)](#) array.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

Constant Value: 1 (0x00000001)

#### **public static final int AccelerateInterpolator\_factor**

This is the amount of deceleration to add when easing in.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [factor](#)  
[\(/reference/android/R.attr.html#factor\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int AccessibilityService\_accessibilityEventTypes**

The event types this service would like to receive as specified in [AccessibilityEvent](#)  
[\(/reference/android/view/accessibility/AccessibilityEvent.html\)](#). This setting can be changed at runtime by calling [android.accessibilityservice.AccessibilityService.setServiceInfo\(android.accessibilityservice.AccessibilityServiceInfo\)](#)  
[\(/reference/android/accessibilityservice/AccessibilityService.html#setServiceInfo\(android.accessibilityservice.AccessibilityServiceInfo\)\)](#).

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
typeViewClicked	0x00000001	Receives <code>TYPE_VIEW_CLICKED</code> events.
typeViewLongClicked	0x00000002	Receives <code>TYPE_VIEW_LONG_CLICKED</code> events.
typeViewSelected	0x00000004	Receives <code>TYPE_VIEW_SELECTED</code> events.
typeViewFocused	0x00000008	Receives <code>TYPE_VIEW_FOCUSED</code> events.
typeViewTextChanged	0x00000010	Receives <code>TYPE_VIEW_TEXT_CHANGED</code> events.
typeWindowStateChanged	0x00000020	Receives <code>TYPE_WINDOW_STATE_CHANGED</code> events. Receives
typeNotificationStateChanged	0x00000040	<code>TYPE_NOTIFICATION_STATE_CHANGED</code> events.
typeViewHoverEnter	0x00000080	Receives <code>TYPE_VIEW_HOVER_ENTER</code> events.
typeViewHoverExit	0x00000100	Receives <code>TYPE_VIEW_HOVER_EXIT</code> events. Receives
typeTouchExplorationGestureStart	0x00000200	<code>TYPE_TOUCH_EXPLORATION_GESTURE_START</code> events. Receives
typeTouchExplorationGestureEnd	0x00000400	<code>TYPE_TOUCH_EXPLORATION_GESTURE_END</code> events.
typeWindowContentChanged	0x00000800	Receives <code>TYPE_WINDOW_CONTENT_CHANGED</code> events.
typeViewScrolled	0x00001000	Receives <code>TYPE_VIEW_SCROLLED</code> events. Receives
typeViewTextSelectionChanged	0x00002000	<code>TYPE_VIEW_TEXT_SELECTION_CHANGED</code> events.
typeAllMask	0xffffffff	Receives <code>TYPES_ALL_MASK</code> i.e. all events.

This corresponds to the global attribute resource symbol `accessibilityEventTypes`  
`(/reference/android/R.attr.html#accessibilityEventTypes)`.

Constant Value: 2 (0x00000002)

#### public static final int AccessibilityService\_accessibilityFeedbackType

The feedback types this service provides as specified in `AccessibilityServiceInfo`  
`(/reference/android/accessibilityservice/AccessibilityServiceInfo.html)`. This setting can be  
changed at runtime by calling  
`android.accessibilityservice.AccessibilityService.setServiceInfo(android.accessibil`  
`ityservice.AccessibleServiceInfo)`  
`(/reference/android/accessibilityservice/AccessibilityService.html#setServiceInfo(android.acces`  
`sibilityservice.AccessibleServiceInfo))`.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
feedbackSpoken	0x00000001	Provides <code>FEEDBACK_SPOKEN</code> feedback.
feedbackHaptic	0x00000002	Provides <code>FEEDBACK_HAPTIC</code> feedback.
feedbackAudible	0x00000004	Provides <code>FEEDBACK_AUDIBLE</code> feedback.
feedbackVisual	0x00000008	Provides <code>FEEDBACK_VISUAL</code> feedback.
feedbackGeneric	0x00000010	Provides <code>FEEDBACK_GENERIC</code> feedback.
feedbackAllMask	0xffffffff	Provides <code>FEEDBACK_ALL_MASK</code> feedback.

This corresponds to the global attribute resource symbol `accessibilityFeedbackType`  
`(/reference/android/R.attr.html#accessibilityFeedbackType)`.

Constant Value: 4 (0x00000004)

#### public static final int AccessibilityService\_accessibilityFlags

Additional flags as specified in `AccessibilityServiceInfo`  
`(/reference/android/accessibilityservice/AccessibilityServiceInfo.html)`. This setting can be  
changed at runtime by calling  
`android.accessibilityservice.AccessibilityService.setServiceInfo(android.accessibil`  
`ityservice.AccessibleServiceInfo)`  
`(/reference/android/accessibilityservice/AccessibilityService.html#setServiceInfo(android.acces`  
`sibilityservice.AccessibleServiceInfo))`.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
flagDefault	0x00000001	Has flag <code>DEFAULT</code>

flagIncludeNotImportantViews 0x00000002 Has flag  
[FLAG\\_INCLUDE\\_NOT\\_IMPORTANT\\_VIEWS](#)  
flagRequestTouchExplorationMode 0x00000004 Has flag  
[FLAG\\_REQUEST\\_TOUCH\\_EXPLORATION\\_MODE](#)

This corresponds to the global attribute resource symbol [accessibilityFlags](#)  
[\(/reference/android/R.attr.html#accessibilityFlags\)](#).

Constant Value: 6 (0x00000006)

#### **public static final int AccessibilityService\_canRetrieveWindowContent**

Flag whether the accessibility service wants to be able to retrieve the active window content. This setting cannot be changed at runtime.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [canRetrieveWindowContent](#)  
[\(/reference/android/R.attr.html#canRetrieveWindowContent\)](#).

Constant Value: 7 (0x00000007)

#### **public static final int AccessibilityService\_description**

Short description of the accessibility service purpose or behavior.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [description](#)  
[\(/reference/android/R.attr.html#description\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int AccessibilityService\_notificationTimeout**

The minimal period in milliseconds between two accessibility events of the same type are sent to this service. This setting can be changed at runtime by calling

[android.accessibilityservice.AccessibilityService.setServiceInfo\(android.accessibilityservice.AccessibilityServiceInfo\)](#)  
[\(/reference/android/accessibilityservice/AccessibilityService.html#setServiceInfo\(android.accessibilityservice.AccessibilityServiceInfo\)\)](#).

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [notificationTimeout](#)  
[\(/reference/android/R.attr.html#notificationTimeout\)](#).

Constant Value: 5 (0x00000005)

#### **public static final int AccessibilityService\_packageNames**

Comma separated package names from which this service would like to receive events (leave out for all packages).

[android.accessibilityservice.AccessibilityService.setServiceInfo\(android.accessibilityservice.AccessibilityServiceInfo\)](#)  
[\(/reference/android/accessibilityservice/AccessibilityService.html#setServiceInfo\(android.accessibilityservice.AccessibilityServiceInfo\)\)](#).

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [packageName](#)  
[\(/reference/android/R.attr.html#packageName\)](#).

Constant Value: 3 (0x00000003)

#### **public static final int AccessibilityService\_settingsActivity**

Component name of an activity that allows the user to modify the settings for this service. This setting

cannot be changed at runtime.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [settingsActivity](#)  
(/reference/android/R.attr.html#settingsActivity).

Constant Value: 1 (0x00000001)

#### public static final int AccountAuthenticator\_accountPreferences

A preferences.xml file for authenticator-specific settings.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [accountPreferences](#)  
(/reference/android/R.attr.html#accountPreferences).

Constant Value: 4 (0x00000004)

#### public static final int AccountAuthenticator\_accountType

The account type this authenticator handles.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [accountType](#)  
(/reference/android/R.attr.html#accountType).

Constant Value: 2 (0x00000002)

#### public static final int AccountAuthenticator\_customTokens

Account handles its own token storage and permissions. Default to false

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [customTokens](#)  
(/reference/android/R.attr.html#customTokens).

Constant Value: 5 (0x00000005)

#### public static final int AccountAuthenticator\_icon

The icon of the authenticator.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [icon](#) (/reference/android/R.attr.html#icon).

Constant Value: 1 (0x00000001)

#### public static final int AccountAuthenticator\_label

The user-visible name of the authenticator.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This corresponds to the global attribute resource symbol [label](#)  
(/reference/android/R.attr.html#label).

Constant Value: 0 (0x00000000)

#### public static final int AccountAuthenticator\_smallIcon

Smaller icon of the authenticator.

Must be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ] [ type: ] name`".

This corresponds to the global attribute resource symbol [smallIcon](#)  
[\(/reference/android/R.attr.html#smallIcon\)](#).

Constant Value: 3 (0x00000003)

#### **public static final int ActionBar.LayoutParams\_layout\_gravity**

Standard gravity constant that a child supplies to its parent. Defines how the child view should be positioned, on both the X and Y axes, within its enclosing layout.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
top	0x30	Push object to the top of its container, not changing its size.
bottom	0x50	Push object to the bottom of its container, not changing its size.
left	0x03	Push object to the left of its container, not changing its size.
right	0x05	Push object to the right of its container, not changing its size.
center_vertical	0x10	Place object in the vertical center of its container, not changing its size.
fill_vertical	0x70	Grow the vertical size of the object if needed so it completely fills its container.
center_horizontal	0x01	Place object in the horizontal center of its container, not changing its size.
fill_horizontal	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
center	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
fill	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
clip_vertical	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
clip_horizontal	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a right gravity will clip the left edge, and neither will clip both edges.
start	0x00800003	Push object to the beginning of its container, not changing its size.
end	0x00800005	Push object to the end of its container, not changing its size.

This corresponds to the global attribute resource symbol [layout\\_gravity](#)  
[\(/reference/android/R.attr.html#layout\\_gravity\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int ActionBar.background**

Specifies a background drawable for the action bar.

May be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ] [ type: ] name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [background](#)  
[\(/reference/android/R.attr.html#background\)](#).

Constant Value: 2 (0x00000002)

#### **public static final int ActionBar.backgroundSplit**

Specifies a background drawable for the bottom component of a split action bar.

May be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ] [ type: ] name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [backgroundSplit](#)

[\(/reference/android/R.attr.html#backgroundSplit\).](#)

Constant Value: 18 (0x00000012)

#### public static final int **ActionBar\_backgroundStacked**

Specifies a background drawable for a second stacked row of the action bar.

May be a reference to another resource, in the form "`@[+][ package: ]type: name`" or to a theme attribute in the form "`?[ package: ][ type: ]name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [backgroundStacked](#)  
[\(/reference/android/R.attr.html#backgroundStacked\).](#)

Constant Value: 17 (0x00000011)

#### public static final int **ActionBar\_customNavigationLayout**

Specifies a layout for custom navigation. Overrides navigationMode.

Must be a reference to another resource, in the form "`@[+][ package: ]type: name`" or to a theme attribute in the form "`?[ package: ][ type: ]name`".

This corresponds to the global attribute resource symbol [customNavigationLayout](#)  
[\(/reference/android/R.attr.html#customNavigationLayout\).](#)

Constant Value: 10 (0x0000000a)

#### public static final int **ActionBar\_displayOptions**

Options affecting how the action bar is displayed.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
none	0	
useLogo	0x1	
showHome	0x2	
homeAsUp	0x4	
showTitle	0x8	
showCustom	0x10	
disableHome	0x20	

This corresponds to the global attribute resource symbol [displayOptions](#)  
[\(/reference/android/R.attr.html#displayOptions\).](#)

Constant Value: 8 (0x00000008)

#### public static final int **ActionBar\_divider**

Specifies the drawable used for item dividers.

May be a reference to another resource, in the form "`@[+][ package: ]type: name`" or to a theme attribute in the form "`?[ package: ][ type: ]name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [divider](#)  
[\(/reference/android/R.attr.html#divider\).](#)

Constant Value: 3 (0x00000003)

#### public static final int **ActionBar\_height**

Specifies a fixed height.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "`@[ package: ]type: name`") or theme attribute (in the form "`?[ package: ][ type: ]name`") containing a value of this type.

This corresponds to the global attribute resource symbol [height](#)  
[\(/reference/android/R.attr.html#height\).](#)

Constant Value: 4 (0x00000004)

#### **public static final int ActionBar\_homeLayout**

Specifies a layout to use for the "home" section of the action bar.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [\\_homeLayout](#)  
(/reference/android/R.attr.html#homeLayout).

Constant Value: 15 (0x0000000f)

#### **public static final int ActionBar\_icon**

Specifies the drawable used for the application icon.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [\\_icon](#) (/reference/android/R.attr.html#icon).

Constant Value: 0 (0x00000000)

#### **public static final int ActionBar\_ineterminateProgressStyle**

Specifies a style resource to use for an indeterminate progress spinner.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [\\_ineterminateProgressStyle](#)  
(/reference/android/R.attr.html#ineterminateProgressStyle).

Constant Value: 13 (0x0000000d)

#### **public static final int ActionBar\_itemPadding**

Specifies padding that should be applied to the left and right sides of system-provided items in the bar.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*][*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [\\_itemPadding](#)  
(/reference/android/R.attr.html#itemPadding).

Constant Value: 16 (0x00000010)

#### **public static final int ActionBar\_logo**

Specifies the drawable used for the application logo.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [\\_logo](#) (/reference/android/R.attr.html#logo).

Constant Value: 6 (0x00000006)

#### **public static final int ActionBar\_navigationMode**

The type of navigation to use.

Must be one of the following constant values.

Constant Value	Description
normal 0	Normal static title text
listMode 1	The action bar will use a selection list for navigation.
tabMode 2	The action bar will use a series of horizontal tabs for navigation.

This corresponds to the global attribute resource symbol [\\_navigationMode](#)  
(/reference/android/R.attr.html#navigationMode).

Constant Value: 7 (0x00000007)

#### **public static final int ActionBar\_progressBarPadding**

Specifies the horizontal padding on either end for an embedded progress bar.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [progressBarPadding](#)  
[\(/reference/android/R.attr.html#progressBarPadding\)](#).

Constant Value: 14 (0x0000000e)

#### **public static final int ActionBar\_progressBarStyle**

Specifies a style resource to use for an embedded progress bar.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [progressBarStyle](#)  
[\(/reference/android/R.attr.html#progressBarStyle\)](#).

Constant Value: 1 (0x00000001)

#### **public static final int ActionBar\_subtitle**

Specifies subtitle text used for navigationMode="normal"

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [subtitle](#)  
[\(/reference/android/R.attr.html#subtitle\)](#).

Constant Value: 9 (0x00000009)

#### **public static final int ActionBar\_subtitleTextStyle**

Specifies a style to use for subtitle text.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [subtitleTextStyle](#)  
[\(/reference/android/R.attr.html#subtitleTextStyle\)](#).

Constant Value: 12 (0x0000000c)

#### **public static final int ActionBar\_title**

Specifies title text used for navigationMode="normal"

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [title](#)  
[\(/reference/android/R.attr.html#title\)](#).

Constant Value: 5 (0x00000005)

#### **public static final int ActionBar\_titleTextStyle**

Specifies a style to use for title text.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [titleTextStyle](#)

[\(/reference/android/R.attr.html#titleTextStyle\)](#).

Constant Value: 11 (0x0000000b)

#### public static final int **ActionMenuItemView\_minWidth**

This symbol is the offset where the [minWidth](#) ([\(/reference/android/R.attr.html#minWidth\)](#)) attribute's value can be found in the [ActionMenuItemView](#) ([\(/reference/android/R.styleable.html#ActionMenuItemView\)](#) array.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[*package*:] type:*name*") or theme attribute (in the form "?[*package*:] [type:*name*]" containing a value of this type.

Constant Value: 0 (0x00000000)

#### public static final int **ActionMode\_background**

Specifies a background for the action mode bar.

May be a reference to another resource, in the form "@[+] [*package*:] type:*name*" or to a theme attribute in the form "?[*package*:] [type:*name*]".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [background](#) ([\(/reference/android/R.attr.html#background\)](#)).

Constant Value: 0 (0x00000000)

#### public static final int **ActionMode\_backgroundSplit**

Specifies a background for the split action mode bar.

May be a reference to another resource, in the form "@[+] [*package*:] type:*name*" or to a theme attribute in the form "?[*package*:] [type:*name*]".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [backgroundSplit](#) ([\(/reference/android/R.attr.html#backgroundSplit\)](#)).

Constant Value: 4 (0x00000004)

#### public static final int **ActionMode\_height**

Specifies a fixed height for the action mode bar.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[*package*:] type:*name*") or theme attribute (in the form "?[*package*:] [type:*name*]" containing a value of this type.

This corresponds to the global attribute resource symbol [height](#) ([\(/reference/android/R.attr.html#height\)](#)).

Constant Value: 1 (0x00000001)

#### public static final int **ActionMode\_subtitleTextStyle**

Specifies a style to use for subtitle text.

May be a reference to another resource, in the form "@[+] [*package*:] type:*name*" or to a theme attribute in the form "?[*package*:] [type:*name*]".

This corresponds to the global attribute resource symbol [subtitleTextStyle](#) ([\(/reference/android/R.attr.html#subtitleTextStyle\)](#)).

Constant Value: 3 (0x00000003)

#### public static final int **ActionMode\_titleTextStyle**

Specifies a style to use for title text.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [titleTextStyle](#)  
(/reference/android/R.attr.html#titleTextStyle).

Constant Value: 2 (0x00000002)

#### public static final int AdapterViewAnimator\_animateFirstView

Defines whether to animate the current View when the ViewAnimation is first displayed.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[+][*package:*]*type:name*") or theme attribute (in the form "?[*package:*][*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [animateFirstView](#)  
(/reference/android/R.attr.html#animateFirstView).

Constant Value: 2 (0x00000002)

#### public static final int AdapterViewAnimator\_inAnimation

Identifier for the animation to use when a view is shown.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [inAnimation](#)  
(/reference/android/R.attr.html#inAnimation).

Constant Value: 0 (0x00000000)

#### public static final int AdapterViewAnimator\_loopViews

Defines whether the animator loops to the first view once it has reached the end of the list.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[+][*package:*]*type:name*") or theme attribute (in the form "?[*package:*][*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [loopViews](#)  
(/reference/android/R.attr.html#loopViews).

Constant Value: 3 (0x00000003)

#### public static final int AdapterViewAnimator\_outAnimation

Identifier for the animation to use when a view is hidden.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [outAnimation](#)  
(/reference/android/R.attr.html#outAnimation).

Constant Value: 1 (0x00000001)

#### public static final int AdapterViewFlipper\_autoStart

When true, automatically start animating

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[+][*package:*]*type:name*") or theme attribute (in the form "?[*package:*][*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [autoStart](#)  
(/reference/android/R.attr.html#autoStart).

Constant Value: 1 (0x00000001)

#### public static final int AdapterViewFlipper\_flipInterval

This symbol is the offset where the [flipInterval](#) (/reference/android/R.attr.html#flipInterval) attribute's value can be found in the [AdapterViewFlipper](#)

[\(/reference/android/R.styleable.html#AdapterViewFlipper\)](#) array.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 0 (0x00000000)

#### **public static final int AlertDialog\_bottomBright**

This symbol is the offset where the [bottomBright](#) ([\(/reference/android/R.styleable.html#bottomBright\)](#)) attribute's value can be found in the [AlertDialog](#) ([\(/reference/android/R.styleable.html#AlertDialog\)](#)) array.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

Constant Value: 7 (0x00000007)

#### **public static final int AlertDialog\_bottomDark**

This symbol is the offset where the [bottomDark](#) ([\(/reference/android/R.styleable.html#bottomDark\)](#)) attribute's value can be found in the [AlertDialog](#) ([\(/reference/android/R.styleable.html#AlertDialog\)](#)) array.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

Constant Value: 3 (0x00000003)

#### **public static final int AlertDialog\_bottomMedium**

This symbol is the offset where the [bottomMedium](#) ([\(/reference/android/R.styleable.html#bottomMedium\)](#)) attribute's value can be found in the [AlertDialog](#) ([\(/reference/android/R.styleable.html#AlertDialog\)](#)) array.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

Constant Value: 8 (0x00000008)

#### **public static final int AlertDialog\_centerBright**

This symbol is the offset where the [centerBright](#) ([\(/reference/android/R.styleable.html#centerBright\)](#)) attribute's value can be found in the [AlertDialog](#) ([\(/reference/android/R.styleable.html#AlertDialog\)](#)) array.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

Constant Value: 6 (0x00000006)

#### **public static final int AlertDialog\_centerDark**

This symbol is the offset where the [centerDark](#) ([\(/reference/android/R.styleable.html#centerDark\)](#)) attribute's value can be found in the [AlertDialog](#) ([\(/reference/android/R.styleable.html#AlertDialog\)](#)) array.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

Constant Value: 2 (0x00000002)

#### **public static final int AlertDialog\_centerMedium**

This symbol is the offset where the [centerMedium](#) ([\(/reference/android/R.styleable.html#centerMedium\)](#))

attribute's value can be found in the [AlertDialog](#) ([/reference/android/R.styleable.html#AlertDialog](#)) array.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

Constant Value: 9 (0x00000009)

#### public static final int **AlertDialog\_fullBright**

This symbol is the offset where the [fullBright](#) ([/reference/android/R.attr.html#fullBright](#)) attribute's value can be found in the [AlertDialog](#) ([/reference/android/R.styleable.html#AlertDialog](#)) array.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

Constant Value: 4 (0x00000004)

#### public static final int **AlertDialog\_fullDark**

This symbol is the offset where the [fullDark](#) ([/reference/android/R.attr.html#fullDark](#)) attribute's value can be found in the [AlertDialog](#) ([/reference/android/R.styleable.html#AlertDialog](#)) array.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

Constant Value: 0 (0x00000000)

#### public static final int **AlertDialog\_layout**

Supply an identifier for the layout resource to inflate when the ViewStub becomes visible or when forced to do so. The layout resource must be a valid reference to a layout.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [layout](#) ([/reference/android/R.attr.html#layout](#)).

Constant Value: 10 (0x0000000a)

#### public static final int **AlertDialog\_topBright**

This symbol is the offset where the [topBright](#) ([/reference/android/R.attr.html#topBright](#)) attribute's value can be found in the [AlertDialog](#) ([/reference/android/R.styleable.html#AlertDialog](#)) array.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

Constant Value: 5 (0x00000005)

#### public static final int **AlertDialog\_topDark**

This symbol is the offset where the [topDark](#) ([/reference/android/R.attr.html#topDark](#)) attribute's value can be found in the [AlertDialog](#) ([/reference/android/R.styleable.html#AlertDialog](#)) array.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

Constant Value: 1 (0x00000001)

#### public static final int **AlphaAnimation\_fromAlpha**

This symbol is the offset where the [fromAlpha](#) ([/reference/android/R.attr.html#fromAlpha](#)) attribute's value can be found in the [AlphaAnimation](#) ([/reference/android/R.styleable.html#AlphaAnimation](#)) array.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 0 (0x00000000)

#### public static final int **AlphaAnimation\_toAlpha**

This symbol is the offset where the [toAlpha](#) (/reference/android/R.attr.html#toAlpha) attribute's value can be found in the [AlphaAnimation](#) (/reference/android/R.styleable.html#AlphaAnimation) array.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 1 (0x00000001)

#### public static final int **AnalogClock\_dial**

This symbol is the offset where the [dial](#) (/reference/android/R.attr.html#dial) attribute's value can be found in the [AnalogClock](#) (/reference/android/R.styleable.html#AnalogClock) array.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

Constant Value: 0 (0x00000000)

#### public static final int **AnalogClock\_hand\_hour**

This symbol is the offset where the [hand\\_hour](#) (/reference/android/R.attr.html#hand\_hour) attribute's value can be found in the [AnalogClock](#) (/reference/android/R.styleable.html#AnalogClock) array.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

Constant Value: 1 (0x00000001)

#### public static final int **AnalogClock\_hand\_minute**

This symbol is the offset where the [hand\\_minute](#) (/reference/android/R.attr.html#hand\_minute) attribute's value can be found in the [AnalogClock](#) (/reference/android/R.styleable.html#AnalogClock) array.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

Constant Value: 2 (0x00000002)

#### public static final int **AndroidManifestAction\_name**

The name of an action that is handled, using the Java-style naming convention. For example, to support [Intent.ACTION\\_VIEW](#) (/reference/android/content/Intent.html#ACTION\_VIEW) you would put android.intent.action.VIEW here. Custom actions should generally use a prefix matching the package name.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) (/reference/android/R.attr.html#name).

Constant Value: 0 (0x00000000)

#### public static final int **AndroidManifestActivityAlias\_description**

Descriptive text for the associated data.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [description](#) (/reference/android/R.attr.html#description).

Constant Value: 6 (0x00000006)

#### public static final int **AndroidManifestActivityAlias\_enabled**

Specify whether the activity-alias is enabled or not (that is, can be instantiated by the system). It can also be specified for an application as a whole, in which case a value of "false" will override any component specific values (a value of "true" will not override the component specific values).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [enabled](#)  
[\(/reference/android/R.attr.html#enabled\)](#).

Constant Value: 4 (0x00000004)

#### public static final int **AndroidManifestActivityAlias\_exported**

Flag indicating whether the given application component is available to other applications. If false, it can only be accessed by applications with its same user id (which usually means only by code in its own package). If true, it can be invoked by external entities, though which ones can do so may be controlled through permissions. The default value is false for activity, receiver, and service components that do not specify any intent filters; it is true for activity, receiver, and service components that do have intent filters (implying they expect to be invoked by others who do not know their particular component name) and for all content providers.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [exported](#)  
[\(/reference/android/R.attr.html#exported\)](#).

Constant Value: 5 (0x00000005)

#### public static final int **AndroidManifestActivityAlias\_icon**

A Drawable resource providing a graphical representation of its associated item. Use with the application tag (to supply a default icon for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific icon for that component). It may also be used with the intent-filter tag to supply an icon to show to the user when an activity is being selected based on a particular Intent.

The given icon will be used to display to the user a graphical representation of its associated component; for example, as the icon for main activity that is displayed in the launcher. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [icon](#) [\(/reference/android/R.attr.html#icon\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **AndroidManifestActivityAlias\_label**

A user-legible name for the given item. Use with the application tag (to supply a default label for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific label for that component). It may also be used with the intent-filter tag to supply a label to show to the user when an activity is being selected based on a particular Intent.

The given label will be used wherever the user sees information about its associated component; for example, as the name of a main activity that is displayed in the launcher. You should generally set this to a reference to a string resource, so that it can be localized, however it is also allowed to supply a plain string for quick and dirty programming.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This corresponds to the global attribute resource symbol [label](#)  
[\(/reference/android/R.attr.html#label\)](#).

Constant Value: 0 (0x00000000)

## **public static final int AndroidManifestActivityAlias\_logo**

A Drawable resource providing an extended graphical logo for its associated item. Use with the application tag (to supply a default logo for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific logo for that component). It may also be used with the intent-filter tag to supply a logo to show to the user when an activity is being selected based on a particular Intent.

The given logo will be used to display to the user a graphical representation of its associated component; for example as the header in the Action Bar. The primary differences between an icon and a logo are that logos are often wider and more detailed, and are used without an accompanying text caption. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [logo](#) ([/reference/android/R.attr.html#logo](#)).

Constant Value: 8 (0x00000008)

## **public static final int AndroidManifestActivityAlias\_name**

Required name of the class implementing the activity, deriving from [Activity](#) ([/reference/android/app/Activity.html](#)). This is a fully qualified class name (for example, com.mycompany.myapp.MyActivity); as a short-hand if the first character of the class is a period then it is appended to your package name.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) ([/reference/android/R.attr.html#name](#)).

Constant Value: 2 (0x00000002)

## **public static final int AndroidManifestActivityAlias\_parentActivityName**

The name of the logical parent of the activity as it appears in the manifest.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [parentActivityName](#) ([/reference/android/R.attr.html#parentActivityName](#)).

Constant Value: 9 (0x00000009)

## **public static final int AndroidManifestActivityAlias\_permission**

Specify a permission that a client is required to have in order to use the associated object. If the client does not hold the named permission, its request will fail. See the [Security and Permissions](#) ([/guide/topics/security/security.html](#)) document for more information on permissions.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [permission](#) ([/reference/android/R.attr.html#permission](#)).

Constant Value: 3 (0x00000003)

## **public static final int AndroidManifestActivityAlias\_targetActivity**

The name of the activity this alias should launch. The activity must be in the same manifest as the alias, and have been defined in that manifest before the alias here. This must use a Java-style naming convention to ensure the name is unique, for example "com.mycompany.MyName".

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [targetActivity](#) ([/reference/android/R.attr.html#targetActivity](#)).

Constant Value: 7 (0x00000007)

#### public static final int **AndroidManifestActivity\_allowTaskReparenting**

Specify that an activity can be moved out of a task it is in to the task it has an affinity for when appropriate. Use with the application tag (to supply a default for all activities in the application), or with an activity tag (to supply a specific setting for that component).

Normally when an application is started, it is associated with the task of the activity that started it and stays there for its entire lifetime. You can use the allowTaskReparenting feature to force an activity to be re-parented to a different task when the task it is in goes to the background. Typically this is used to cause the activities of an application to move back to the main task associated with that application. The activity is re-parented to the task with the same [taskAffinity](#)  
[\(/reference/android/R.attr.html#taskAffinity\)](#) as it has.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [allowTaskReparenting](#)  
[\(/reference/android/R.attr.html#allowTaskReparenting\)](#).

Constant Value: 19 (0x00000013)

#### public static final int **AndroidManifestActivity\_alwaysRetainTaskState**

Specify whether an activity's task state should always be maintained by the system, or if it is allowed to reset the task to its initial state in certain situations.

Normally the system will reset a task (remove all activities from the stack and reset the root activity) in certain situations when the user re-selects that task from the home screen. Typically this will be done if the user hasn't visited that task for a certain amount of time, such as 30 minutes.

By setting this attribute, the user will always return to your task in its last state, regardless of how they get there. This is useful, for example, in an application like the web browser where there is a lot of state (such as multiple open tabs) that the application would not like to lose.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [alwaysRetainTaskState](#)  
[\(/reference/android/R.attr.html#alwaysRetainTaskState\)](#).

Constant Value: 18 (0x00000012)

#### public static final int **AndroidManifestActivity\_clearTaskOnLaunch**

Specify whether an activity's task should be cleared when it is re-launched from the home screen. As a result, every time the user starts the task, they will be brought to its root activity, regardless of whether they used BACK or HOME to last leave it. This flag only applies to activities that are used to start the root of a new task.

An example of the use of this flag would be for the case where a user launches activity A from home, and from there goes to activity B. They now press home, and then return to activity A. Normally they would see activity B, since that is what they were last doing in A's task. However, if A has set this flag to true, then upon going to the background all of the tasks on top of it (B in this case) are removed, so when the user next returns to A they will restart at its original activity.

When this option is used in conjunction with [allowTaskReparenting](#)  
[\(/reference/android/R.attr.html#allowTaskReparenting\)](#), the allowTaskReparenting trumps the clear. That is, all activities above the root activity of the task will be removed: those that have an affinity will be moved to the task they are associated with, otherwise they will simply be dropped as described here.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [clearTaskOnLaunch](#)  
[\(/reference/android/R.attr.html#clearTaskOnLaunch\)](#).

Constant Value: 11 (0x0000000b)

#### public static final int **AndroidManifestActivity\_configChanges**

Specify one or more configuration changes that the activity will handle itself. If not specified, the activity will be restarted if any of these configuration changes happen in the system. Otherwise, the activity will remain running and its [Activity.onConfigurationChanged](#)  
[\(/reference/android/app/Activity.html#onConfigurationChanged\(android.content.res.Configuration\)}](#)  
\_ method called with the new configuration.

Note that all of these configuration changes can impact the resource values seen by the application, so you will generally need to re-retrieve all resources (including view layouts, drawables, etc) to correctly handle any configuration change.

These values must be kept in sync with those in [ActivityInfo](#)  
[\(/reference/android/content/pm/ActivityInfo.html\)](#) and include/utils/ResourceTypes.h.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
mcc	0x0001	The IMSI MCC has changed, that is a SIM has been detected and updated the Mobile Country Code.
mnc	0x0002	The IMSI MNC has changed, that is a SIM has been detected and updated the Mobile Network Code.
locale	0x0004	The locale has changed, that is the user has selected a new language that text should be displayed in.
touchscreen	0x0008	The touchscreen has changed. Should never normally happen.
keyboard	0x0010	The keyboard type has changed, for example the user has plugged in an external keyboard.
keyboardHidden	0x0020	The keyboard or navigation accessibility has changed, for example the user has slid the keyboard out to expose it. Note that despite its name, this applied to any accessibility: keyboard or navigation.
navigation	0x0040	The navigation type has changed. Should never normally happen.
orientation	0x0080	The screen orientation has changed, that is the user has rotated the device.
screenLayout	0x0100	The screen layout has changed. This might be caused by a different display being activated.
uiMode	0x0200	The global user interface mode has changed. For example, going in or out of car mode, night mode changing, etc.
screenSize	0x0400	The current available screen size has changed. If applications don't target at least <a href="#">HONEYCOMB_MR2</a> then the activity will always handle this itself (the change will not result in a restart). This represents a change in the currently available size, so will change when the user switches between landscape and portrait.
smallestScreenSize	0x0800	The physical screen size has changed. If applications don't target at least <a href="#">HONEYCOMB_MR2</a> then the activity will always handle this itself (the change will not result in a restart). This represents a change in size regardless of orientation, so will only change when the actual physical screen size has changed such as switching to an external display.
layoutDirection	0x2000	The layout direction has changed. For example going from LTR to RTL.
fontScale	0x40000000	The font scaling factor has changed, that is the user has selected a new global font size.

This corresponds to the global attribute resource symbol [configChanges](#)

[\(/reference/android/R.attr.html#configChanges\).](#)

Constant Value: 16 (0x00000010)

#### public static final int **AndroidManifestActivity\_description**

Descriptive text for the associated data.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [description](#)  
[\(/reference/android/R.attr.html#description\).](#)

Constant Value: 17 (0x00000011)

#### public static final int **AndroidManifestActivity\_enabled**

Specify whether the activity is enabled or not (that is, can be instantiated by the system). It can also be specified for an application as a whole, in which case a value of "false" will override any component specific values (a value of "true" will not override the component specific values).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [enabled](#)  
[\(/reference/android/R.attr.html#enabled\)](#).

Constant Value: 5 (0x00000005)

#### **public static final int `AndroidManifestActivity_excludeFromRecents`**

Indicates that an Activity should be excluded from the list of recently launched activities.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [excludeFromRecents](#)  
[\(/reference/android/R.attr.html#excludeFromRecents\)](#).

Constant Value: 13 (0x0000000d)

#### **public static final int `AndroidManifestActivity_exported`**

Flag indicating whether the given application component is available to other applications. If false, it can only be accessed by applications with its same user id (which usually means only by code in its own package). If true, it can be invoked by external entities, though which ones can do so may be controlled through permissions. The default value is false for activity, receiver, and service components that do not specify any intent filters; it is true for activity, receiver, and service components that do have intent filters (implying they expect to be invoked by others who do not know their particular component name) and for all content providers.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [exported](#)  
[\(/reference/android/R.attr.html#exported\)](#).

Constant Value: 6 (0x00000006)

#### **public static final int `AndroidManifestActivity_finishOnCloseSystemDialogs`**

Specify whether an activity should be finished when a "close system windows" request has been made. This happens, for example, when the home key is pressed, when the device is locked, when a system dialog showing recent applications is displayed, etc.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [finishOnCloseSystemDialogs](#)  
[\(/reference/android/R.attr.html#finishOnCloseSystemDialogs\)](#).

Constant Value: 22 (0x00000016)

#### **public static final int `AndroidManifestActivity_finishOnTaskLaunch`**

Specify whether an activity should be finished when its task is brought to the foreground by relaunching from the home screen.

If both this option and [allowTaskReparenting](#)  
[\(/reference/android/R.attr.html#allowTaskReparenting\)](#) are specified, the finish trumps the affinity: the affinity will be ignored and the activity simply finished.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [finishOnTaskLaunch](#)  
[\(/reference/android/R.attr.html#finishOnTaskLaunch\)](#).

Constant Value: 10 (0x0000000a)

## **public static final int AndroidManifestActivity\_hardwareAccelerated**

Flag indicating whether the application's rendering should be hardware accelerated if possible. This flag is turned on by default for applications that are targeting [ICE\\_CREAM\\_SANDWICH](#) ([/reference/android/os/Build.VERSION\\_CODES.html#ICE\\_CREAM\\_SANDWICH](#)) or later.

This flag can be set on the application and any activity declared in the manifest. When enabled for the application, each activity is automatically assumed to be hardware accelerated. This flag can be overridden in the activity tags, either turning it off (if on for the application) or on (if off for the application.)

When this flag is turned on for an activity (either directly or via the application tag), every window created from the activity, including the activity's own window, will be hardware accelerated, if possible.

Please refer to the documentation of [FLAG\\_HARDWARE\\_ACCELERATED](#) ([/reference/android/view/WindowManager.LayoutParams.html#FLAG\\_HARDWARE\\_ACCELERATED](#)) for more information on how to control this flag programmatically.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [hardwareAccelerated](#) ([/reference/android/R.attr.html#hardwareAccelerated](#)).

Constant Value: 25 (0x00000019)

## **public static final int AndroidManifestActivity\_icon**

A Drawable resource providing a graphical representation of its associated item. Use with the application tag (to supply a default icon for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific icon for that component). It may also be used with the intent-filter tag to supply an icon to show to the user when an activity is being selected based on a particular Intent.

The given icon will be used to display to the user a graphical representation of its associated component; for example, as the icon for main activity that is displayed in the launcher. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [icon](#) ([/reference/android/R.attr.html#icon](#)).

Constant Value: 2 (0x00000002)

## **public static final int AndroidManifestActivity\_immersive**

Flag declaring this activity to be 'immersive'; immersive activities should not be interrupted with other activities or notifications.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [immersive](#) ([/reference/android/R.attr.html#immersive](#)).

Constant Value: 24 (0x00000018)

## **public static final int AndroidManifestActivity\_label**

A user-legible name for the given item. Use with the application tag (to supply a default label for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific label for that component). It may also be used with the intent-filter tag to supply a label to show to the user when an activity is being selected based on a particular Intent.

The given label will be used wherever the user sees information about its associated component; for example, as the name of a main activity that is displayed in the launcher. You should generally set this to a reference to a string resource, so that it can be localized, however it is also allowed to supply a plain string for quick and dirty programming.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This corresponds to the global attribute resource symbol [label](#)  
[\(/reference/android/R.attr.html#label\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **AndroidManifestActivity\_launchMode**

Specify how an activity should be launched. See the [Tasks and Back Stack](#) ([/guide/topics/fundamentals/tasks-and-back-stack.html](#)) document for important information on how these options impact the behavior of your application.

If this attribute is not specified, standard launch mode will be used. Note that the particular launch behavior can be changed in some ways at runtime through the [Intent](#)  
[\(/reference/android/content/Intent.html\)](#) flags [FLAG\\_ACTIVITY\\_SINGLE\\_TOP](#)  
[\(/reference/android/content/Intent.html#FLAG\\_ACTIVITY\\_SINGLE\\_TOP\)](#), [FLAG\\_ACTIVITY\\_NEW\\_TASK](#)  
[\(/reference/android/content/Intent.html#FLAG\\_ACTIVITY\\_NEW\\_TASK\)](#), and  
[FLAG\\_ACTIVITY\\_MULTIPLE\\_TASK](#)  
[\(/reference/android/content/Intent.html#FLAG\\_ACTIVITY\\_MULTIPLE\\_TASK\)](#).

Must be one of the following constant values.

Constant	Value	Description
standard	0	The default mode, which will usually create a new instance of the activity when it is started, though this behavior may change with the introduction of other options such as <a href="#">Intent.FLAG_ACTIVITY_NEW_TASK</a> .
singleTop	1	If, when starting the activity, there is already an instance of the same activity class in the foreground that is interacting with the user, then re-use that instance. This existing instance will receive a call to <a href="#">Activity.onNewIntent()</a> with the new Intent that is being started.
singleTask	2	If, when starting the activity, there is already a task running that starts with this activity, then instead of starting a new instance the current task is brought to the front. The existing instance will receive a call to <a href="#">Activity.onNewIntent()</a> with the new Intent that is being started, and with the <a href="#">Intent.FLAG_ACTIVITY_BROUGHT_TO_FRONT</a> flag set. This is a superset of the singleTop mode, where if there is already an instance of the activity being started at the top of the stack, it will receive the Intent as described there (without the FLAG_ACTIVITY_BROUGHT_TO_FRONT flag set). See the <a href="#">Tasks and Back Stack</a> document for more details about tasks.
singleInstance	3	Only allow one instance of this activity to ever be running. This activity gets a unique task with only itself running in it; if it is ever launched again with the same Intent, then that task will be brought forward and its <a href="#">Activity.onNewIntent()</a> method called. If this activity tries to start a new activity, that new activity will be launched in a separate task. See the <a href="#">Tasks and Back Stack</a> document for more details about tasks.

This corresponds to the global attribute resource symbol [launchMode](#)  
[\(/reference/android/R.attr.html#launchMode\)](#).

Constant Value: 14 (0x0000000e)

#### public static final int **AndroidManifestActivity\_logo**

A Drawable resource providing an extended graphical logo for its associated item. Use with the application tag (to supply a default logo for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific logo for that component). It may also be used with the intent-filter tag to supply a logo to show to the user when an activity is being selected based on a particular Intent.

The given logo will be used to display to the user a graphical representation of its associated component; for example as the header in the Action Bar. The primary differences between an icon and a logo are that logos are often wider and more detailed, and are used without an accompanying text caption. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[+][ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [logo](#) [\(/reference/android/R.attr.html#logo\)](#).

Constant Value: 23 (0x00000017)

#### public static final int **AndroidManifestActivity\_multiprocess**

Specify whether a component is allowed to have multiple instances of itself running in different processes. Use with the activity and provider tags.

Normally the system will ensure that all instances of a particular component are only running in a single

process. You can use this attribute to disable that behavior, allowing the system to create instances wherever they are used (provided permissions allow it). This is most often used with content providers, so that instances of a provider can be created in each client process, allowing them to be used without performing IPC.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [multiprocess](#)  
(/reference/android/R.attr.html#multiprocess).

Constant Value: 9 (0x00000009)

#### public static final int **AndroidManifestActivity\_name**

Required name of the class implementing the activity, deriving from [Activity](#)  
(/reference/android/app/Activity.html). This is a fully qualified class name (for example, com.mycompany.myapp.MyActivity); as a short-hand if the first character of the class is a period then it is appended to your package name.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) (/reference/android/R.attr.html#name).

Constant Value: 3 (0x00000003)

#### public static final int **AndroidManifestActivity\_noHistory**

Specify whether an activity should be kept in its history stack. If this attribute is set, then as soon as the user navigates away from the activity it will be finished and they will no longer be able to return to it.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [noHistory](#)  
(/reference/android/R.attr.html#noHistory).

Constant Value: 21 (0x00000015)

#### public static final int **AndroidManifestActivity\_parentActivityName**

The name of the logical parent of the activity as it appears in the manifest.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [parentActivityName](#)  
(/reference/android/R.attr.html#parentActivityName).

Constant Value: 27 (0x0000001b)

#### public static final int **AndroidManifestActivity\_permission**

Specify a permission that a client is required to have in order to use the associated object. If the client does not hold the named permission, its request will fail. See the [Security and Permissions](#)  
(/guide/topics/security/security.html) document for more information on permissions.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [permission](#)  
(/reference/android/R.attr.html#permission).

Constant Value: 4 (0x00000004)

#### public static final int **AndroidManifestActivity\_process**

Specify a specific process that the associated code is to run in. Use with the application tag (to supply a

default process for all application components), or with the activity, receiver, service, or provider tag (to supply a specific icon for that component).

Application components are normally run in a single process that is created for the entire application. You can use this tag to modify where they run. If the process name begins with a ':' character, a new process private to that application will be created when needed to run that component (allowing you to spread your application across multiple processes). If the process name begins with a lower-case character, the component will be run in a global process of that name, provided that you have permission to do so, allowing multiple applications to share one process to reduce resource usage.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [process](#)  
[\(/reference/android/R.attr.html#process\)](#).

Constant Value: 7 (0x00000007)

#### public static final int **AndroidManifestActivity\_screenOrientation**

Specify the orientation an activity should be run in. If not specified, it will run in the current preferred orientation of the screen.

Must be one of the following constant values.

Constant	Value	Description
unspecified	-1	No preference specified: let the system decide the best orientation. This will either be the orientation selected by the activity below, or the user's preferred orientation if this activity is the bottom of a task. If the user explicitly turned off sensor based orientation through settings sensor based device rotation will be ignored. If not by default sensor based orientation will be taken into account and the orientation will change based on how the user rotates the device
landscape	0	Would like to have the screen in a landscape orientation: that is, with the display wider than it is tall, ignoring sensor data.
portrait	1	Would like to have the screen in a portrait orientation: that is, with the display taller than it is wide, ignoring sensor data.
user	2	Use the user's current preferred orientation of the handset.
behind	3	Keep the screen in the same orientation as whatever is behind this activity.
sensor	4	Orientation is determined by a physical orientation sensor: the display will rotate based on how the user moves the device.
nosensor	5	Always ignore orientation determined by orientation sensor: the display will not rotate when the user moves the device.
sensorLandscape	6	Would like to have the screen in landscape orientation, but can use the sensor to change which direction the screen is facing.
sensorPortrait	7	Would like to have the screen in portrait orientation, but can use the sensor to change which direction the screen is facing.
reverseLandscape	8	Would like to have the screen in landscape orientation, turned in the opposite direction from normal landscape.
reversePortrait	9	Would like to have the screen in portrait orientation, turned in the opposite direction from normal portrait.
fullSensor	10	Orientation is determined by a physical orientation sensor: the display will rotate based on how the user moves the device. This allows any of the 4 possible rotations, regardless of what the device will normally do (for example some devices won't normally use 180 degree rotation).

This corresponds to the global attribute resource symbol [screenOrientation](#)  
[\(/reference/android/R.attr.html#screenOrientation\)](#).

Constant Value: 15 (0x0000000f)

#### public static final int **AndroidManifestActivity\_showOnLockScreen**

Specify that an Activity should be shown over the lock screen and, in a multiuser environment, across all users' windows

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [showOnLockScreen](#)  
[\(/reference/android/R.attr.html#showOnLockScreen\)](#).

Constant Value: 29 (0x0000001d)

#### **public static final int [AndroidManifestActivity\\_singleUser](#)**

If set to true, a single instance of this component will run for all users. That instance will run as user 0, the default/primary user. When the app running is in processes for other users and interacts with this component (by binding to a service for example) those processes will always interact with the instance running for user 0. Enabling single user mode forces "exported" of the component to be false, to help avoid introducing multi-user security bugs. This feature is only available to applications built in to the system image; you must hold the permission INTERACT\_ACROSS\_USERS in order to use this feature. This flag can only be used with services, receivers, and providers; it can not be used with activities.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*] type:*name*") or theme attribute (in the form "?[*package:*] [ type:*name*]" containing a value of this type.

This corresponds to the global attribute resource symbol [singleUser](#)  
(/reference/android/R.attr.html#singleUser).

Constant Value: 28 (0x0000001c)

#### **public static final int [AndroidManifestActivity\\_stateNotNeeded](#)**

Indicates that an Activity does not need to have its freeze state (as returned by [onSaveInstanceState\(Bundle\)](#) (/reference/android/app/Activity.html#onSaveInstanceState(android.os.Bundle)) retained in order to be restarted. Generally you use this for activities that do not store any state. When this flag is set, if for some reason the activity is killed before it has a chance to save its state, then the system will not remove it from the activity stack like it normally would. Instead, the next time the user navigates to it its [onCreate\(Bundle\)](#) (/reference/android/app/Activity.html#onCreate(android.os.Bundle)) method will be called with a null icicle, just like it was starting for the first time.

This is used by the Home activity to make sure it does not get removed if it crashes for some reason.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*] type:*name*") or theme attribute (in the form "?[*package:*] [ type:*name*]" containing a value of this type.

This corresponds to the global attribute resource symbol [stateNotNeeded](#)  
(/reference/android/R.attr.html#stateNotNeeded).

Constant Value: 12 (0x0000000c)

#### **public static final int [AndroidManifestActivity\\_taskAffinity](#)**

Specify a task name that activities have an "affinity" to. Use with the application tag (to supply a default affinity for all activities in the application), or with the activity tag (to supply a specific affinity for that component).

The default value for this attribute is the same as the package name, indicating that all activities in the manifest should generally be considered a single "application" to the user. You can use this attribute to modify that behavior: either giving them an affinity for another task, if the activities are intended to be part of that task from the user's perspective, or using an empty string for activities that have no affinity to a task.

Must be a string value, using '\\.' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[*package:*] type:*name*") or theme attribute (in the form "?[*package:*] [ type:*name*]" containing a value of this type.

This corresponds to the global attribute resource symbol [taskAffinity](#)  
(/reference/android/R.attr.html#taskAffinity).

Constant Value: 8 (0x00000008)

#### **public static final int [AndroidManifestActivity\\_theme](#)**

The overall theme to use for an activity. Use with either the application tag (to supply a default theme for all activities) or the activity tag (to supply a specific theme for that activity).

This automatically sets your activity's Context to use this theme, and may also be used for "starting" animations prior to the activity being launched (to better match what the activity actually looks like). It is a reference to a style resource defining the theme. If not set, the default system theme will be used.

Must be a reference to another resource, in the form "@[+] [*package:*] type:*name*" or to a theme attribute in the form "?[*package:*] [ type:*name*]".

This corresponds to the global attribute resource symbol [theme](#)  
[\(/reference/android/R.attr.html#theme\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int `AndroidManifestActivity_uiOptions`**

Extra options for an activity's UI. Applies to either the <activity> or <application> tag. If specified on the <application> tag these will be considered defaults for all activities in the application.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
none	0	No extra UI options. This is the default.
splitActionBarWhenNarrow	1	Split the options menu into a separate bar at the bottom of the screen when severely constrained for horizontal space. (e.g. portrait mode on a phone.) Instead of a small number of action buttons appearing in the action bar at the top of the screen, the action bar will split into the top navigation section and the bottom menu section. Menu items will not be split across the two bars; they will always appear together.

This corresponds to the global attribute resource symbol [uiOptions](#)  
[\(/reference/android/R.attr.html#uiOptions\)](#).

Constant Value: 26 (0x0000001a)

#### **public static final int `AndroidManifestActivity_windowSoftInputMode`**

Specify the default soft-input mode for the main window of this activity. A value besides "unspecified" here overrides any value in the theme.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
stateUnspecified	0	Not specified, use what the system thinks is best. This is the default.
stateUnchanged	1	Leave the soft input window as-is, in whatever state it last was.
stateHidden	2	Make the soft input area hidden when normally appropriate (when the user is navigating forward to your window).
stateAlwaysHidden	3	Always make the soft input area hidden when this window has input focus.
stateVisible	4	Make the soft input area visible when normally appropriate (when the user is navigating forward to your window).
stateAlwaysVisible	5	Always make the soft input area visible when this window has input focus.
adjustUnspecified	0x00	The window resize/pan adjustment has not been specified, the system will automatically select between resize and pan modes, depending on whether the content of the window has any layout views that can scroll their contents. If there is such a view, then the window will be resized, with the assumption being that the resizable area can be reduced to make room for the input UI.
adjustResize	0x10	Always resize the window: the content area of the window is reduced to make room for the soft input area.
adjustPan	0x20	Don't resize the window to make room for the soft input area; instead pan the contents of the window as focus moves inside of it so that the user can see what they are typing. This is generally less desireable than panning because the user may need to close the input area to get at and interact with parts of the window.
adjustNothing	0x30	Don't resize or pan the window to make room for the soft input area; the window is never adjusted for it.

This corresponds to the global attribute resource symbol [windowSoftInputMode](#)  
[\(/reference/android/R.attr.html#windowSoftInputMode\)](#).

Constant Value: 20 (0x00000014)

#### **public static final int `AndroidManifestApplication_allowBackup`**

Whether to allow the application to participate in the backup and restore infrastructure. If this attribute is set to false, no backup or restore of the application will ever be performed, even by a full-system backup that would otherwise cause all application data to be saved via adb. The default value of this attribute is true.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*] *type:*  
name") or theme attribute (in the form "?[*package:*] [*type:*] *name*") containing a value of this type.

This corresponds to the global attribute resource symbol [allowBackup](#)  
(/reference/android/R.attr.html#allowBackup).

Constant Value: 17 (0x00000011)

#### public static final int **AndroidManifestApplication\_allowClearUserData**

Option to let applications specify that user data can/cannot be cleared. This flag is turned on by default.  
*This attribute is usable only by applications included in the system image. Third-party apps cannot use it.*

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*] *type:*  
name") or theme attribute (in the form "?[*package:*] [*type:*] *name*") containing a value of this type.

This corresponds to the global attribute resource symbol [allowClearUserData](#)  
(/reference/android/R.attr.html#allowClearUserData).

Constant Value: 5 (0x00000005)

#### public static final int **AndroidManifestApplication\_allowTaskReparenting**

Specify that an activity can be moved out of a task it is in to the task it has an affinity for when appropriate. Use with the application tag (to supply a default for all activities in the application), or with an activity tag (to supply a specific setting for that component).

Normally when an application is started, it is associated with the task of the activity that started it and stays there for its entire lifetime. You can use the allowTaskReparenting feature to force an activity to be re-parented to a different task when the task it is in goes to the background. Typically this is used to cause the activities of an application to move back to the main task associated with that application. The activity is re-parented to the task with the same [taskAffinity](#)  
(/reference/android/R.attr.html#taskAffinity) as it has.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*] *type:*  
name") or theme attribute (in the form "?[*package:*] [*type:*] *name*") containing a value of this type.

This corresponds to the global attribute resource symbol [allowTaskReparenting](#)  
(/reference/android/R.attr.html#allowTaskReparenting).

Constant Value: 14 (0x0000000e)

#### public static final int **AndroidManifestApplication\_backupAgent**

The name of the class subclassing BackupAgent to manage backup and restore of the application's data on external storage.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[*package:*] *type:*  
name") or theme attribute (in the form "?[*package:*] [*type:*] *name*") containing a value of this type.

This corresponds to the global attribute resource symbol [backupAgent](#)  
(/reference/android/R.attr.html#backupAgent).

Constant Value: 16 (0x00000010)

#### public static final int **AndroidManifestApplication\_debuggable**

Flag indicating whether the application can be debugged, even when running on a device that is running in user mode.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*] *type:*  
name") or theme attribute (in the form "?[*package:*] [*type:*] *name*") containing a value of this type.

This corresponds to the global attribute resource symbol [debuggable](#)  
(/reference/android/R.attr.html#debuggable).

Constant Value: 10 (0x0000000a)

#### public static final int **AndroidManifestApplication\_description**

Descriptive text for the associated data.

Must be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

This corresponds to the global attribute resource symbol [description](#)  
(/reference/android/R.attr.html#description).

Constant Value: 13 (0x0000000d)

#### **public static final int `AndroidManifestApplication_enabled`**

Specify whether the components in this application are enabled or not (that is, can be instantiated by the system). If "false", it overrides any component specific values (a value of "true" will not override the component specific values).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package*:]*type*:*name*") or theme attribute (in the form "?[*package*:]*type*:*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [enabled](#)  
(/reference/android/R.attr.html#enabled).

Constant Value: 9 (0x00000009)

#### **public static final int `AndroidManifestApplication_hardwareAccelerated`**

Flag indicating whether the application's rendering should be hardware accelerated if possible. This flag is turned on by default for applications that are targeting [ICE\\_CREAM\\_SANDWICH](#)  
(/reference/android/os/Build.VERSION\_CODES.html#ICE\_CREAM\_SANDWICH) or later.

This flag can be set on the application and any activity declared in the manifest. When enabled for the application, each activity is automatically assumed to be hardware accelerated. This flag can be overridden in the activity tags, either turning it off (if on for the application) or on (if off for the application.)

When this flag is turned on for an activity (either directly or via the application tag), every window created from the activity, including the activity's own window, will be hardware accelerated, if possible.

Please refer to the documentation of [FLAG\\_HARDWARE\\_ACCELERATED](#)  
(/reference/android/view/WindowManager.LayoutParams.html#FLAG\_HARDWARE\_ACCELERATED) for more information on how to control this flag programmatically.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package*:]*type*:*name*") or theme attribute (in the form "?[*package*:]*type*:*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [hardwareAccelerated](#)  
(/reference/android/R.attr.html#hardwareAccelerated).

Constant Value: 23 (0x00000017)

#### **public static final int `AndroidManifestApplication_hasCode`**

Indicate whether this application contains code. If set to false, there is no code associated with it and thus the system will not try to load its code when launching components. The default is true for normal behavior.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package*:]*type*:*name*") or theme attribute (in the form "?[*package*:]*type*:*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [hasCode](#)  
(/reference/android/R.attr.html#hasCode).

Constant Value: 7 (0x00000007)

#### **public static final int `AndroidManifestApplication_icon`**

A Drawable resource providing a graphical representation of its associated item. Use with the application tag (to supply a default icon for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific icon for that component). It may also be used with the intent-filter tag to supply an icon to show to the user when an activity is being selected based on a particular Intent.

The given icon will be used to display to the user a graphical representation of its associated component;

for example, as the icon for main activity that is displayed in the launcher. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ][ type: ] name`".

This corresponds to the global attribute resource symbol [icon](#) ([/reference/android/R.attr.html#icon](#)).

Constant Value: 2 (0x00000002)

#### **public static final int [AndroidManifestApplication\\_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 usually 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.

Must be a boolean value, either "`true`" or "`false`".

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ][ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [killAfterRestore](#) ([/reference/android/R.attr.html#killAfterRestore](#)).

Constant Value: 18 (0x00000012)

#### **public static final int [AndroidManifestApplication\\_label](#)**

A user-legible name for the given item. Use with the application tag (to supply a default label for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific label for that component). It may also be used with the intent-filter tag to supply a label to show to the user when an activity is being selected based on a particular Intent.

The given label will be used wherever the user sees information about its associated component; for example, as the name of a main activity that is displayed in the launcher. You should generally set this to a reference to a string resource, so that it can be localized, however it is also allowed to supply a plain string for quick and dirty programming.

May be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ][ type: ] name`".

May be a string value, using '\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This corresponds to the global attribute resource symbol [label](#) ([/reference/android/R.attr.html#label](#)).

Constant Value: 1 (0x00000001)

#### **public static final int [AndroidManifestApplication\\_largeHeap](#)**

Request that your application's processes be created with a large Dalvik heap. This applies to *all* processes created for the application. It only applies to the first application loaded into a process; if using a sharedUserId to allow multiple applications to use a process, they all must use this option consistently or will get unpredictable results.

Must be a boolean value, either "`true`" or "`false`".

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ][ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [largeHeap](#) ([/reference/android/R.attr.html#largeHeap](#)).

Constant Value: 24 (0x00000018)

#### **public static final int [AndroidManifestApplication\\_logo](#)**

A Drawable resource providing an extended graphical logo for its associated item. Use with the application tag (to supply a default logo for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific logo for that component). It may also be used with the intent-filter tag to supply a logo to show to the user when an activity is being selected based on a particular Intent.

The given logo will be used to display to the user a graphical representation of its associated component; for example as the header in the Action Bar. The primary differences between an icon and a logo are that

Logos are often wider and more detailed, and are used without an accompanying text caption. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ][ type: ] name`".

This corresponds to the global attribute resource symbol [logo](#) ([/reference/android/R.attr.html#logo](#)).

Constant Value: 22 (0x00000016)

#### **public static final int `AndroidManifestApplication_manageSpaceActivity`**

Name of activity to be launched for managing the application's space on the device.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ][ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [manageSpaceActivity](#) ([/reference/android/R.attr.html#manageSpaceActivity](#)).

Constant Value: 4 (0x00000004)

#### **public static final int `AndroidManifestApplication_name`**

An optional name of a class implementing the overall [Application](#) ([/reference/android/app/Application.html](#)) for this package. When the process for your package is started, this class is instantiated before any of the other application components. Note that this is not required, and in fact most applications will probably not need it.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ][ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) ([/reference/android/R.attr.html#name](#)).

Constant Value: 3 (0x00000003)

#### **public static final int `AndroidManifestApplication_permission`**

Specify a permission that a client is required to have in order to use the associated object. If the client does not hold the named permission, its request will fail. See the [Security and Permissions](#) ([/guide/topics/security/security.html](#)) document for more information on permissions.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ][ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [permission](#) ([/reference/android/R.attr.html#permission](#)).

Constant Value: 6 (0x00000006)

#### **public static final int `AndroidManifestApplication_persistent`**

Flag to control special persistent mode of an application. This should not normally be used by applications; it requires that the system keep your application running at all times.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ][ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [persistent](#) ([/reference/android/R.attr.html#persistent](#)).

Constant Value: 8 (0x00000008)

#### **public static final int `AndroidManifestApplication_process`**

Specify a specific process that the associated code is to run in. Use with the application tag (to supply a default process for all application components), or with the activity, receiver, service, or provider tag (to supply a specific icon for that component).

Application components are normally run in a single process that is created for the entire application. You can use this tag to modify where they run. If the process name begins with a ':' character, a new process

private to that application will be created when needed to run that component (allowing you to spread your application across multiple processes). If the process name begins with a lower-case character, the component will be run in a global process of that name, provided that you have permission to do so, allowing multiple applications to share one process to reduce resource usage.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [process](#)  
[\(/reference/android/R.attr.html#process\)](#).

Constant Value: 11 (0x0000000b)

#### public static final int **AndroidManifestApplication\_restoreAnyVersion**

Indicate that the application is prepared to attempt a restore of any backed-up dataset, even if the backup is apparently from 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.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [restoreAnyVersion](#)  
[\(/reference/android/R.attr.html#restoreAnyVersion\)](#).

Constant Value: 21 (0x00000015)

#### public static final int **AndroidManifestApplication\_restoreNeedsApplication**

This constant is deprecated.

This attribute is not used by the Android operating system.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [restoreNeedsApplication](#)  
[\(/reference/android/R.attr.html#restoreNeedsApplication\)](#).

Constant Value: 19 (0x00000013)

#### public static final int **AndroidManifestApplication\_supportsRtl**

Declare that your application will be able to deal with RTL (right to left) layouts. If set to false (default value), your application will not care about RTL layouts.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [supportsRtl](#)  
[\(/reference/android/R.attr.html#supportsRtl\)](#).

Constant Value: 26 (0x0000001a)

#### public static final int **AndroidManifestApplication\_taskAffinity**

Specify a task name that activities have an "affinity" to. Use with the application tag (to supply a default affinity for all activities in the application), or with the activity tag (to supply a specific affinity for that component).

The default value for this attribute is the same as the package name, indicating that all activities in the manifest should generally be considered a single "application" to the user. You can use this attribute to modify that behavior: either giving them an affinity for another task, if the activities are intended to be part of that task from the user's perspective, or using an empty string for activities that have no affinity to a task.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in

the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [taskAffinity](#)  
(/reference/android/R.attr.html#taskAffinity).

Constant Value: 12 (0x0000000c)

#### public static final int **AndroidManifestApplication\_testOnly**

Option to indicate this application is only for testing purposes. For example, it may expose functionality or data outside of itself that would cause a security hole, but is useful for testing. This kind of application can not be installed without the INSTALL\_ALLOW\_TEST flag, which means only through adb install.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [testOnly](#)  
(/reference/android/R.attr.html#testOnly).

Constant Value: 15 (0x0000000f)

#### public static final int **AndroidManifestApplication\_theme**

The overall theme to use for an activity. Use with either the application tag (to supply a default theme for all activities) or the activity tag (to supply a specific theme for that activity).

This automatically sets your activity's Context to use this theme, and may also be used for "starting" animations prior to the activity being launched (to better match what the activity actually looks like). It is a reference to a style resource defining the theme. If not set, the default system theme will be used.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [theme](#)  
(/reference/android/R.attr.html#theme).

Constant Value: 0 (0x00000000)

#### public static final int **AndroidManifestApplication\_uiOptions**

Extra options for an activity's UI. Applies to either the <activity> or <application> tag. If specified on the <application> tag these will be considered defaults for all activities in the application.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
none	0	No extra UI options. This is the default.
splitActionBarWhenNarrow	1	Split the options menu into a separate bar at the bottom of the screen when severely constrained for horizontal space. (e.g. portrait mode on a phone.) Instead of a small number of action buttons appearing in the action bar at the top of the screen, the action bar will split into the top navigation section and the bottom menu section. Menu items will not be split across the two bars; they will always appear together.

This corresponds to the global attribute resource symbol [uiOptions](#)  
(/reference/android/R.attr.html#uiOptions).

Constant Value: 25 (0x00000019)

#### public static final int **AndroidManifestApplication\_vmSafeMode**

Flag indicating whether the application requests the VM to operate in the safe mode.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [vmSafeMode](#)  
(/reference/android/R.attr.html#vmSafeMode).

Constant Value: 20 (0x00000014)

#### public static final int **AndroidManifestCategory\_name**

The name of category that is handled, using the Java-style naming convention. For example, to support [Intent.CATEGORY\\_LAUNCHER](#) you would put android.intent.category.LAUNCHER here. Custom actions should generally use a prefix matching the package name.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#).

Constant Value: 0 (0x00000000)

#### **public static final int AndroidManifestCompatibleScreensScreenDensity**

Specifies a compatible screen density, as per the device configuration screen density bins.

May be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

May be one of the following constant values.

Constant Value	Description
ldpi	120 A low density screen, approximately 120dpi.
mdpi	160 A medium density screen, approximately 160dpi.
hdpi	240 A high density screen, approximately 240dpi.
xhdpi	320 An extra high density screen, approximately 320dpi.

This corresponds to the global attribute resource symbol [screenDensity](#).

Constant Value: 1 (0x00000001)

#### **public static final int AndroidManifestCompatibleScreensScreenSize**

Specifies a compatible screen size, as per the device configuration screen size bins.

Must be one of the following constant values.

Constant Value	Description
small	200 A small screen configuration, at least 240x320dp.
normal	300 A normal screen configuration, at least 320x480dp.
large	400 A large screen configuration, at least 400x530dp.
xlarge	500 An extra large screen configuration, at least 600x800dp.

This corresponds to the global attribute resource symbol [screenSize](#).

Constant Value: 0 (0x00000000)

#### **public static final int AndroidManifestData\_host**

Specify a URI authority host that is handled, as per [IntentFilter.addDataAuthority\(\)](#).

*Note: host name matching in the Android framework is case-sensitive, unlike the formal RFC. As a result, host names here should always use lower case letters.*

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [host](#).

Constant Value: 2 (0x00000002)

#### **public static final int AndroidManifestData\_mimeType**

Specify a MIME type that is handled, as per [IntentFilter.addDataType\(\)](#).

*Note: MIME type matching in the Android framework is case-sensitive, unlike formal RFC MIME types. As a result,*

*MIME types here should always use lower case letters.*

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [mime\\_type](#)  
(/reference/android/R.attr.html#mime\_type).

Constant Value: 0 (0x00000000)

#### **public static final int AndroidManifestData\_path**

Specify a URI path that must exactly match, as per [IntentFilter.addDataAuthority\(\)](#)  
(/reference/android/content/IntentFilter.html#addDataPath(java.lang.String, int)) with  
[PATTERN\\_LITERAL](#) (/reference/android/os/PatternMatcher.html#PATTERN\_LITERAL).

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [path](#) (/reference/android/R.attr.html#path).

Constant Value: 4 (0x00000004)

#### **public static final int AndroidManifestData\_pathPattern**

Specify a URI path that matches a simple pattern, as per [IntentFilter.addDataAuthority\(\)](#)  
(/reference/android/content/IntentFilter.html#addDataPath(java.lang.String, int)) with  
[PATTERN\\_SIMPLE\\_GLOB](#) (/reference/android/os/PatternMatcher.html#PATTERN\_SIMPLE\_GLOB). Note that because '\\' is used as an escape character when reading the string from XML (before it is parsed as a pattern), you will need to double-escape: for example a literal "\*" would be written as "\\\*" and a literal "\\" would be written as "\\\\". This is basically the same as what you would need to write if constructing the string in Java code.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [pathPattern](#)  
(/reference/android/R.attr.html#pathPattern).

Constant Value: 6 (0x00000006)

#### **public static final int AndroidManifestData\_pathPrefix**

Specify a URI path that must be a prefix to match, as per [IntentFilter.addDataAuthority\(\)](#)  
(/reference/android/content/IntentFilter.html#addDataPath(java.lang.String, int)) with  
[PATTERN\\_PREFIX](#) (/reference/android/os/PatternMatcher.html#PATTERN\_PREFIX).

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [pathPrefix](#)  
(/reference/android/R.attr.html#pathPrefix).

Constant Value: 5 (0x00000005)

#### **public static final int AndroidManifestData\_port**

Specify a URI authority port that is handled, as per [IntentFilter.addDataAuthority\(\)](#)  
(/reference/android/content/IntentFilter.html#addDataAuthority(java.lang.String,  
java.lang.String)). If a host is supplied but not a port, any port is matched.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [port](#) (/reference/android/R.attr.html#port).

Constant Value: 3 (0x00000003)

**public static final int AndroidManifestData\_scheme**

Specify a URI scheme that is handled, as per [IntentFilter.addDataScheme\(\)](#)  
[\(/reference/android/content/IntentFilter.html#addDataScheme\(java.lang.String\)\)](#).

*Note: scheme matching in the Android framework is case-sensitive, unlike the formal RFC. As a result, schemes here should always use lower case letters.*

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [scheme](#)  
[\(/reference/android/R.attr.html#scheme\)](#).

Constant Value: 1 (0x00000001)

**public static final int AndroidManifestGrantUriPermission\_path**

Specify a URI path that must exactly match, as per [PatternMatcher](#)  
[\(/reference/android/os/PatternMatcher.html\)](#) With [PATTERN\\_LITERAL](#)  
[\(/reference/android/os/PatternMatcher.html#PATTERN\\_LITERAL\)](#).

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [path](#) [\(/reference/android/R.attr.html#path\)](#).

Constant Value: 0 (0x00000000)

**public static final int AndroidManifestGrantUriPermission\_pathPattern**

Specify a URI path that matches a simple pattern, as per [PatternMatcher](#)  
[\(/reference/android/os/PatternMatcher.html\)](#) With [PATTERN\\_SIMPLE\\_GLOB](#)  
[\(/reference/android/os/PatternMatcher.html#PATTERN\\_SIMPLE\\_GLOB\)](#). Note that because '\' is used as an escape character when reading the string from XML (before it is parsed as a pattern), you will need to double-escape: for example a literal "\*" would be written as "\\\*" and a literal "\" would be written as "\\\\". This is basically the same as what you would need to write if constructing the string in Java code.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [pathPattern](#)  
[\(/reference/android/R.attr.html#pathPattern\)](#).

Constant Value: 2 (0x00000002)

**public static final int AndroidManifestGrantUriPermission\_pathPrefix**

Specify a URI path that must be a prefix to match, as per [PatternMatcher](#)  
[\(/reference/android/os/PatternMatcher.html\)](#) With [PATTERN\\_PREFIX](#)  
[\(/reference/android/os/PatternMatcher.html#PATTERN\\_PREFIX\)](#).

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [pathPrefix](#)  
[\(/reference/android/R.attr.html#pathPrefix\)](#).

Constant Value: 1 (0x00000001)

**public static final int AndroidManifestInstrumentation\_functionalTest**

Flag indicating that an Instrumentation class should be run as a functional test.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [functionalTest](#)

[\(/reference/android/R.attr.html#functionalTest\)](#).

Constant Value: 5 (0x00000005)

#### public static final int **AndroidManifestInstrumentation\_handleProfiling**

Flag indicating that an Instrumentation class wants to take care of starting/stopping profiling itself, rather than relying on the default behavior of profiling the complete time it is running. This allows it to target profiling data at a specific set of operations.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [handleProfiling](#)

[\(/reference/android/R.attr.html#handleProfiling\)](#).

Constant Value: 4 (0x00000004)

#### public static final int **AndroidManifestInstrumentation\_icon**

A Drawable resource providing a graphical representation of its associated item. Use with the application tag (to supply a default icon for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific icon for that component). It may also be used with the intent-filter tag to supply an icon to show to the user when an activity is being selected based on a particular Intent.

The given icon will be used to display to the user a graphical representation of its associated component; for example, as the icon for main activity that is displayed in the launcher. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [icon](#) [\(/reference/android/R.attr.html#icon\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **AndroidManifestInstrumentation\_label**

A user-legible name for the given item. Use with the application tag (to supply a default label for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific label for that component). It may also be used with the intent-filter tag to supply a label to show to the user when an activity is being selected based on a particular Intent.

The given label will be used wherever the user sees information about its associated component; for example, as the name of a main activity that is displayed in the launcher. You should generally set this to a reference to a string resource, so that it can be localized, however it is also allowed to supply a plain string for quick and dirty programming.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This corresponds to the global attribute resource symbol [label](#)

[\(/reference/android/R.attr.html#label\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **AndroidManifestInstrumentation\_logo**

A Drawable resource providing an extended graphical logo for its associated item. Use with the application tag (to supply a default logo for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific logo for that component). It may also be used with the intent-filter tag to supply a logo to show to the user when an activity is being selected based on a particular Intent.

The given logo will be used to display to the user a graphical representation of its associated component; for example as the header in the Action Bar. The primary differences between an icon and a logo are that logos are often wider and more detailed, and are used without an accompanying text caption. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [logo](#) [\(/reference/android/R.attr.html#logo\)](#).

Constant Value: 6 (0x00000006)

#### public static final int **AndroidManifestInstrumentation\_name**

Required name of the class implementing the instrumentation, deriving from [Instrumentation](#). This is a fully qualified class name (for example, com.mycompany.myapp.MyActivity); as a short-hand if the first character of the class is a period then it is appended to your package name.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#).

Constant Value: 2 (0x00000002)

#### public static final int **AndroidManifestInstrumentation\_targetPackage**

The name of the application package that an Instrumentation object will run against.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [targetPackage](#).

Constant Value: 3 (0x00000003)

#### public static final int **AndroidManifestIntentFilter\_icon**

A Drawable resource providing a graphical representation of its associated item. Use with the application tag (to supply a default icon for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific icon for that component). It may also be used with the intent-filter tag to supply an icon to show to the user when an activity is being selected based on a particular Intent.

The given icon will be used to display to the user a graphical representation of its associated component; for example, as the icon for main activity that is displayed in the launcher. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:]type:name".

This corresponds to the global attribute resource symbol [icon](#).

Constant Value: 1 (0x00000001)

#### public static final int **AndroidManifestIntentFilter\_label**

A user-legible name for the given item. Use with the application tag (to supply a default label for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific label for that component). It may also be used with the intent-filter tag to supply a label to show to the user when an activity is being selected based on a particular Intent.

The given label will be used wherever the user sees information about its associated component; for example, as the name of a main activity that is displayed in the launcher. You should generally set this to a reference to a string resource, so that it can be localized, however it is also allowed to supply a plain string for quick and dirty programming.

May be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:]type:name".

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This corresponds to the global attribute resource symbol [label](#).

Constant Value: 0 (0x00000000)

#### public static final int **AndroidManifestIntentFilter\_logo**

A Drawable resource providing an extended graphical logo for its associated item. Use with the application tag (to supply a default logo for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific logo for that component). It may also be used with the

intent-filter tag to supply a logo to show to the user when an activity is being selected based on a particular Intent.

The given logo will be used to display to the user a graphical representation of its associated component; for example as the header in the Action Bar. The primary differences between an icon and a logo are that logos are often wider and more detailed, and are used without an accompanying text caption. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [logo](#) ([/reference/android/R.attr.html#logo](#)).

Constant Value: 3 (0x00000003)

#### **public static final int AndroidManifestIntentFilter\_priority**

Specify the relative importance or ability in handling a particular Intent. For receivers, this controls the order in which they are executed to receive a broadcast (note that for asynchronous broadcasts, this order is ignored). For activities, this provides information about how good an activity is handling an Intent; when multiple activities match an intent and have different priorities, only those with the higher priority value will be considered a match.

Only use if you really need to impose some specific order in which the broadcasts are received, or want to forcibly place an activity to always be preferred over others. The value is a single integer, with higher numbers considered to be better.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [priority](#) ([/reference/android/R.attr.html#priority](#)).

Constant Value: 2 (0x00000002)

#### **public static final int AndroidManifestMetaData\_name**

A unique name for the given item. This must use a Java-style naming convention to ensure the name is unique, for example "com.mycompany.MyName".

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) ([/reference/android/R.attr.html#name](#)).

Constant Value: 0 (0x00000000)

#### **public static final int AndroidManifestMetaData\_resource**

Resource identifier to assign to this piece of named meta-data. The resource identifier can later be retrieved from the meta data Bundle through [Bundle.getInt](#) ([/reference/android/os/Bundle.html#getInt\(java.lang.String\)](#)).

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [resource](#) ([/reference/android/R.attr.html#resource](#)).

Constant Value: 2 (0x00000002)

#### **public static final int AndroidManifestMetaData\_value**

Concrete value to assign to this piece of named meta-data. The data can later be retrieved from the meta data Bundle through [Bundle.getString](#) ([/reference/android/os/Bundle.html#getString\(java.lang.String\)](#)), [Bundle.getInt](#) ([/reference/android/os/Bundle.html#getInt\(java.lang.String\)](#)), [Bundle.getBoolean](#) ([/reference/android/os/Bundle.html#getBoolean\(java.lang.String\)](#)), or [Bundle.getFloat](#) ([/reference/android/os/Bundle.html#getFloat\(java.lang.String\)](#)) depending on the type used here.

May be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

May be an integer value, such as "100".

May be a boolean value, either "true" or "false".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

May be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [value](#)  
(/reference/android/R.attr.html#value).

Constant Value: 1 (0x00000001)

#### **public static final int [AndroidManifestOriginalPackage\\_name](#)**

A unique name for the given item. This must use a Java-style naming convention to ensure the name is unique, for example "com.mycompany.MyName".

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) (/reference/android/R.attr.html#name).

Constant Value: 0 (0x00000000)

#### **public static final int [AndroidManifestPackageVerifier\\_name](#)**

Specifies the Java-style package name that defines this package verifier.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) (/reference/android/R.attr.html#name).

Constant Value: 0 (0x00000000)

#### **public static final int [AndroidManifestPackageVerifier\\_publicKey](#)**

The Base64 encoded public key of the package verifier's signature.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [publicKey](#)  
(/reference/android/R.attr.html#publicKey).

Constant Value: 1 (0x00000001)

#### **public static final int [AndroidManifestPathPermission\\_path](#)**

Specify a URI path that must exactly match, as per [PatternMatcher](#)  
(/reference/android/os/PatternMatcher.html) With [PATTERN\\_LITERAL](#)  
(/reference/android/os/PatternMatcher.html#PATTERN\_LITERAL).

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [path](#) (/reference/android/R.attr.html#path).

Constant Value: 3 (0x00000003)

#### **public static final int [AndroidManifestPathPermission\\_pathPattern](#)**

Specify a URI path that matches a simple pattern, as per [PatternMatcher](#)  
(/reference/android/os/PatternMatcher.html) With [PATTERN\\_SIMPLE\\_GLOB](#)  
(/reference/android/os/PatternMatcher.html#PATTERN\_SIMPLE\_GLOB). Note that because '\\' is used as an escape character when reading the string from XML (before it is parsed as a pattern), you will need to double-escape: for example a literal "\*" would be written as "\\\*" and a literal "/" would be written as "\\\\". This is basically the same as what you would need to write if constructing the string in Java code.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [pathPattern](#)  
(/reference/android/R.attr.html#pathPattern).

Constant Value: 5 (0x00000005)

#### public static final int **AndroidManifestPathPermission\_pathPrefix**

Specify a URI path that must be a prefix to match, as per [PatternMatcher](#)  
(/reference/android/os/PatternMatcher.html) with [PATTERN\\_PREFIX](#)  
(/reference/android/os/PatternMatcher.html#PATTERN\_PREFIX).

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [pathPrefix](#)  
(/reference/android/R.attr.html#pathPrefix).

Constant Value: 4 (0x00000004)

#### public static final int **AndroidManifestPathPermission\_permission**

Specify a permission that a client is required to have in order to use the associated object. If the client does not hold the named permission, its request will fail. See the [Security and Permissions](#)  
(/guide/topics/security/security.html) document for more information on permissions.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [permission](#)  
(/reference/android/R.attr.html#permission).

Constant Value: 0 (0x00000000)

#### public static final int **AndroidManifestPathPermission\_readPermission**

A specific [permission](#) (/reference/android/R.attr.html#permission) name for read-only access to a [ContentProvider](#) (/reference/android/content/ContentProvider.html). See the [Security and Permissions](#) (/guide/topics/security/security.html) document for more information on permissions.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [readPermission](#)  
(/reference/android/R.attr.html#readPermission).

Constant Value: 1 (0x00000001)

#### public static final int **AndroidManifestPathPermission\_writePermission**

A specific [permission](#) (/reference/android/R.attr.html#permission) name for write access to a [ContentProvider](#) (/reference/android/content/ContentProvider.html). See the [Security and Permissions](#) (/guide/topics/security/security.html) document for more information on permissions.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [writePermission](#)  
(/reference/android/R.attr.html#writePermission).

Constant Value: 2 (0x00000002)

#### public static final int **AndroidManifestPermissionGroup\_description**

Descriptive text for the associated data.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [description](#) ([/reference/android/R.attr.html#description](#)).

Constant Value: 4 (0x00000004)

#### public static final int **AndroidManifestPermissionGroup\_icon**

A Drawable resource providing a graphical representation of its associated item. Use with the application tag (to supply a default icon for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific icon for that component). It may also be used with the intent-filter tag to supply an icon to show to the user when an activity is being selected based on a particular Intent.

The given icon will be used to display to the user a graphical representation of its associated component; for example, as the icon for main activity that is displayed in the launcher. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[+][ package: ] type: name" or to a theme attribute in the form "?[ package: ][ type: ] name".

This corresponds to the global attribute resource symbol [icon](#) ([/reference/android/R.attr.html#icon](#)).

Constant Value: 1 (0x00000001)

#### public static final int **AndroidManifestPermissionGroup\_label**

A user-legible name for the given item. Use with the application tag (to supply a default label for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific label for that component). It may also be used with the intent-filter tag to supply a label to show to the user when an activity is being selected based on a particular Intent.

The given label will be used wherever the user sees information about its associated component; for example, as the name of a main activity that is displayed in the launcher. You should generally set this to a reference to a string resource, so that it can be localized, however it is also allowed to supply a plain string for quick and dirty programming.

May be a reference to another resource, in the form "@[+][ package: ] type: name" or to a theme attribute in the form "?[ package: ][ type: ] name".

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This corresponds to the global attribute resource symbol [label](#) ([/reference/android/R.attr.html#label](#)).

Constant Value: 0 (0x00000000)

#### public static final int **AndroidManifestPermissionGroup\_logo**

A Drawable resource providing an extended graphical logo for its associated item. Use with the application tag (to supply a default logo for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific logo for that component). It may also be used with the intent-filter tag to supply a logo to show to the user when an activity is being selected based on a particular Intent.

The given logo will be used to display to the user a graphical representation of its associated component; for example as the header in the Action Bar. The primary differences between an icon and a logo are that logos are often wider and more detailed, and are used without an accompanying text caption. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[+][ package: ] type: name" or to a theme attribute in the form "?[ package: ][ type: ] name".

This corresponds to the global attribute resource symbol [logo](#) ([/reference/android/R.attr.html#logo](#)).

Constant Value: 5 (0x00000005)

#### public static final int **AndroidManifestPermissionGroup\_name**

Required public name of the permission group, permissions will use to specify the group they are in. This is a string using Java-style scoping to ensure it is unique. The prefix will often be the same as our overall package name, for example "com.mycompany.android.myapp.SomePermission".

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ][ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) ([/reference/android/R.attr.html#name](#)).

Constant Value: 2 (0x00000002)

#### public static final int **AndroidManifestPermissionGroup\_permissionGroupFlags**

Flags indicating more context for a permission group.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
	Set to indicate that this permission group contains permissions protecting personalInfo 0x0001 access to some information that is considered personal to the user (such as contacts, e-mails, etc).	

This corresponds to the global attribute resource symbol [permissionGroupFlags](#)  
[\(/reference/android/R.attr.html#permissionGroupFlags\)](#).

Constant Value: 6 (0x00000006)

#### public static final int **AndroidManifestPermissionGroup\_priority**

Specify the relative importance or ability in handling a particular Intent. For receivers, this controls the order in which they are executed to receive a broadcast (note that for asynchronous broadcasts, this order is ignored). For activities, this provides information about how good an activity is handling an Intent; when multiple activities match an intent and have different priorities, only those with the higher priority value will be considered a match.

Only use if you really need to impose some specific order in which the broadcasts are received, or want to forcibly place an activity to always be preferred over others. The value is a single integer, with higher numbers considered to be better.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [priority](#)  
[\(/reference/android/R.attr.html#priority\)](#).

Constant Value: 3 (0x00000003)

#### public static final int **AndroidManifestPermissionTree\_icon**

A Drawable resource providing a graphical representation of its associated item. Use with the application tag (to supply a default icon for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific icon for that component). It may also be used with the intent-filter tag to supply an icon to show to the user when an activity is being selected based on a particular Intent.

The given icon will be used to display to the user a graphical representation of its associated component; for example, as the icon for main activity that is displayed in the launcher. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [icon](#) [\(/reference/android/R.attr.html#icon\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **AndroidManifestPermissionTree\_label**

A user-legible name for the given item. Use with the application tag (to supply a default label for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific label for that component). It may also be used with the intent-filter tag to supply a label to show to the user when an activity is being selected based on a particular Intent.

The given label will be used wherever the user sees information about its associated component; for example, as the name of a main activity that is displayed in the launcher. You should generally set this to a reference to a string resource, so that it can be localized, however it is also allowed to supply a plain string for quick and dirty programming.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This corresponds to the global attribute resource symbol [label](#)  
[\(/reference/android/R.attr.html#label\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int AndroidManifestPermissionTree\_logo**

A Drawable resource providing an extended graphical logo for its associated item. Use with the application tag (to supply a default logo for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific logo for that component). It may also be used with the intent-filter tag to supply a logo to show to the user when an activity is being selected based on a particular Intent.

The given logo will be used to display to the user a graphical representation of its associated component; for example as the header in the Action Bar. The primary differences between an icon and a logo are that logos are often wider and more detailed, and are used without an accompanying text caption. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [logo](#) ([/reference/android/R.attr.html#logo](#)).

Constant Value: 3 (0x00000003)

#### **public static final int AndroidManifestPermissionTree\_name**

Required public name of the permission tree, which is the base name of all permissions under it. This is a string using Java-style scoping to ensure it is unique. The prefix will often be the same as our overall package name, for example "com.mycompany.android.myapp.SomePermission". A permission tree name must have more than two segments in its path; that is, "com.me.foo" is okay, but not "com.me" or "com".

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) ([/reference/android/R.attr.html#name](#)).

Constant Value: 2 (0x00000002)

#### **public static final int AndroidManifestPermission\_description**

Descriptive text for the associated data.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [description](#) ([/reference/android/R.attr.html#description](#)).

Constant Value: 5 (0x00000005)

#### **public static final int AndroidManifestPermission\_icon**

A Drawable resource providing a graphical representation of its associated item. Use with the application tag (to supply a default icon for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific icon for that component). It may also be used with the intent-filter tag to supply an icon to show to the user when an activity is being selected based on a particular Intent.

The given icon will be used to display to the user a graphical representation of its associated component; for example, as the icon for main activity that is displayed in the launcher. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [icon](#) ([/reference/android/R.attr.html#icon](#)).

Constant Value: 1 (0x00000001)

#### **public static final int AndroidManifestPermission\_label**

A user-legible name for the given item. Use with the application tag (to supply a default label for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific label for that component). It may also be used with the intent-filter tag to supply a label to show to the user when an activity is being selected based on a particular Intent.

The given label will be used wherever the user sees information about its associated component; for example, as the name of a main activity that is displayed in the launcher. You should generally set this to a

reference to a string resource, so that it can be localized, however it is also allowed to supply a plain string for quick and dirty programming.

May be a reference to another resource, in the form "@[+][ package: ] type: name" or to a theme attribute in the form "?[ package: ][ type: ] name".

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This corresponds to the global attribute resource symbol [label](#)  
(/reference/android/R.attr.html#label).

Constant Value: 0 (0x00000000)

#### public static final int **AndroidManifestPermission\_logo**

A Drawable resource providing an extended graphical logo for its associated item. Use with the application tag (to supply a default logo for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific logo for that component). It may also be used with the intent-filter tag to supply a logo to show to the user when an activity is being selected based on a particular Intent.

The given logo will be used to display to the user a graphical representation of its associated component; for example as the header in the Action Bar. The primary differences between an icon and a logo are that logos are often wider and more detailed, and are used without an accompanying text caption. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[+][ package: ] type: name" or to a theme attribute in the form "?[ package: ][ type: ] name".

This corresponds to the global attribute resource symbol [logo](#) (/reference/android/R.attr.html#logo).

Constant Value: 6 (0x00000006)

#### public static final int **AndroidManifestPermission\_name**

Required public name of the permission, which other components and packages will use when referring to this permission. This is a string using Java-style scoping to ensure it is unique. The prefix will often be the same as our overall package name, for example "com.mycompany.android.myapp.SomePermission".

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ][ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) (/reference/android/R.attr.html#name).

Constant Value: 2 (0x00000002)

#### public static final int **AndroidManifestPermission\_permissionFlags**

Flags indicating more context for a permission.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
	Set to indicate that this permission allows an operation that may cost the user costsMoney 0x0001 money. Such permissions may be highlighted when shown to the user with this additional information.	

This corresponds to the global attribute resource symbol [permissionFlags](#)  
(/reference/android/R.attr.html#permissionFlags).

Constant Value: 7 (0x00000007)

#### public static final int **AndroidManifestPermission\_permissionGroup**

Specified the name of a group that this permission is associated with. The group must have been defined with the [permission-group](#) (/reference/android/R.styleable.html#AndroidManifestPermissionGroup) tag.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ][ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [permissionGroup](#)  
(/reference/android/R.attr.html#permissionGroup).

Constant Value: 4 (0x00000004)

## `public static final int AndroidManifestPermission_protectionLevel`

Characterizes the potential risk implied in a permission and indicates the procedure the system should follow when determining whether to grant the permission to an application requesting it. [Standard permissions](#) ([/reference/android/Manifest.permission.html](#)) have a predefined and permanent protectionLevel. If you are creating a custom permission in an application, you can define a protectionLevel attribute with one of the values listed below. If no protectionLevel is defined for a custom permission, the system assigns the default ("normal").

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
normal	0	A lower-risk permission that gives an application access to isolated application-level features, with minimal risk to other applications, the system, or the user. The system automatically grants this type of permission to a requesting application at installation, without asking for the user's explicit approval (though the user always has the option to review these permissions before installing).
dangerous	1	A higher-risk permission that would give a requesting application access to private user data or control over the device that can negatively impact the user. Because this type of permission introduces potential risk, the system may not automatically grant it to the requesting application. For example, any dangerous permissions requested by an application may be displayed to the user and require confirmation before proceeding, or some other approach may be taken to avoid the user automatically allowing the use of such facilities.
signature	2	A permission that the system is to grant only if the requesting application is signed with the same certificate as the application that declared the permission. If the certificates match, the system automatically grants the permission without notifying the user or asking for the user's explicit approval.
signatureOrSystem	3	A permission that the system is to grant only to packages in the Android system image or that are signed with the same certificates. Please avoid using this option, as the signature protection level should be sufficient for most needs and works regardless of exactly where applications are installed. This permission is used for certain special situations where multiple vendors have applications built in to a system image which need to share specific features explicitly because they are being built together.
system	0x10	Additional flag from base permission type: this permission can also be granted to any applications installed on the system image. Please avoid using this option, as the signature protection level should be sufficient for most needs and works regardless of exactly where applications are installed. This permission flag is used for certain special situations where multiple vendors have applications built in to a system image which need to share specific features explicitly because they are being built together.
development	0x20	Additional flag from base permission type: this permission can also (optionally) be granted to development applications.

This corresponds to the global attribute resource symbol [protectionLevel](#) ([/reference/android/R.attr.html#protectionLevel](#)).

Constant Value: 3 (0x00000003)

## `public static final int AndroidManifestProtectedBroadcast_name`

A unique name for the given item. This must use a Java-style naming convention to ensure the name is unique, for example "com.mycompany.MyName".

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) ([/reference/android/R.attr.html#name](#)).

Constant Value: 0 (0x00000000)

## `public static final int AndroidManifestProvider_authorities`

Specify the authorities under which this content provider can be found. Multiple authorities may be supplied by separating them with a semicolon. Authority names should use a Java-style naming convention (such as com.google.provider.MyProvider) in order to avoid conflicts. Typically this name is the same as the class implementation describing the provider's data structure.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in

the form "?[*package*: ][*type*: ]*name*" containing a value of this type.

This corresponds to the global attribute resource symbol [authorities](#) ([/reference/android/R.attr.html#authorities](#)).

Constant Value: 10 (0x0000000a)

#### **public static final int AndroidManifestProvider\_description**

Descriptive text for the associated data.

Must be a reference to another resource, in the form "@[+][*package*: ]*type*:*name*" or to a theme attribute in the form "?[*package*: ][*type*: ]*name*".

This corresponds to the global attribute resource symbol [description](#) ([/reference/android/R.attr.html#description](#)).

Constant Value: 14 (0x0000000e)

#### **public static final int AndroidManifestProvider\_enabled**

Specify whether this provider is enabled or not (that is, can be instantiated by the system). It can also be specified for an application as a whole, in which case a value of "false" will override any component specific values (a value of "true" will not override the component specific values).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*: ]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [enabled](#) ([/reference/android/R.attr.html#enabled](#)).

Constant Value: 6 (0x00000006)

#### **public static final int AndroidManifestProvider\_exported**

Flag indicating whether the given application component is available to other applications. If false, it can only be accessed by applications with its same user id (which usually means only by code in its own package). If true, it can be invoked by external entities, though which ones can do so may be controlled through permissions. The default value is false for activity, receiver, and service components that do not specify any intent filters; it is true for activity, receiver, and service components that do have intent filters (implying they expect to be invoked by others who do not know their particular component name) and for all content providers.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*: ]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [exported](#) ([/reference/android/R.attr.html#exported](#)).

Constant Value: 7 (0x00000007)

#### **public static final int AndroidManifestProvider\_grantUriPermissions**

If true, the [Context.grantUriPermission](#) ([/reference/android/content/Context.html#grantUriPermission\(java.lang.String, android.net.Uri, int\)](#)) or corresponding Intent flags can be used to allow others to access specific URLs in the content provider, even if they do not have an explicit read or write permission. If you are supporting this feature, you must be sure to call [Context.revokeUriPermission](#) ([/reference/android/content/Context.html#revokeUriPermission\(android.net.Uri, int\)](#)) when URLs are deleted from your provider.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*: ]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [grantUriPermissions](#) ([/reference/android/R.attr.html#grantUriPermissions](#)).

Constant Value: 13 (0x0000000d)

#### **public static final int AndroidManifestProvider\_icon**

A Drawable resource providing a graphical representation of its associated item. Use with the application

tag (to supply a default icon for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific icon for that component). It may also be used with the intent-filter tag to supply an icon to show to the user when an activity is being selected based on a particular Intent.

The given icon will be used to display to the user a graphical representation of its associated component; for example, as the icon for main activity that is displayed in the launcher. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

This corresponds to the global attribute resource symbol [icon](#) ([/reference/android/R.attr.html#icon](#)).

Constant Value: 1 (0x00000001)

#### **public static final int AndroidManifestProvider\_initOrder**

Specify the order in which content providers hosted by a process are instantiated when that process is created. Not needed unless you have providers with dependencies between each other, to make sure that they are created in the order needed by those dependencies. The value is a simple integer, with higher numbers being initialized first.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[+][*package*:]*type*:*name*"") or theme attribute (in the form "?[*package*:]*type*:*name*"") containing a value of this type.

This corresponds to the global attribute resource symbol [initOrder](#) ([/reference/android/R.attr.html#initOrder](#)).

Constant Value: 12 (0x0000000c)

#### **public static final int AndroidManifestProvider\_label**

A user-legible name for the given item. Use with the application tag (to supply a default label for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific label for that component). It may also be used with the intent-filter tag to supply a label to show to the user when an activity is being selected based on a particular Intent.

The given label will be used wherever the user sees information about its associated component; for example, as the name of a main activity that is displayed in the launcher. You should generally set this to a reference to a string resource, so that it can be localized, however it is also allowed to supply a plain string for quick and dirty programming.

May be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This corresponds to the global attribute resource symbol [label](#) ([/reference/android/R.attr.html#label](#)).

Constant Value: 0 (0x00000000)

#### **public static final int AndroidManifestProvider\_logo**

A Drawable resource providing an extended graphical logo for its associated item. Use with the application tag (to supply a default logo for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific logo for that component). It may also be used with the intent-filter tag to supply a logo to show to the user when an activity is being selected based on a particular Intent.

The given logo will be used to display to the user a graphical representation of its associated component; for example as the header in the Action Bar. The primary differences between an icon and a logo are that logos are often wider and more detailed, and are used without an accompanying text caption. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

This corresponds to the global attribute resource symbol [logo](#) ([/reference/android/R.attr.html#logo](#)).

Constant Value: 15 (0x0000000f)

#### **public static final int AndroidManifestProvider\_multiprocess**

Specify whether a component is allowed to have multiple instances of itself running in different processes. Use with the activity and provider tags.

Normally the system will ensure that all instances of a particular component are only running in a single process. You can use this attribute to disable that behavior, allowing the system to create instances wherever they are used (provided permissions allow it). This is most often used with content providers, so that instances of a provider can be created in each client process, allowing them to be used without performing IPC.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [multiprocess](#)  
(/reference/android/R.attr.html#multiprocess).

Constant Value: 9 (0x00000009)

#### public static final int **AndroidManifestProvider\_name**

Required name of the class implementing the provider, deriving from [ContentProvider](#)  
(/reference/android/content/ContentProvider.html). This is a fully qualified class name (for example, com.mycompany.myapp.MyProvider); as a short-hand if the first character of the class is a period then it is appended to your package name.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) ([/reference/android/R.attr.html#name](#)).

Constant Value: 2 (0x00000002)

#### public static final int **AndroidManifestProvider\_permission**

Specify a permission that a client is required to have in order to use the associated object. If the client does not hold the named permission, its request will fail. See the [Security and Permissions](#)  
(/guide/topics/security/security.html) document for more information on permissions.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [permission](#)  
(/reference/android/R.attr.html#permission).

Constant Value: 3 (0x00000003)

#### public static final int **AndroidManifestProvider\_process**

Specify a specific process that the associated code is to run in. Use with the application tag (to supply a default process for all application components), or with the activity, receiver, service, or provider tag (to supply a specific icon for that component).

Application components are normally run in a single process that is created for the entire application. You can use this tag to modify where they run. If the process name begins with a ':' character, a new process private to that application will be created when needed to run that component (allowing you to spread your application across multiple processes). If the process name begins with a lower-case character, the component will be run in a global process of that name, provided that you have permission to do so, allowing multiple applications to share one process to reduce resource usage.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [process](#)  
(/reference/android/R.attr.html#process).

Constant Value: 8 (0x00000008)

#### public static final int **AndroidManifestProvider\_readPermission**

A specific [permission](#) ([/reference/android/R.attr.html#permission](#)) name for read-only access to a [ContentProvider](#) ([/reference/android/content/ContentProvider.html](#)). See the [Security and Permissions](#)  
(/guide/topics/security/security.html) document for more information on permissions.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [readPermission](#)  
(/reference/android/R.attr.html#readPermission).

Constant Value: 4 (0x00000004)

#### public static final int **AndroidManifestProvider\_singleUser**

If set to true, a single instance of this component will run for all users. That instance will run as user 0, the default/primary user. When the app running is in processes for other users and interacts with this component (by binding to a service for example) those processes will always interact with the instance running for user 0. Enabling single user mode forces "exported" of the component to be false, to help avoid introducing multi-user security bugs. This feature is only available to applications built in to the system image; you must hold the permission INTERACT\_ACROSS\_USERS in order to use this feature. This flag can only be used with services, receivers, and providers; it can not be used with activities.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [singleUser](#)  
(/reference/android/R.attr.html#singleUser).

Constant Value: 16 (0x00000010)

#### public static final int **AndroidManifestProvider\_syncable**

Flag indicating whether this content provider would like to participate in data synchronization.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [syncable](#)  
(/reference/android/R.attr.html#syncable).

Constant Value: 11 (0x0000000b)

#### public static final int **AndroidManifestProvider\_writePermission**

A specific [permission](#) (/reference/android/R.attr.html#permission) name for write access to a [ContentProvider](#) (/reference/android/content/ContentProvider.html). See the [Security and Permissions](#) (/guide/topics/security/security.html) document for more information on permissions.

Must be a string value, using '\\.' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [writePermission](#)  
(/reference/android/R.attr.html#writePermission).

Constant Value: 5 (0x00000005)

#### public static final int **AndroidManifestReceiver\_description**

Descriptive text for the associated data.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [description](#)  
(/reference/android/R.attr.html#description).

Constant Value: 7 (0x00000007)

#### public static final int **AndroidManifestReceiver\_enabled**

Specify whether the receiver is enabled or not (that is, can be instantiated by the system). It can also be specified for an application as a whole, in which case a value of "false" will override any component specific values (a value of "true" will not override the component specific values).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [enabled](#)  
(/reference/android/R.attr.html#enabled).

Constant Value: 4 (0x00000004)

#### public static final int **AndroidManifestReceiver\_exported**

Flag indicating whether the given application component is available to other applications. If false, it can only be accessed by applications with its same user id (which usually means only by code in its own package). If true, it can be invoked by external entities, though which ones can do so may be controlled through permissions. The default value is false for activity, receiver, and service components that do not specify any intent filters; it is true for activity, receiver, and service components that do have intent filters (implying they expect to be invoked by others who do not know their particular component name) and for all content providers.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [exported](#)  
(/reference/android/R.attr.html#exported).

Constant Value: 5 (0x00000005)

#### public static final int **AndroidManifestReceiver\_icon**

A Drawable resource providing a graphical representation of its associated item. Use with the application tag (to supply a default icon for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific icon for that component). It may also be used with the intent-filter tag to supply an icon to show to the user when an activity is being selected based on a particular Intent.

The given icon will be used to display to the user a graphical representation of its associated component; for example, as the icon for main activity that is displayed in the launcher. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [icon](#) (/reference/android/R.attr.html#icon).

Constant Value: 1 (0x00000001)

#### public static final int **AndroidManifestReceiver\_label**

A user-legible name for the given item. Use with the application tag (to supply a default label for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific label for that component). It may also be used with the intent-filter tag to supply a label to show to the user when an activity is being selected based on a particular Intent.

The given label will be used wherever the user sees information about its associated component; for example, as the name of a main activity that is displayed in the launcher. You should generally set this to a reference to a string resource, so that it can be localized, however it is also allowed to supply a plain string for quick and dirty programming.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a string value, using '\;' to escape characters such as '\n' or '\\xxxx' for a unicode character.

This corresponds to the global attribute resource symbol [label](#)  
(/reference/android/R.attr.html#label).

Constant Value: 0 (0x00000000)

#### public static final int **AndroidManifestReceiver\_logo**

A Drawable resource providing an extended graphical logo for its associated item. Use with the application tag (to supply a default logo for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific logo for that component). It may also be used with the intent-filter tag to supply a logo to show to the user when an activity is being selected based on a particular Intent.

The given logo will be used to display to the user a graphical representation of its associated component;

for example as the header in the Action Bar. The primary differences between an icon and a logo are that logos are often wider and more detailed, and are used without an accompanying text caption. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [logo](#) ([/reference/android/R.attr.html#logo](#)).

Constant Value: 8 (0x00000008)

#### **public static final int [AndroidManifestReceiver\\_name](#)**

Required name of the class implementing the receiver, deriving from [BroadcastReceiver](#) ([/reference/android/content/BroadcastReceiver.html](#)). This is a fully qualified class name (for example, com.mycompany.myapp.MyReceiver); as a short-hand if the first character of the class is a period then it is appended to your package name.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) ([/reference/android/R.attr.html#name](#)).

Constant Value: 2 (0x00000002)

#### **public static final int [AndroidManifestReceiver\\_permission](#)**

Specify a permission that a client is required to have in order to use the associated object. If the client does not hold the named permission, its request will fail. See the [Security and Permissions](#) ([/guide/topics/security/security.html](#)) document for more information on permissions.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [permission](#) ([/reference/android/R.attr.html#permission](#)).

Constant Value: 3 (0x00000003)

#### **public static final int [AndroidManifestReceiver\\_process](#)**

Specify a specific process that the associated code is to run in. Use with the application tag (to supply a default process for all application components), or with the activity, receiver, service, or provider tag (to supply a specific icon for that component).

Application components are normally run in a single process that is created for the entire application. You can use this tag to modify where they run. If the process name begins with a ':' character, a new process private to that application will be created when needed to run that component (allowing you to spread your application across multiple processes). If the process name begins with a lower-case character, the component will be run in a global process of that name, provided that you have permission to do so, allowing multiple applications to share one process to reduce resource usage.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [process](#) ([/reference/android/R.attr.html#process](#)).

Constant Value: 6 (0x00000006)

#### **public static final int [AndroidManifestReceiver\\_singleUser](#)**

If set to true, a single instance of this component will run for all users. That instance will run as user 0, the default/primary user. When the app running is in processes for other users and interacts with this component (by binding to a service for example) those processes will always interact with the instance running for user 0. Enabling single user mode forces "exported" of the component to be false, to help avoid introducing multi-user security bugs. This feature is only available to applications built in to the system image; you must hold the permission INTERACT\_ACROSS\_USERS in order to use this feature. This flag can only be used with services, receivers, and providers; it can not be used with activities.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in

the form "?[*package*: ][*type*: ]*name*" containing a value of this type.

This corresponds to the global attribute resource symbol [singleUser](#)  
[\(/reference/android/R.attr.html#singleUser\)](#).

Constant Value: 9 (0x00000009)

#### **public static final int `AndroidManifestService_description`**

Descriptive text for the associated data.

Must be a reference to another resource, in the form "@[+][*package*: ]*type*:*name*" or to a theme attribute in the form "?[*package*: ][*type*: ]*name*".

This corresponds to the global attribute resource symbol [description](#)  
[\(/reference/android/R.attr.html#description\)](#).

Constant Value: 7 (0x00000007)

#### **public static final int `AndroidManifestService_enabled`**

Specify whether the service is enabled or not (that is, can be instantiated by the system). It can also be specified for an application as a whole, in which case a value of "false" will override any component specific values (a value of "true" will not override the component specific values).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*: ]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [enabled](#)  
[\(/reference/android/R.attr.html#enabled\)](#).

Constant Value: 4 (0x00000004)

#### **public static final int `AndroidManifestService_exported`**

Flag indicating whether the given application component is available to other applications. If false, it can only be accessed by applications with its same user id (which usually means only by code in its own package). If true, it can be invoked by external entities, though which ones can do so may be controlled through permissions. The default value is false for activity, receiver, and service components that do not specify any intent filters; it is true for activity, receiver, and service components that do have intent filters (implying they expect to be invoked by others who do not know their particular component name) and for all content providers.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*: ]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [exported](#)  
[\(/reference/android/R.attr.html#exported\)](#).

Constant Value: 5 (0x00000005)

#### **public static final int `AndroidManifestService_icon`**

A Drawable resource providing a graphical representation of its associated item. Use with the application tag (to supply a default icon for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific icon for that component). It may also be used with the intent-filter tag to supply an icon to show to the user when an activity is being selected based on a particular Intent.

The given icon will be used to display to the user a graphical representation of its associated component; for example, as the icon for main activity that is displayed in the launcher. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[+][*package*: ]*type*:*name*" or to a theme attribute in the form "?[*package*: ][*type*: ]*name*".

This corresponds to the global attribute resource symbol [icon](#) [\(/reference/android/R.attr.html#icon\)](#).

Constant Value: 1 (0x00000001)

#### **public static final int `AndroidManifestService_isolatedProcess`**

If set to true, this service will run under a special process that is isolated from the rest of the system. The only communication with it is through the Service API (binding and starting).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [isolatedProcess](#)  
(/reference/android/R.attr.html#isolatedProcess).

Constant Value: 10 (0x0000000a)

#### public static final int **AndroidManifestService\_label**

A user-legible name for the given item. Use with the application tag (to supply a default label for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific label for that component). It may also be used with the intent-filter tag to supply a label to show to the user when an activity is being selected based on a particular Intent.

The given label will be used wherever the user sees information about its associated component; for example, as the name of a main activity that is displayed in the launcher. You should generally set this to a reference to a string resource, so that it can be localized, however it is also allowed to supply a plain string for quick and dirty programming.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This corresponds to the global attribute resource symbol [label](#)  
(/reference/android/R.attr.html#label).

Constant Value: 0 (0x00000000)

#### public static final int **AndroidManifestService\_logo**

A Drawable resource providing an extended graphical logo for its associated item. Use with the application tag (to supply a default logo for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific logo for that component). It may also be used with the intent-filter tag to supply a logo to show to the user when an activity is being selected based on a particular Intent.

The given logo will be used to display to the user a graphical representation of its associated component; for example as the header in the Action Bar. The primary differences between an icon and a logo are that logos are often wider and more detailed, and are used without an accompanying text caption. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [logo](#) (/reference/android/R.attr.html#logo).

Constant Value: 8 (0x00000008)

#### public static final int **AndroidManifestService\_name**

Required name of the class implementing the service, deriving from [Service](#)  
(/reference/android/app/Service.html). This is a fully qualified class name (for example, com.mycompany.myapp.MyService); as a short-hand if the first character of the class is a period then it is appended to your package name.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) (/reference/android/R.attr.html#name).

Constant Value: 2 (0x00000002)

#### public static final int **AndroidManifestService\_permission**

Specify a permission that a client is required to have in order to use the associated object. If the client does not hold the named permission, its request will fail. See the [Security and Permissions](#)  
(/guide/topics/security/security.html) document for more information on permissions.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [permission](#)  
(/reference/android/R.attr.html#permission).

Constant Value: 3 (0x00000003)

#### **public static final int AndroidManifestService\_process**

Specify a specific process that the associated code is to run in. Use with the application tag (to supply a default process for all application components), or with the activity, receiver, service, or provider tag (to supply a specific icon for that component).

Application components are normally run in a single process that is created for the entire application. You can use this tag to modify where they run. If the process name begins with a ':' character, a new process private to that application will be created when needed to run that component (allowing you to spread your application across multiple processes). If the process name begins with a lower-case character, the component will be run in a global process of that name, provided that you have permission to do so, allowing multiple applications to share one process to reduce resource usage.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [process](#)  
(/reference/android/R.attr.html#process).

Constant Value: 6 (0x00000006)

#### **public static final int AndroidManifestService\_singleUser**

If set to true, a single instance of this component will run for all users. That instance will run as user 0, the default/primary user. When the app running is in processes for other users and interacts with this component (by binding to a service for example) those processes will always interact with the instance running for user 0. Enabling single user mode forces "exported" of the component to be false, to help avoid introducing multi-user security bugs. This feature is only available to applications built in to the system image; you must hold the permission INTERACT\_ACROSS\_USERS in order to use this feature. This flag can only be used with services, receivers, and providers; it can not be used with activities.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [singleUser](#)  
(/reference/android/R.attr.html#singleUser).

Constant Value: 11 (0x0000000b)

#### **public static final int AndroidManifestService\_stopWithTask**

If set to true, this service will be automatically stopped when the user remove a task rooted in an activity owned by the application. The default is false.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [stopWithTask](#)  
(/reference/android/R.attr.html#stopWithTask).

Constant Value: 9 (0x00000009)

#### **public static final int AndroidManifestSupportsScreens\_anyDensity**

Indicates whether the application can accommodate any screen density. Older applications are assumed to not be able to, new ones able to. You can explicitly supply your abilities here.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [anyDensity](#)  
(/reference/android/R.attr.html#anyDensity).

Constant Value: 0 (0x00000000)

## **public static final int `AndroidManifestSupportsScreens_compatibleWidthLimitDp`**

Starting with [HONEYCOMB\\_MR2](#) ([/reference/android/os/Build.VERSION\\_CODES.html#HONEYCOMB\\_MR2](#)), this is the new way to specify the largest screens an application is compatible with. This attribute provides the maximum "smallest screen width" (as per the -swNNNdp resource configuration) that the application is designed for. If this value is smaller than the "smallest screen width" of the device it is running on, the user is offered to run it in a compatibility mode that emulates a smaller screen and zooms it to fit the screen. Currently the compatibility mode only emulates phone screens with a 320dp width, so compatibility mode is not applied if the value for compatibleWidthLimitDp is larger than 320.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[*package:*] type:*name*") or theme attribute (in the form "?[*package:*] [type:*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [compatibleWidthLimitDp](#) ([/reference/android/R.attr.html#compatibleWidthLimitDp](#)).

Constant Value: 7 (0x00000007)

## **public static final int `AndroidManifestSupportsScreens_largeScreens`**

Indicates whether the application supports larger screen form-factors. A large screen is defined as a screen that is significantly larger than a normal phone screen, and thus may require some special care on the application's part to make good use of it. An example would be a VGA *normal density* screen, though even larger screens are certainly possible. An application that does not support large screens will be placed as a postage stamp on such a screen, so that it retains the dimensions it was originally designed for.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*] type:*name*") or theme attribute (in the form "?[*package:*] [type:*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [largeScreens](#) ([/reference/android/R.attr.html#largeScreens](#)).

Constant Value: 3 (0x00000003)

## **public static final int `AndroidManifestSupportsScreens_largestWidthLimitDp`**

Starting with [HONEYCOMB\\_MR2](#) ([/reference/android/os/Build.VERSION\\_CODES.html#HONEYCOMB\\_MR2](#)), this is the new way to specify the screens an application is compatible with. This attribute provides the maximum "smallest screen width" (as per the -swNNNdp resource configuration) that the application can work well on. If this value is smaller than the "smallest screen width" of the device it is running on, the application will be forced into screen compatibility mode with no way for the user to turn it off. Currently the compatibility mode only emulates phone screens with a 320dp width, so compatibility mode is not applied if the value for largestWidthLimitDp is larger than 320.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[*package:*] type:*name*") or theme attribute (in the form "?[*package:*] [type:*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [largestWidthLimitDp](#) ([/reference/android/R.attr.html#largestWidthLimitDp](#)).

Constant Value: 8 (0x00000008)

## **public static final int `AndroidManifestSupportsScreens_normalScreens`**

Indicates whether an application supports the normal screen form-factors. Traditionally this is an HVGA normal density screen, but WQVGA low density and WVGA high density are also considered to be normal. This attribute is true by default, and applications currently should leave it that way.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*] type:*name*") or theme attribute (in the form "?[*package:*] [type:*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [normalScreens](#) ([/reference/android/R.attr.html#normalScreens](#)).

Constant Value: 2 (0x00000002)

## **public static final int `AndroidManifestSupportsScreens_requiresSmallestWidthDp`**

Starting with [HONEYCOMB\\_MR2](#) ([/reference/android/os/Build.VERSION\\_CODES.html#HONEYCOMB\\_MR2](#)), this

is the new way to specify the minimum screen size an application is compatible with. This attribute provides the required minimum "smallest screen width" (as per the -swNNNdp resource configuration) that the application can run on. For example, a typical phone screen is 320, a 7" tablet 600, and a 10" tablet 720. If the smallest screen width of the device is below the value supplied here, then the application is considered incompatible with that device. If not supplied, then any old smallScreens, normalScreens, largeScreens, or xlargeScreens attributes will be used instead.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [requiresSmallestWidthDp](#)  
(/reference/android/R.attr.html#requiresSmallestWidthDp).

Constant Value: 6 (0x00000006)

#### **public static final int `AndroidManifestSupportsScreens_resizeable`**

Indicates whether the application can resize itself to newer screen sizes. This is mostly used to distinguish between old applications that may not be compatible with newly introduced screen sizes and newer applications that should be; it will be set for you automatically based on whether you are targeting a newer platform that supports more screens.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [resizeable](#)  
(/reference/android/R.attr.html#resizeable).

Constant Value: 4 (0x00000004)

#### **public static final int `AndroidManifestSupportsScreens_smallScreens`**

Indicates whether the application supports smaller screen form-factors. A small screen is defined as one with a smaller aspect ratio than the traditional HVGA screen; that is, for a portrait screen, less tall than an HVGA screen. In practice, this means a QVGA low density or VGA high density screen. An application that does not support small screens *will not be available* for small screen devices, since there is little the platform can do to make such an application work on a smaller screen.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [smallScreens](#)  
(/reference/android/R.attr.html#smallScreens).

Constant Value: 1 (0x00000001)

#### **public static final int `AndroidManifestSupportsScreens_xlargeScreens`**

Indicates whether the application supports extra large screen form-factors.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [xlargeScreens](#)  
(/reference/android/R.attr.html#xlargeScreens).

Constant Value: 5 (0x00000005)

#### **public static final int `AndroidManifestUsesConfiguration_reqFiveWayNav`**

Application's requirement for five way navigation

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [reqFiveWayNav](#)  
(/reference/android/R.attr.html#reqFiveWayNav).

Constant Value: 4 (0x00000004)

## `public static final int AndroidManifestUsesConfiguration_reqHardKeyboard`

Application's requirement for a hard keyboard

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ] [ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [reqHardKeyboard](#)  
[\(/reference/android/R.attr.html#reqHardKeyboard\)](#).

Constant Value: 2 (0x00000002)

## `public static final int AndroidManifestUsesConfiguration_reqKeyboardType`

The input method preferred by an application.

Must be one of the following constant values.

Constant	Value	Description
----------	-------	-------------

undefined 0

nokeys 1

qwerty 2

twelvekey 3

This corresponds to the global attribute resource symbol [reqKeyboardType](#)  
[\(/reference/android/R.attr.html#reqKeyboardType\)](#).

Constant Value: 1 (0x00000001)

## `public static final int AndroidManifestUsesConfiguration_reqNavigation`

The navigation device preferred by an application.

Must be one of the following constant values.

Constant	Value	Description
----------	-------	-------------

undefined 0

nonav 1

dpad 2

trackball 3

wheel 4

This corresponds to the global attribute resource symbol [reqNavigation](#)  
[\(/reference/android/R.attr.html#reqNavigation\)](#).

Constant Value: 3 (0x00000003)

## `public static final int AndroidManifestUsesConfiguration_reqTouchScreen`

The type of touch screen used by an application.

Must be one of the following constant values.

Constant	Value	Description
----------	-------	-------------

undefined 0

notouch 1

stylus 2

finger 3

This corresponds to the global attribute resource symbol [reqTouchScreen](#)  
[\(/reference/android/R.attr.html#reqTouchScreen\)](#).

Constant Value: 0 (0x00000000)

## `public static final int AndroidManifestUsesFeature_glEsVersion`

The GLES driver version number needed by an application. The higher 16 bits represent the major number and the lower 16 bits represent the minor number. For example for GL 1.2 referring to 0x00000102, the actual value should be set as 0x00010002.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ] [ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [glEsVersion](#)  
(/reference/android/R.attr.html#glEsVersion).

Constant Value: 1 (0x00000001)

#### public static final int **AndroidManifestUsesFeature\_name**

The name of the feature that is being used.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) (/reference/android/R.attr.html#name).

Constant Value: 0 (0x00000000)

#### public static final int **AndroidManifestUsesFeature\_required**

Specify whether this feature is required for the application. The default is true, meaning the application requires the feature, and does not want to be installed on devices that don't support it. If you set this to false, then this will not impose a restriction on where the application can be installed.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [required](#)  
(/reference/android/R.attr.html#required).

Constant Value: 2 (0x00000002)

#### public static final int **AndroidManifestUsesLibrary\_name**

Required name of the library you use.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) (/reference/android/R.attr.html#name).

Constant Value: 0 (0x00000000)

#### public static final int **AndroidManifestUsesLibrary\_required**

Specify whether this library is required for the application. The default is true, meaning the application requires the library, and does not want to be installed on devices that don't support it. If you set this to false, then this will allow the application to be installed even if the library doesn't exist, and you will need to check for its presence dynamically at runtime.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [required](#)  
(/reference/android/R.attr.html#required).

Constant Value: 1 (0x00000001)

#### public static final int **AndroidManifestUsesPermission\_name**

Required name of the permission you use, as published with the corresponding name attribute of a [<permission>](#) (/reference/android/R.styleable.html#AndroidManifestPermission) tag; often this is one of the [standard system permissions](#) (/reference/android/Manifest.permission.html).

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) (/reference/android/R.attr.html#name).

Constant Value: 0 (0x00000000)

## **public static final int `AndroidManifestUsesSdk_maxSdkVersion`**

This is the maximum SDK version number that an application works on. You can use this to ensure your application is filtered out of later versions of the platform when you know you have incompatibility with them.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[*package:*] *type:name*") or theme attribute (in the form "?@[*package:*] [*type:*] *name*") containing a value of this type.

This corresponds to the global attribute resource symbol [maxSdkVersion](#)  
[\(/reference/android/R.attr.html#maxSdkVersion\)](#).

Constant Value: 2 (0x00000002)

## **public static final int `AndroidManifestUsesSdk_minSdkVersion`**

This is the minimum SDK version number that the application requires. This number is an abstract integer, from the list in [Build.VERSION\\_CODES](#) ([/reference/android/os/Build.VERSION\\_CODES.html](#)). If not supplied, the application will work on any SDK. This may also be string (such as "Donut") if the application was built against a development branch, in which case it will only work against the development builds.

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

May be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[*package:*] *type:name*") or theme attribute (in the form "?@[*package:*] [*type:*] *name*") containing a value of this type.

This corresponds to the global attribute resource symbol [minSdkVersion](#)  
[\(/reference/android/R.attr.html#minSdkVersion\)](#).

Constant Value: 0 (0x00000000)

## **public static final int `AndroidManifestUsesSdk_targetSdkVersion`**

This is the SDK version number that the application is targeting. It is able to run on older versions (down to minSdkVersion), but was explicitly tested to work with the version specified here. Specifying this version allows the platform to disable compatibility code that are not required or enable newer features that are not available to older applications. This may also be a string (such as "Donut") if this is built against a development branch, in which case minSdkVersion is also forced to be that string.

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

May be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[*package:*] *type:name*") or theme attribute (in the form "?@[*package:*] [*type:*] *name*") containing a value of this type.

This corresponds to the global attribute resource symbol [targetSdkVersion](#)  
[\(/reference/android/R.attr.html#targetSdkVersion\)](#).

Constant Value: 1 (0x00000001)

## **public static final int `AndroidManifest_installLocation`**

The default install location defined by an application.

Must be one of the following constant values.

Constant	Value	Description
auto	0	Let the system decide ideal install location
internalOnly	1	Explicitly request to be installed on internal phone storage only.
preferExternal	2	Prefer to be installed on SD card. There is no guarantee that the system will honor this request. The application might end up being installed on internal storage if external media is unavailable or too full.

This corresponds to the global attribute resource symbol [installLocation](#)  
[\(/reference/android/R.attr.html#installLocation\)](#).

Constant Value: 4 (0x00000004)

## **public static final int `AndroidManifest_sharedUserId`**

Specify the name of a user ID that will be shared between multiple packages. By default, each package gets its own unique user-id. By setting this value on two or more packages, each of these packages will be given a single shared user ID, so they can for example run in the same process. Note that for them to

actually get the same user ID, they must also be signed with the same signature.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [sharedUserId](#)  
(/reference/android/R.attr.html#sharedUserId).

Constant Value: 0 (0x00000000)

#### public static final int **AndroidManifest\_sharedUserLabel**

Specify a label for the shared user UID of this package. This is only used if you have also used android:sharedUserId. This must be a reference to a string resource; it can not be an explicit string.

Must be a reference to another resource, in the form "@[ + ][ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [sharedUserLabel](#)  
(/reference/android/R.attr.html#sharedUserLabel).

Constant Value: 3 (0x00000003)

#### public static final int **AndroidManifest\_versionCode**

Internal version code. This is the number used to determine whether one version is more recent than another: it has no other meaning than that higher numbers are more recent. You could use this number to encode a "x.y" in the lower and upper 16 bits, make it a build number, simply increase it by one each time a new version is released, or define it however else you want, as long as each successive version has a higher number. This is not a version number generally shown to the user, that is usually supplied with [versionName](#) (/reference/android/R.attr.html#versionName).

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [versionCode](#)  
(/reference/android/R.attr.html#versionCode).

Constant Value: 1 (0x00000001)

#### public static final int **AndroidManifest\_versionName**

The text shown to the user to indicate the version they have. This is used for no other purpose than display to the user; the actual significant version number is given by [versionCode](#)  
(/reference/android/R.attr.html#versionCode).

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [versionName](#)  
(/reference/android/R.attr.html#versionName).

Constant Value: 2 (0x00000002)

#### public static final int **AnimatedRotateDrawable\_drawable**

Reference to a drawable resource to use for the frame. If not given, the drawable must be defined by the first child tag.

Must be a reference to another resource, in the form "@[ + ][ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [drawable](#)  
(/reference/android/R.attr.html#drawable).

Constant Value: 1 (0x00000001)

#### public static final int **AnimatedRotateDrawable\_pivotX**

This symbol is the offset where the [pivotX](#) (/reference/android/R.attr.html#pivotX) attribute's value can be found in the [AnimatedRotateDrawable](#)  
(/reference/android/R.styleable.html#AnimatedRotateDrawable) array.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:][type:]name") containing a value of this type.

Constant Value: 2 (0x00000002)

#### public static final int **AnimatedRotateDrawable\_pivotY**

This symbol is the offset where the `pivotY` ([/reference/android/R.attr.html#pivotY](#)) attribute's value can be found in the `AnimatedRotateDrawable` ([/reference/android/R.styleable.html#AnimatedRotateDrawable](#)) array.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:][type:]name") containing a value of this type.

Constant Value: 3 (0x00000003)

#### public static final int **AnimatedRotateDrawable\_visible**

Provides initial visibility state of the drawable; the default value is false. See `setVisible(boolean, boolean)` ([/reference/android/graphics/drawable/Drawable.html#setVisible\(boolean, boolean\)](#)).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol `visible` ([/reference/android/R.attr.html#visible](#)).

Constant Value: 0 (0x00000000)

#### public static final int **AnimationDrawableItem\_drawable**

Reference to a drawable resource to use for the frame. If not given, the drawable must be defined by the first child tag.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?@[package:][type:]name".

This corresponds to the global attribute resource symbol `drawable` ([/reference/android/R.attr.html#drawable](#)).

Constant Value: 1 (0x00000001)

#### public static final int **AnimationDrawableItem\_duration**

Amount of time (in milliseconds) to display this frame.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol `duration` ([/reference/android/R.attr.html#duration](#)).

Constant Value: 0 (0x00000000)

#### public static final int **AnimationDrawable\_oneshot**

If true, the animation will only run a single time and then stop. If false (the default), it will continually run, restarting at the first frame after the last has finished.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in

the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [oneshot](#)  
(/reference/android/R.attr.html#oneshot).

Constant Value: 2 (0x00000002)

#### public static final int **AnimationDrawable\_variablePadding**

If true, allows the drawable's padding to change based on the current state that is selected. If false, the padding will stay the same (based on the maximum padding of all the states). Enabling this feature requires that the owner of the drawable deal with performing layout when the state changes, which is often not supported.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [variablePadding](#)  
(/reference/android/R.attr.html#variablePadding).

Constant Value: 1 (0x00000001)

#### public static final int **AnimationDrawable\_visible**

Provides initial visibility state of the drawable; the default value is false. See [setVisible\(boolean, boolean\)](#) (/reference/android/graphics/drawable.Drawable.html#setVisible(boolean, boolean)).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [visible](#)  
(/reference/android/R.attr.html#visible).

Constant Value: 0 (0x00000000)

#### public static final int **AnimationSet\_duration**

Amount of time (in milliseconds) to display this frame.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [duration](#)  
(/reference/android/R.attr.html#duration).

Constant Value: 0 (0x00000000)

#### public static final int **AnimationSet\_fillAfter**

When set to true, the animation transformation is applied after the animation is over. The default value is false. If fillEnabled is not set to true and the animation is not set on a View, fillAfter is assumed to be true.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [fillAfter](#)  
(/reference/android/R.attr.html#fillAfter).

Constant Value: 3 (0x00000003)

#### public static final int **AnimationSet\_fillBefore**

When set to true or when fillEnabled is not set to true, the animation transformation is applied before the animation has started. The default value is true.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [fillBefore](#)

[\(/reference/android/R.attr.html#fillBefore\).](#)

Constant Value: 2 (0x00000002)

#### public static final int **AnimationSet\_repeatMode**

Defines the animation behavior when it reaches the end and the repeat count is greater than 0 or infinite.  
The default value is restart.

Must be one of the following constant values.

Constant Value	Description
restart 1	The animation starts again from the beginning.
reverse 2	The animation plays backward.

This corresponds to the global attribute resource symbol [repeatMode](#)  
[\(/reference/android/R.attr.html#repeatMode\).](#)

Constant Value: 5 (0x00000005)

#### public static final int **AnimationSet\_shareInterpolator**

This symbol is the offset where the [shareInterpolator](#)  
[\(/reference/android/R.attr.html#shareInterpolator\)](#) attribute's value can be found in the  
[AnimationSet](#) [\(/reference/android/R.styleable.html#AnimationSet\)](#) array.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in  
the form "?@[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 1 (0x00000001)

#### public static final int **AnimationSet\_startOffset**

Delay in milliseconds before the animation runs, once start time is reached.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in  
the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [startOffset](#)  
[\(/reference/android/R.attr.html#startOffset\).](#)

Constant Value: 4 (0x00000004)

#### public static final int **Animation\_background**

Special background behind animation. Only for use with window animations. Can only be a color, and  
only black. If 0, the default, there is no background.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute  
in the form "?@[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [background](#)  
[\(/reference/android/R.attr.html#background\).](#)

Constant Value: 0 (0x00000000)

#### public static final int **Animation\_detachWallpaper**

Special option for window animations: if this window is on top of a wallpaper, don't animate the  
wallpaper with it.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in  
the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [detachWallpaper](#)  
[\(/reference/android/R.attr.html#detachWallpaper\).](#)

Constant Value: 10 (0x0000000a)

#### public static final int **Animation\_duration**

Amount of time (in milliseconds) for the animation to run.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [duration](#)  
[\(/reference/android/R.attr.html#duration\)](#).

Constant Value: 2 (0x00000002)

#### **public static final int Animation\_fillAfter**

When set to true, the animation transformation is applied after the animation is over. The default value is false. If fillEnabled is not set to true and the animation is not set on a View, fillAfter is assumed to be true.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [fillAfter](#)  
[\(/reference/android/R.attr.html#fillAfter\)](#).

Constant Value: 4 (0x00000004)

#### **public static final int Animation\_fillBefore**

When set to true or when fillEnabled is not set to true, the animation transformation is applied before the animation has started. The default value is true.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [fillBefore](#)  
[\(/reference/android/R.attr.html#fillBefore\)](#).

Constant Value: 3 (0x00000003)

#### **public static final int Animation\_fillEnabled**

When set to true, the value of fillBefore is taken into account.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [fillEnabled](#)  
[\(/reference/android/R.attr.html#fillEnabled\)](#).

Constant Value: 9 (0x00000009)

#### **public static final int Animation\_interpolator**

Defines the interpolator used to smooth the animation movement in time.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [interpolator](#)  
[\(/reference/android/R.attr.html#interpolator\)](#).

Constant Value: 1 (0x00000001)

#### **public static final int Animation\_repeatCount**

Defines how many times the animation should repeat. The default value is 0.

May be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

May be one of the following constant values.

## Constant Value Description

infinite -1

This corresponds to the global attribute resource symbol `repeatCount` (`/reference/android/R.attr.html#repeatCount`).

Constant Value: 6 (0x00000006)

```
public static final int Animation_repeatMode
```

Defines the animation behavior when it reaches the end and the repeat count is greater than 0 or infinite. The default value is restart.

Must be one of the following constant values.

Constant Value	Description
----------------	-------------

**restart 1** The animation starts again from the beginning.

**reverse 2** The animation plays backward.

This corresponds to the global attribute resource symbol `repeatMode` ([/reference/android/R.attr.html#repeatMode](#)).

Constant Value: 7 (0x00000007)

```
public static final int Animation_startOffset
```

Delay in milliseconds before the animation runs, once start time is reached

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package:] type: name") or theme attribute (in the form "[ package:] [ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol `startOffset` (`</reference android:R.attr.html#startOffset>`).

Constant Value: 5 (0x00000005)

```
public static final int Animation_zAdjustment
```

Allows for an adjustment of the Z ordering of the content being animated for the duration of the animation. The default value is normal.

Must be one of the following constant values.

Constant	Value	Description
normal	0	The content being animated be kept in its current Z order.
top	1	The content being animated is forced on top of all other content for the duration of the animation.
bottom	-1	The content being animated is forced under all other content for the duration of the animation.

This corresponds to the global attribute resource symbol `zAdjustment` (`/reference/android/R.attr.html#zAdjustment`).

Constant Value: 8 (0x00000008)

```
public static final int AnimatorSet_ordering
```

Name of the property being animated.

Must be one of the following constant values.

Constant	Value	Description
together	0	child animations should be played together.
sequentially	1	child animations should be played sequentially, in the same order as the xml.

This corresponds to the `globalAttribute.resourceSymbol` value in:

## [\(/reference/android/R.attr.html\)](#)

**public static final int Animator\_duration**

Amount of time (in milliseconds) for the

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*:]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [duration](#)  
(/reference/android/R.attr.html#duration).

Constant Value: 1 (0x00000001)

#### public static final int **Animator\_interpolator**

Defines the interpolator used to smooth the animation movement in time.

Must be a reference to another resource, in the form "@[+][*package*: ]*type*:*name*" or to a theme attribute (in the form "?[*package*: ][*type*:]*name*".

This corresponds to the global attribute resource symbol [interpolator](#)  
(/reference/android/R.attr.html#interpolator).

Constant Value: 0 (0x00000000)

#### public static final int **Animator\_repeatCount**

Defines how many times the animation should repeat. The default value is 0.

May be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*:]*name*") containing a value of this type.

May be one of the following constant values.

##### Constant Value Description

infinite -1

This corresponds to the global attribute resource symbol [repeatCount](#)  
(/reference/android/R.attr.html#repeatCount).

Constant Value: 3 (0x00000003)

#### public static final int **Animator\_repeatMode**

Defines the animation behavior when it reaches the end and the repeat count is greater than 0 or infinite. The default value is restart.

Must be one of the following constant values.

##### Constant Value Description

restart 1 The animation starts again from the beginning.

reverse 2 The animation plays backward.

This corresponds to the global attribute resource symbol [repeatMode](#)  
(/reference/android/R.attr.html#repeatMode).

Constant Value: 4 (0x00000004)

#### public static final int **Animator\_startOffset**

Delay in milliseconds before the animation runs, once start time is reached.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*:]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [startOffset](#)  
(/reference/android/R.attr.html#startOffset).

Constant Value: 2 (0x00000002)

#### public static final int **Animator\_valueFrom**

Value the animation starts from.

May be an integer value, such as "100".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

May be a floating point value, such as "1.2".

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [valueFrom](#) [\(/reference/android/R.attr.html#valueFrom\)](#).

Constant Value: 5 (0x00000005)

#### public static final int Animator\_valueTo

Value the animation animates to.

May be an integer value, such as "100".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

May be a floating point value, such as "1.2".

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [valueTo](#) [\(/reference/android/R.attr.html#valueTo\)](#).

Constant Value: 6 (0x00000006)

#### public static final int AnimatorValueType

The type of valueFrom and valueTo.

Must be one of the following constant values.

Constant	Value	Description
floatType	0	valueFrom and valueTo are floats. This is the default value if valueType is unspecified. Note that if either valueFrom or valueTo represent colors (beginning with "#"), then this attribute is ignored and the color values are interpreted as integers.
intType	1	valueFrom and valueTo are integers.

This corresponds to the global attribute resource symbol [valueType](#) [\(/reference/android/R.attr.html#valueType\)](#).

Constant Value: 7 (0x00000007)

#### public static final int AnticipateInterpolator\_tension

This is the amount of tension.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [tension](#) [\(/reference/android/R.attr.html#tension\)](#).

Constant Value: 0 (0x00000000)

#### public static final int AnticipateOvershootInterpolator\_extraTension

This is the amount by which to multiply the tension.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [extraTension](#) [\(/reference/android/R.attr.html#extraTension\)](#).

Constant Value: 1 (0x00000001)

**public static final int AnticipateOvershootInterpolator\_tension**

This is the amount of tension.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [tension](#)  
[\(/reference/android/R.attr.html#tension\)](#).

Constant Value: 0 (0x00000000)

**public static final int AppWidgetProviderInfo\_autoAdvanceViewId**

The view id of the AppWidget subview which should be auto-advanced by the widget's host.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [autoAdvanceViewId](#)  
[\(/reference/android/R.attr.html#autoAdvanceViewId\)](#).

Constant Value: 6 (0x00000006)

**public static final int AppWidgetProviderInfo\_configure**

A class name in the AppWidget's package to be launched to configure. If not supplied, then no activity will be launched.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [configure](#)  
[\(/reference/android/R.attr.html#configure\)](#).

Constant Value: 4 (0x00000004)

**public static final int AppWidgetProviderInfo\_initialKeyguardLayout**

A resource id of a layout.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [initialKeyguardLayout](#)  
[\(/reference/android/R.attr.html#initialKeyguardLayout\)](#).

Constant Value: 10 (0x0000000a)

**public static final int AppWidgetProviderInfo\_initialLayout**

A resource id of a layout.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [initialLayout](#)  
[\(/reference/android/R.attr.html#initialLayout\)](#).

Constant Value: 3 (0x00000003)

**public static final int AppWidgetProviderInfo\_minHeight**

Minimum height of the AppWidget.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [minHeight](#)  
[\(/reference/android/R.attr.html#minHeight\)](#).

Constant Value: 1 (0x00000001)

#### public static final int AppWidgetProviderInfo\_minResizeHeight

Minimum height that the AppWidget can be resized to.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [minResizeHeight](#)  
(/reference/android/R.attr.html#minResizeHeight).

Constant Value: 9 (0x00000009)

#### public static final int AppWidgetProviderInfo\_minResizeWidth

Minimum width that the AppWidget can be resized to.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [minResizeWidth](#)  
(/reference/android/R.attr.html#minResizeWidth).

Constant Value: 8 (0x00000008)

#### public static final int AppWidgetProviderInfo minWidth

Minimum width of the AppWidget.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [minWidth](#)  
(/reference/android/R.attr.html#minWidth).

Constant Value: 0 (0x00000000)

#### public static final int AppWidgetProviderInfo\_previewImage

A preview of what the AppWidget will look like after it's configured. If not supplied, the AppWidget's icon will be used.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [previewImage](#)  
(/reference/android/R.attr.html#previewImage).

Constant Value: 5 (0x00000005)

#### public static final int AppWidgetProviderInfo\_resizeMode

Optional parameter which indicates if and how this widget can be resized. Supports combined values using | operator.

May be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

May be one or more (separated by '|') of the following constant values.

Constant	Value	Description
none	0x0	
horizontal	0x1	

`vertical 0x2`  
This corresponds to the global attribute resource symbol [resizeMode](#)  
[\(/reference/android/R.attr.html#resizeMode\)](#).

Constant Value: 7 (0x00000007)

#### **public static final int AppWidgetProviderInfo\_updatePeriodMillis**

Update period in milliseconds, or 0 if the AppWidget will update itself.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [updatePeriodMillis](#)  
[\(/reference/android/R.attr.html#updatePeriodMillis\)](#).

Constant Value: 2 (0x00000002)

#### **public static final int AppWidgetProviderInfo\_widgetCategory**

Optional parameter which indicates where this widget can be shown, ie. home screen, keyguard or both. resized. Supports combined values using | operator.

May be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

May be one or more (separated by '|') of the following constant values.

Constant	Value	Description
home_screen	0x1	
keyguard	0x2	This corresponds to the global attribute resource symbol <a href="#">widgetCategory</a> <a href="#">(/reference/android/R.attr.html#widgetCategory)</a> .

Constant Value: 11 (0x0000000b)

#### **public static final int AutoCompleteTextView\_completionHint**

Defines the hint displayed in the drop down menu.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [completionHint](#)  
[\(/reference/android/R.attr.html#completionHint\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int AutoCompleteTextView\_completionHintView**

Defines the hint view displayed in the drop down menu.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [completionHintView](#)  
[\(/reference/android/R.attr.html#completionHintView\)](#).

Constant Value: 1 (0x00000001)

#### **public static final int AutoCompleteTextView\_completionThreshold**

Defines the number of characters that the user must type before completion suggestions are displayed in a drop down menu.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [completionThreshold](#)

[\(/reference/android/R.attr.html#completionThreshold\).](#)

Constant Value: 2 (0x00000002)

#### public static final int AutoCompleteTextView\_dropDownAnchor

View to anchor the auto-complete dropdown to. If not specified, the text view itself is used.

Must be a reference to another resource, in the form "@[+][ package: ] type: name" or to a theme attribute in the form "?[ package: ][ type: ] name".

This corresponds to the global attribute resource symbol [dropDownAnchor](#)

[\(/reference/android/R.attr.html#dropDownAnchor\).](#)

Constant Value: 6 (0x00000006)

#### public static final int AutoCompleteTextView\_dropDownHeight

Specifies the basic height of the dropdown. Its value may be a dimension (such as "12dip") for a constant height, fill\_parent or match\_parent to fill the height of the screen, or wrap\_content to match the height of the content of the drop down.

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ][ type: ] name") containing a value of this type.

May be one of the following constant values.

Constant	Value	Description
fill_parent	-1	The dropdown should fit the height of the screen. This constant is deprecated starting from API Level 8 and is replaced by match_parent.
match_parent	-1	The dropdown should fit the height of the screen. Introduced in API Level 8.
wrap_content	-2	The dropdown should fit the height of the content.

This corresponds to the global attribute resource symbol [dropDownHeight](#)

[\(/reference/android/R.attr.html#dropDownHeight\).](#)

Constant Value: 7 (0x00000007)

#### public static final int AutoCompleteTextView\_dropDownHorizontalOffset

Amount of pixels by which the drop down should be offset horizontally.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ][ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [dropDownHorizontalOffset](#)

[\(/reference/android/R.attr.html#dropDownHorizontalOffset\).](#)

Constant Value: 8 (0x00000008)

#### public static final int AutoCompleteTextView\_dropDownSelector

Selector in a drop down list.

May be a reference to another resource, in the form "@[+][ package: ] type: name" or to a theme attribute in the form "?[ package: ][ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [dropDownSelector](#)

[\(/reference/android/R.attr.html#dropDownSelector\).](#)

Constant Value: 3 (0x00000003)

#### public static final int AutoCompleteTextView\_dropDownVerticalOffset

Amount of pixels by which the drop down should be offset vertically.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font

size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [dropDownVerticalOffset](#)  
(/reference/android/R.attr.html#dropDownVerticalOffset).

Constant Value: 9 (0x00000009)

#### public static final int AutoCompleteTextView.DropDownWidth

Specifies the basic width of the dropdown. Its value may be a dimension (such as "12dip") for a constant width, fill\_parent or match\_parent to match the width of the screen, or wrap\_content to match the width of the anchored view.

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

May be one of the following constant values.

Constant	Value	Description
fill_parent	-1	The dropdown should fill the width of the screen. This constant is deprecated starting from API Level 8 and is replaced by match_parent.
match_parent	-1	The dropdown should fit the width of the screen. Introduced in API Level 8.
wrap_content	-2	The dropdown should fit the width of its anchor.

This corresponds to the global attribute resource symbol [dropDownWidth](#)  
(/reference/android/R.attr.html#dropDownWidth).

Constant Value: 5 (0x00000005)

#### public static final int AutoCompleteTextView.InputType

The type of data being placed in a text field, used to help an input method decide how to let the user enter text. The constants here correspond to those defined by [InputType](#)  
(/reference/android/text/InputType.html). Generally you can select a single value, though some can be combined together as indicated. Setting this attribute to anything besides *none* also implies that the text is editable.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
none	0x00000000	There is no content type. The text is not editable.
text	0x00000001	Just plain old text. Corresponds to <a href="#">TYPE_CLASS_TEXT</a>   <a href="#">TYPE_TEXT_VARIATION_NORMAL</a> .
textCapCharacters	0x00001001	Can be combined with <i>text</i> and its variations to request capitalization of all characters. Corresponds to <a href="#">TYPE_TEXT_FLAG_CAP_CHARACTERS</a> .
textCapWords	0x00002001	Can be combined with <i>text</i> and its variations to request capitalization of the first character of every word. Corresponds to <a href="#">TYPE_TEXT_FLAG_CAP_WORDS</a> .
textCapSentences	0x00004001	Can be combined with <i>text</i> and its variations to request capitalization of the first character of every sentence. Corresponds to <a href="#">TYPE_TEXT_FLAG_CAP_SENTENCES</a> .
textAutoCorrect	0x00008001	Can be combined with <i>text</i> and its variations to request auto-correction of text being input. Corresponds to <a href="#">TYPE_TEXT_FLAG_AUTO_CORRECT</a> .
textAutoComplete	0x00010001	Can be combined with <i>text</i> and its variations to specify that this field will be doing its own auto-completion and talking with the input method appropriately. Corresponds to <a href="#">TYPE_TEXT_FLAG_AUTO_COMPLETE</a> .
textMultiLine	0x00020001	Can be combined with <i>text</i> and its variations to allow multiple lines of text in the field. If this flag is not set, the text field will be constrained to a single line. Corresponds to <a href="#">TYPE_TEXT_FLAG_MULTI_LINE</a> .
textImeMultiLine	0x00040001	Can be combined with <i>text</i> and its variations to indicate that though the regular text view should not be multiple lines, the IME should provide multiple lines if it can. Corresponds to <a href="#">TYPE_TEXT_FLAG_IME_MULTI_LINE</a> .

textNoSuggestions	0x00080001	Can be combined with <i>text</i> and its variations to indicate that the IME should not show any dictionary-based word suggestions. Corresponds to <a href="#">TYPE_TEXT_FLAG_NO_SUGGESTIONS</a> .
textUri	0x00000011	Text that will be used as a URI. Corresponds to <a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_URI</a> .
textEmailAddress	0x00000021	Text that will be used as an e-mail address. Corresponds to <a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_EMAIL_ADDRESS</a> .
textEmailSubject	0x00000031	Text that is being supplied as the subject of an e-mail. Corresponds to <a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_EMAIL_SUBJECT</a> .
textShortMessage	0x00000041	Text that is the content of a short message. Corresponds to <a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_SHORT_MESSAGE</a> .
textLongMessage	0x00000051	Text that is the content of a long message. Corresponds to <a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_LONG_MESSAGE</a> .
textPersonName	0x00000061	Text that is the name of a person. Corresponds to <a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_PERSON_NAME</a> .
textPostalAddress	0x00000071	Text that is being supplied as a postal mailing address. Corresponds to <a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_POSTAL_ADDRESS</a> .
textPassword	0x00000081	Text that is a password. Corresponds to <a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_PASSWORD</a> .
textVisiblePassword	0x00000091	Text that is a password that should be visible. Corresponds to <a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_VISIBLE_PASSWORD</a> .
textWebEditText	0x000000a1	Text that is being supplied as text in a web form. Corresponds to <a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_WEB_EDIT_TEXT</a> .
textFilter	0x000000b1	Text that is filtering some other data. Corresponds to <a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_FILTER</a> .
textPhonetic	0x000000c1	Text that is for phonetic pronunciation, such as a phonetic name field in a contact entry. Corresponds to <a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_PHONETIC</a> .
textWebEmailAddress	0x000000d1	Text that will be used as an e-mail address on a web form. Corresponds to <a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_WEB_EMAIL_ADDRESS</a> .
textWebPassword	0x000000e1	Text that will be used as a password on a web form. Corresponds to <a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_WEB_PASSWORD</a> .
number	0x00000002	A numeric only field. Corresponds to <a href="#">TYPE_CLASS_NUMBER TYPE_NUMBER_VARIATION_NORMAL</a> .
numberSigned	0x00001002	Can be combined with <i>number</i> and its other options to allow a signed number. Corresponds to <a href="#">TYPE_CLASS_NUMBER TYPE_NUMBER_FLAG_SIGNED</a> .
numberDecimal	0x00002002	Can be combined with <i>number</i> and its other options to allow a decimal (fractional) number. Corresponds to <a href="#">TYPE_CLASS_NUMBER TYPE_NUMBER_FLAG_DECIMAL</a> .
numberPassword	0x00000012	A numeric password field. Corresponds to <a href="#">TYPE_CLASS_NUMBER TYPE_NUMBER_VARIATION_PASSWORD</a> .
phone	0x00000003	For entering a phone number. Corresponds to <a href="#">TYPE_CLASS_PHONE</a> .
datetime	0x00000004	For entering a date and time. Corresponds to <a href="#">TYPE_CLASS_DATETIME TYPE_DATETIME_VARIATION_NORMAL</a> .
date	0x00000014	For entering a date. Corresponds to <a href="#">TYPE_CLASS_DATETIME TYPE_DATETIME_VARIATION_DATE</a> .
time	0x00000024	For entering a time. Corresponds to <a href="#">TYPE_CLASS_DATETIME TYPE_DATETIME_VARIATION_TIME</a> .

This corresponds to the global attribute resource symbol [inputType](#)  
[\(/reference/android/R.attr.html#inputType\)](#).

Constant Value: 4 (0x00000004)

```
public static final int BitmapDrawable_antialias
```

Enables or disables antialiasing.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [antialias](#)  
[\(/reference/android/R.attr.html#antialias\)](#).

Constant Value: 2 (0x00000002)

#### public static final int **BitmapDrawable\_dither**

Enables or disables dithering of the bitmap if the bitmap does not have the same pixel configuration as the screen (for instance: a ARGB 8888 bitmap with an RGB 565 screen).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol dither

[\(/reference/android/R.attr.html#dither\)](#).

Constant Value: 4 (0x00000004)

#### public static final int **BitmapDrawable\_filter**

Enables or disables bitmap filtering. Filtering is used when the bitmap is shrunk or stretched to smooth its appearance.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol filter

[\(/reference/android/R.attr.html#filter\)](#).

Constant Value: 3 (0x00000003)

#### public static final int **BitmapDrawable\_gravity**

Defines the gravity for the bitmap. The gravity indicates where to position the drawable in its container if the bitmap is smaller than the container.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
top	0x30	Push object to the top of its container, not changing its size.
bottom	0x50	Push object to the bottom of its container, not changing its size.
left	0x03	Push object to the left of its container, not changing its size.
right	0x05	Push object to the right of its container, not changing its size.
center_vertical	0x10	Place object in the vertical center of its container, not changing its size.
fill_vertical	0x70	Grow the vertical size of the object if needed so it completely fills its container.
center_horizontal	0x01	Place object in the horizontal center of its container, not changing its size.
fill_horizontal	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
center	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
fill	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
clip_vertical	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
clip_horizontal	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a right gravity will clip the left edge, and neither will clip both edges.
start	0x00800003	Push object to the beginning of its container, not changing its size.
end	0x00800005	Push object to the end of its container, not changing its size.

This corresponds to the global attribute resource symbol gravity

[\(/reference/android/R.attr.html#gravity\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **BitmapDrawable\_src**

Identifier of the bitmap file. This attribute is mandatory.

May be a reference to another resource, in the form "`@[+][ package: ]type: name`" or to a theme attribute in the form "`?[ package: ][ type: ]name`".

May be a color value, in the form of "#`rgb`", "#`argb`", "#`rrggbb`", or "#`aarrggbb`".

This corresponds to the global attribute resource symbol [src](#) ([/reference/android/R.attr.html#src](#)).

Constant Value: 1 (0x00000001)

#### **public static final int BitmapDrawable\_tileMode**

Defines the tile mode. When the tile mode is enabled, the bitmap is repeated. Gravity is ignored when the tile mode is enabled.

Must be one of the following constant values.

Constant Value	Description
disabled -1	Do not tile the bitmap. This is the default value.
clamp 0	Replicates the edge color.
repeat 1	Repeats the bitmap in both direction.
mirror 2	Repeats the shader's image horizontally and vertically, alternating mirror images so that adjacent images always seam.

This corresponds to the global attribute resource symbol [tileMode](#) ([/reference/android/R.attr.html#tileMode](#)).

Constant Value: 5 (0x00000005)

#### **public static final int CalendarView\_dateTextAppearance**

The text appearance for the calendar dates.

Must be a reference to another resource, in the form "`@[+][ package: ]type: name`" or to a theme attribute in the form "`?[ package: ][ type: ]name`".

This corresponds to the global attribute resource symbol [dateTextAppearance](#) ([/reference/android/R.attr.html#dateTextAppearance](#)).

Constant Value: 12 (0x0000000c)

#### **public static final int CalendarView\_firstDayOfWeek**

The first day of week according to [Calendar](#) ([/reference/java/util/Calendar.html](#)).

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "`@[ package: ]type: name`") or theme attribute (in the form "`?[ package: ][ type: ]name`") containing a value of this type.

This corresponds to the global attribute resource symbol [firstDayOfWeek](#) ([/reference/android/R.attr.html#firstDayOfWeek](#)).

Constant Value: 0 (0x00000000)

#### **public static final int CalendarView\_focusedMonthDateColor**

The color for the dates of the focused month.

May be a reference to another resource, in the form "`@[+][ package: ]type: name`" or to a theme attribute in the form "`?[ package: ][ type: ]name`".

May be a color value, in the form of "#`rgb`", "#`argb`", "#`rrggbb`", or "#`aarrggbb`".

This corresponds to the global attribute resource symbol [focusedMonthDateColor](#) ([/reference/android/R.attr.html#focusedMonthDateColor](#)).

Constant Value: 6 (0x00000006)

#### **public static final int CalendarView\_maxDate**

The minimal date shown by this calendar view in mm/dd/yyyy format.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxxx' for a unicode character.

This may also be a reference to a resource (in the form "`@[ package: ]type: name`") or theme attribute (in the form "`?[ package: ][ type: ]name`") containing a value of this type.

This corresponds to the global attribute resource symbol [maxDate](#)  
[\(/reference/android/R.attr.html#maxDate\)](#).

Constant Value: 3 (0x00000003)

#### public static final int **CalendarView\_minDate**

The minimal date shown by this calendar view in mm/dd/yyyy format.

Must be a string value, using '\\,' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [minDate](#)  
[\(/reference/android/R.attr.html#minDate\)](#).

Constant Value: 2 (0x00000002)

#### public static final int **CalendarView\_selectedDateVerticalBar**

Drawable for the vertical bar shown at the beginning and at the end of the selected date.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [selectedDateVerticalBar](#)  
[\(/reference/android/R.attr.html#selectedDateVerticalBar\)](#).

Constant Value: 10 (0x0000000a)

#### public static final int **CalendarView\_selectedWeekBackgroundColor**

The background color for the selected week.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [selectedWeekBackgroundColor](#)  
[\(/reference/android/R.attr.html#selectedWeekBackgroundColor\)](#).

Constant Value: 5 (0x00000005)

#### public static final int **CalendarView\_showWeekNumber**

Whether do show week numbers.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [showWeekNumber](#)  
[\(/reference/android/R.attr.html#showWeekNumber\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **CalendarView\_shownWeekCount**

The number of weeks to be shown.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [shownWeekCount](#)  
[\(/reference/android/R.attr.html#shownWeekCount\)](#).

Constant Value: 4 (0x00000004)

#### public static final int **CalendarView\_unfocusedMonthTextColor**

The color for the dates of an unfocused month.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute

in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [unfocusedMonthDateColor](#)  
(/reference/android/R.attr.html#unfocusedMonthDateColor).

Constant Value: 7 (0x00000007)

#### public static final int **CalendarView\_weekDayTextAppearance**

The text appearance for the week day abbreviation of the calendar header.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [weekDayTextAppearance](#)  
(/reference/android/R.attr.html#weekDayTextAppearance).

Constant Value: 11 (0x0000000b)

#### public static final int **CalendarView\_weekNumberColor**

The color for the week numbers.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [weekNumberColor](#)  
(/reference/android/R.attr.html#weekNumberColor).

Constant Value: 8 (0x00000008)

#### public static final int **CalendarView\_weekSeparatorLineColor**

The color for the separator line between weeks.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [weekSeparatorLineColor](#)  
(/reference/android/R.attr.html#weekSeparatorLineColor).

Constant Value: 9 (0x00000009)

#### public static final int **CheckBoxPreference\_disableDependentsState**

The state (true for on, or false for off) that causes dependents to be disabled. By default, dependents will be disabled when this is unchecked, so the value of this preference is false.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [disableDependentsState](#)  
(/reference/android/R.attr.html#disableDependentsState).

Constant Value: 2 (0x00000002)

#### public static final int **CheckBoxPreference\_summaryOff**

The summary for the Preference in a PreferenceActivity screen when the CheckBoxPreference is unchecked. If separate on/off summaries are not needed, the summary attribute can be used instead.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [summaryOff](#)  
(/reference/android/R.attr.html#summaryOff).

Constant Value: 1 (0x00000001)

## **public static final int CheckBoxPreference\_summaryOn**

The summary for the Preference in a PreferenceActivity screen when the CheckBoxPreference is checked. If separate on/off summaries are not needed, the summary attribute can be used instead.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[+][package: ]type: name") or theme attribute (in the form "?[package: ][type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [summaryOn](#)  
[\(/reference/android/R.attr.html#summaryOn\)](#).

Constant Value: 0 (0x00000000)

## **public static final int CheckedTextView\_checkMark**

Drawable used for the check mark graphic.

Must be a reference to another resource, in the form "@[+][package: ]type: name" or to a theme attribute in the form "?[package: ][type: ]name".

This corresponds to the global attribute resource symbol [checkMark](#)  
[\(/reference/android/R.attr.html#checkMark\)](#).

Constant Value: 1 (0x00000001)

## **public static final int CheckedTextView\_checked**

Indicates the initial checked state of this text.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[+][package: ]type: name") or theme attribute (in the form "?[package: ][type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [checked](#)  
[\(/reference/android/R.attr.html#checked\)](#).

Constant Value: 0 (0x00000000)

## **public static final int Chronometer\_format**

Format string: if specified, the Chronometer will display this string, with the first "%" replaced by the current timer value in "MM:SS" or "H:MM:SS" form. If no format string is specified, the Chronometer will simply display "MM:SS" or "H:MM:SS".

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[+][package: ]type: name") or theme attribute (in the form "?[package: ][type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [format](#)  
[\(/reference/android/R.attr.html#format\)](#).

Constant Value: 0 (0x00000000)

## **public static final int ClipDrawable\_clipOrientation**

The orientation for the clip.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
horizontal	1	Clip the drawable horizontally.
vertical	2	Clip the drawable vertically.

This corresponds to the global attribute resource symbol [clipOrientation](#)  
[\(/reference/android/R.attr.html#clipOrientation\)](#).

Constant Value: 2 (0x00000002)

## **public static final int ClipDrawable\_drawable**

Reference to a drawable resource to draw with the specified scale.

Must be a reference to another resource, in the form "@[+][package: ]type: name" or to a theme attribute in the form "?[package: ][type: ]name".

This corresponds to the global attribute resource symbol [drawable](#)  
[\(/reference/android/R.attr.html#drawable\)](#).

Constant Value: 1 (0x00000001)

#### public static final int ClipDrawable\_gravity

Specifies where to clip within the drawable. The default value is left.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
top	0x30	Push object to the top of its container, not changing its size.
bottom	0x50	Push object to the bottom of its container, not changing its size.
left	0x03	Push object to the left of its container, not changing its size.
right	0x05	Push object to the right of its container, not changing its size.
center_vertical	0x10	Place object in the vertical center of its container, not changing its size.
fill_vertical	0x70	Grow the vertical size of the object if needed so it completely fills its container.
center_horizontal	0x01	Place object in the horizontal center of its container, not changing its size.
fill_horizontal	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
center	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
fill	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
clip_vertical	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
clip_horizontal	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a right gravity will clip the left edge, and neither will clip both edges.
start	0x00800003	Push object to the beginning of its container, not changing its size.
end	0x00800005	Push object to the end of its container, not changing its size.

This corresponds to the global attribute resource symbol [gravity](#)  
[\(/reference/android/R.attr.html#gravity\)](#).

Constant Value: 0 (0x00000000)

#### public static final int ColorDrawable\_color

The color to use.

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [color](#)  
[\(/reference/android/R.attr.html#color\)](#).

Constant Value: 0 (0x00000000)

#### public static final int CompoundButton\_button

Drawable used for the button graphic (e.g. checkbox, radio button, etc).

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [button](#)  
[\(/reference/android/R.attr.html#button\)](#).

Constant Value: 1 (0x00000001)

#### public static final int CompoundButton\_checked

Indicates the initial checked state of this button.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [checked](#)  
[\(/reference/android/R.attr.html#checked\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int ContactsDataKind\_allContactsName**

Resource representing the term "All Contacts" (e.g. "All Friends" or "All connections"). Optional (Default is "All Contacts").

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [allContactsName](#)  
[\(/reference/android/R.attr.html#allContactsName\)](#).

Constant Value: 5 (0x00000005)

#### **public static final int ContactsDataKind\_detailColumn**

Column in data table that contains details for this data.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [detailColumn](#)  
[\(/reference/android/R.attr.html#detailColumn\)](#).

Constant Value: 3 (0x00000003)

#### **public static final int ContactsDataKind\_detailSocialSummary**

Flag indicating that detail should be built from SocialProvider.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [detailSocialSummary](#)  
[\(/reference/android/R.attr.html#detailSocialSummary\)](#).

Constant Value: 4 (0x00000004)

#### **public static final int ContactsDataKind\_icon**

Icon used to represent data of this kind.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [icon](#) [\(/reference/android/R.attr.html#icon\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int ContactsDataKind\_mimeType**

Mime-type handled by this mapping.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [mimeType](#)  
[\(/reference/android/R.attr.html#mimeType\)](#).

Constant Value: 1 (0x00000001)

#### **public static final int ContactsDataKind\_summaryColumn**

Column in data table that summarizes this data.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [summaryColumn](#)  
(/reference/android/R.attr.html#summaryColumn).

Constant Value: 2 (0x00000002)

#### public static final int **CycleInterpolator\_cycles**

This symbol is the offset where the [cycles](#) (/reference/android/R.attr.html#cycles) attribute's value can be found in the [CycleInterpolator](#) (/reference/android/R.styleable.html#CycleInterpolator) array.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 0 (0x00000000)

#### public static final int **DatePicker\_calendarViewShown**

Whether the calendar view is shown.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [calendarViewShown](#)  
(/reference/android/R.attr.html#calendarViewShown).

Constant Value: 5 (0x00000005)

#### public static final int **DatePicker\_endYear**

The last year (inclusive), for example "2010".

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [endYear](#)  
(/reference/android/R.attr.html#endYear).

Constant Value: 1 (0x00000001)

#### public static final int **DatePicker\_maxDate**

The maximal date shown by this calendar view in mm/dd/yyyy format.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [maxDate](#)  
(/reference/android/R.attr.html#maxDate).

Constant Value: 3 (0x00000003)

#### public static final int **DatePicker\_minDate**

The minimal date shown by this calendar view in mm/dd/yyyy format.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [minDate](#)  
(/reference/android/R.attr.html#minDate).

Constant Value: 2 (0x00000002)

#### **public static final int DatePicker\_spinnersShown**

Whether the spinners are shown.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [spinnersShown](#)  
[\(/reference/android/R.attr.html#spinnersShown\)](#).

Constant Value: 4 (0x00000004)

#### **public static final int DatePicker\_startYear**

The first year (inclusive), for example "1940".

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [startYear](#)  
[\(/reference/android/R.attr.html#startYear\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int DecelerateInterpolator\_factor**

This is the amount of acceleration to add when easing out.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [factor](#)  
[\(/reference/android/R.attr.html#factor\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int DeviceAdmin\_visible**

Control whether the admin is visible to the user, even when it is not enabled. This is true by default. You may want to make it false if your admin does not make sense to be turned on unless some explicit action happens in your app.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [visible](#)  
[\(/reference/android/R.attr.html#visible\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int DialogPreference\_dialogIcon**

The icon for the dialog.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [dialogIcon](#)  
[\(/reference/android/R.attr.html#dialogIcon\)](#).

Constant Value: 2 (0x00000002)

#### **public static final int DialogPreference\_dialogLayout**

A layout to be used as the content View for the dialog. By default, this shouldn't be needed. If a custom DialogPreference is required, this should be set. For example, the EditTextPreference uses a layout with an EditText as this attribute.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [dialogLayout](#)  
(/reference/android/R.attr.html#dialogLayout).

Constant Value: 5 (0x00000005)

#### **public static final int DialogPreference\_dialogMessage**

The message in the dialog. If a dialogLayout is provided and contains a TextView with ID android:id/message, this message will be placed in there.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*][*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [dialogMessage](#)  
(/reference/android/R.attr.html#dialogMessage).

Constant Value: 1 (0x00000001)

#### **public static final int DialogPreference\_dialogTitle**

The title in the dialog.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*][*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [dialogTitle](#)  
(/reference/android/R.attr.html#dialogTitle).

Constant Value: 0 (0x00000000)

#### **public static final int DialogPreference\_negativeButtonText**

The negative button text for the dialog. Set to @null to hide the negative button.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*][*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [negativeButtonText](#)  
(/reference/android/R.attr.html#negativeButtonText).

Constant Value: 4 (0x00000004)

#### **public static final int DialogPreference\_positiveButtonText**

The positive button text for the dialog. Set to @null to hide the positive button.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*][*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [positiveButtonText](#)  
(/reference/android/R.attr.html#positiveButtonText).

Constant Value: 3 (0x00000003)

#### **public static final int DrawableCorners\_bottomLeftRadius**

Radius of the bottom left corner.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*][*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [bottomLeftRadius](#)  
(/reference/android/R.attr.html#bottomLeftRadius).

Constant Value: 3 (0x00000003)

#### public static final int **DrawableCorners\_bottomRightRadius**

Radius of the bottom right corner.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [bottomRightRadius](#)  
[\(/reference/android/R.attr.html#bottomRightRadius\)](#).

Constant Value: 4 (0x00000004)

#### public static final int **DrawableCorners\_radius**

Defines the radius of the four corners.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [radius](#)  
[\(/reference/android/R.attr.html#radius\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **DrawableCorners\_topLeftRadius**

Radius of the top left corner.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [topLeftRadius](#)  
[\(/reference/android/R.attr.html#topLeftRadius\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **DrawableCorners\_topRightRadius**

Radius of the top right corner.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [topRightRadius](#)  
[\(/reference/android/R.attr.html#topRightRadius\)](#).

Constant Value: 2 (0x00000002)

#### public static final int **DrawableStates\_state\_accelerated**

State value for [StateListDrawable](#)

[\(/reference/android/graphics/drawable/StateListDrawable.html\)](#), indicating that the Drawable is in a view that is hardware accelerated. This means that the device can at least render a full-screen scaled bitmap with one layer of text and bitmaps composited on top of it at 60fps. When this is set, the colorBackgroundCacheHint will be ignored even if it specifies a solid color, since that optimization is not needed.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in

the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_accelerated](#) ([/reference/android/R.attr.html#state\\_accelerated](#)).

Constant Value: 13 (0x0000000d)

#### public static final int **DrawableStates\_state\_activated**

State value for [StateListDrawable](#)

([/reference/android/graphics/drawable/StateListDrawable.html](#)), set when a view or its parent has been "activated" meaning the user has currently marked it as being of interest. This is an alternative representation of state\_checked for when the state should be propagated down the view hierarchy.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_activated](#) ([/reference/android/R.attr.html#state\\_activated](#)).

Constant Value: 12 (0x0000000c)

#### public static final int **DrawableStates\_state\_active**

State value for [StateListDrawable](#)

([/reference/android/graphics/drawable/StateListDrawable.html](#)).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_active](#) ([/reference/android/R.attr.html#state\\_active](#)).

Constant Value: 6 (0x00000006)

#### public static final int **DrawableStates\_state\_checkable**

State identifier indicating that the object *may* display a check mark. See [state\\_checked](#) ([/reference/android/R.attr.html#state\\_checked](#)) for the identifier that indicates whether it is actually checked.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_checkable](#) ([/reference/android/R.attr.html#state\\_checkable](#)).

Constant Value: 3 (0x00000003)

#### public static final int **DrawableStates\_state\_checked**

State identifier indicating that the object is currently checked. See [state\\_checkable](#) ([/reference/android/R.attr.html#state\\_checkable](#)) for an additional identifier that can indicate if any object may ever display a check, regardless of whether state\_checked is currently set.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_checked](#) ([/reference/android/R.attr.html#state\\_checked](#)).

Constant Value: 4 (0x00000004)

#### public static final int **DrawableStates\_state\_drag\_can\_accept**

State for [StateListDrawable](#) ([/reference/android/graphics/drawable/StateListDrawable.html](#)) indicating that the Drawable is in a view that is capable of accepting a drop of the content currently being manipulated in a drag-and-drop operation.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_drag\\_can\\_accept](#) ([/reference/android/R.attr.html#state\\_drag\\_can\\_accept](#)).

Constant Value: 15 (0x0000000f)

#### public static final int **DrawableStates\_state\_drag\_hovered**

State for [StateListDrawable](#) ([/reference/android/graphics/drawable/StateListDrawable.html](#)) indicating that a drag operation (for which the Drawable's view is a valid recipient) is currently positioned over the Drawable.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_drag\\_hovered](#) ([/reference/android/R.attr.html#state\\_drag\\_hovered](#)).

Constant Value: 16 (0x00000010)

#### public static final int **DrawableStates\_state\_enabled**

State value for [StateListDrawable](#) ([/reference/android/graphics/drawable/StateListDrawable.html](#)), set when a view is enabled.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_enabled](#) ([/reference/android/R.attr.html#state\\_enabled](#)).

Constant Value: 2 (0x00000002)

#### public static final int **DrawableStates\_state\_first**

State value for [StateListDrawable](#) ([/reference/android/graphics/drawable/StateListDrawable.html](#)).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_first](#) ([/reference/android/R.attr.html#state\\_first](#)).

Constant Value: 8 (0x00000008)

#### public static final int **DrawableStates\_state\_focused**

State value for [StateListDrawable](#) ([/reference/android/graphics/drawable/StateListDrawable.html](#)), set when a view has input focus.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_focused](#) ([/reference/android/R.attr.html#state\\_focused](#)).

Constant Value: 0 (0x00000000)

#### public static final int **DrawableStates\_state\_hovered**

State value for [StateListDrawable](#) ([/reference/android/graphics/drawable/StateListDrawable.html](#)), set when a pointer is hovering over the view.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in

the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_hovered](#) ([/reference/android/R.attr.html#state\\_hovered](#)).

Constant Value: 14 (0x0000000e)

#### public static final int **DrawableStates\_state\_last**

State value for [StateListDrawable](#) ([/reference/android/graphics/drawable/StateListDrawable.html](#)).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_last](#) ([/reference/android/R.attr.html#state\\_last](#)).

Constant Value: 10 (0x0000000a)

#### public static final int **DrawableStates\_state\_middle**

State value for [StateListDrawable](#) ([/reference/android/graphics/drawable/StateListDrawable.html](#)).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_middle](#) ([/reference/android/R.attr.html#state\\_middle](#)).

Constant Value: 9 (0x00000009)

#### public static final int **DrawableStates\_state\_pressed**

State value for [StateListDrawable](#) ([/reference/android/graphics/drawable/StateListDrawable.html](#)), set when the user is pressing down in a view.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_pressed](#) ([/reference/android/R.attr.html#state\\_pressed](#)).

Constant Value: 11 (0x0000000b)

#### public static final int **DrawableStates\_state\_selected**

State value for [StateListDrawable](#) ([/reference/android/graphics/drawable/StateListDrawable.html](#)), set when a view (or one of its parents) is currently selected.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_selected](#) ([/reference/android/R.attr.html#state\\_selected](#)).

Constant Value: 5 (0x00000005)

#### public static final int **DrawableStates\_state\_single**

State value for [StateListDrawable](#) ([/reference/android/graphics/drawable/StateListDrawable.html](#)).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_single](#)  
(/reference/android/R.attr.html#state\_single).

Constant Value: 7 (0x00000007)

#### public static final int **DrawableStates\_state\_window\_focused**

State value for [StateListDrawable](#)  
(/reference/android/graphics/drawable/StateListDrawable.html), set when a view's window has input focus.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_windowFocused](#)  
(/reference/android/R.attr.html#state\_window\_focused).

Constant Value: 1 (0x00000001)

#### public static final int **Drawable\_visible**

Provides initial visibility state of the drawable; the default value is false. See [setVisible\(boolean, boolean\)](#)  
(/reference/android/graphics/drawable/Drawable.html#setVisible(boolean, boolean)).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [visible](#)  
(/reference/android/R.attr.html#visible).

Constant Value: 0 (0x00000000)

#### public static final int **Dream\_settingsActivity**

Component name of an activity that allows the user to modify the settings for this dream.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [settingsActivity](#)  
(/reference/android/R.attr.html#settingsActivity).

Constant Value: 0 (0x00000000)

#### public static final int **ExpandableListChildIndicatorState\_state\_last**

State identifier indicating the child is the last child within its group.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [stateLast](#)  
(/reference/android/R.attr.html#stateLast).

Constant Value: 0 (0x00000000)

#### public static final int **ExpandableListGroupIndicatorState\_state\_empty**

State identifier indicating the group is empty (has no children).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [stateEmpty](#)  
(/reference/android/R.attr.html#stateEmpty).

Constant Value: 1 (0x00000001)

**public static final int ExpandableListGroupIndicatorState\_state\_expanded**

State identifier indicating the group is expanded.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_expanded](#)  
(/reference/android/R.attr.html#state\_expanded).

Constant Value: 0 (0x00000000)

**public static final int ExpandableListView\_childDivider**

Drawable or color that is used as a divider for children. (It will drawn below and above child items.) The height of this will be the same as the height of the normal list item divider.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [childDivider](#)  
(/reference/android/R.attr.html#childDivider).

Constant Value: 6 (0x00000006)

**public static final int ExpandableListView\_childIndicator**

Indicator shown beside the child View. This can be a stateful Drawable.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [childIndicator](#)  
(/reference/android/R.attr.html#childIndicator).

Constant Value: 1 (0x00000001)

**public static final int ExpandableListView\_childIndicatorLeft**

The left bound for a child's indicator.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [childIndicatorLeft](#)  
(/reference/android/R.attr.html#childIndicatorLeft).

Constant Value: 4 (0x00000004)

**public static final int ExpandableListView\_childIndicatorRight**

The right bound for a child's indicator.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [childIndicatorRight](#)  
(/reference/android/R.attr.html#childIndicatorRight).

Constant Value: 5 (0x00000005)

**public static final int ExpandableListView\_groupIndicator**

Indicator shown beside the group View. This can be a stateful Drawable.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [groupIndicator](#)  
(/reference/android/R.attr.html#groupIndicator).

Constant Value: 0 (0x00000000)

#### public static final int **ExpandableListView\_indicatorLeft**

The left bound for an item's indicator. To specify a left bound specific to children, use childIndicatorLeft.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [indicatorLeft](#)  
(/reference/android/R.attr.html#indicatorLeft).

Constant Value: 2 (0x00000002)

#### public static final int **ExpandableListView\_indicatorRight**

The right bound for an item's indicator. To specify a right bound specific to children, use childIndicatorRight.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [indicatorRight](#)  
(/reference/android/R.attr.html#indicatorRight).

Constant Value: 3 (0x00000003)

#### public static final int **Extra\_name**

Required name of the extra data.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) (/reference/android/R.attr.html#name).

Constant Value: 0 (0x00000000)

#### public static final int **Extra\_value**

Concrete value to put for this named extra data.

May be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

May be an integer value, such as "100".

May be a boolean value, either "true" or "false".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

May be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [value](#)  
(/reference/android/R.attr.html#value).

Constant Value: 1 (0x00000001)

#### public static final int **FragmentAnimation\_fragmentCloseEnterAnimation**

This symbol is the offset where the [fragmentCloseEnterAnimation](#)  
(/reference/android/R.attr.html#fragmentCloseEnterAnimation) attribute's value can be found in the  
[FragmentAnimation](#) (/reference/android/R.styleable.html#FragmentAnimation) array.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

Constant Value: 2 (0x00000002)

#### public static final int FragmentAnimation\_fragmentCloseExitAnimation

This symbol is the offset where the [fragmentCloseExitAnimation](#) ([/reference/android/R.attr.html#fragmentCloseExitAnimation](#)) attribute's value can be found in the [FragmentAnimation](#) ([/reference/android/R.styleable.html#FragmentAnimation](#)) array.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

Constant Value: 3 (0x00000003)

#### public static final int FragmentAnimation\_fragmentFadeEnterAnimation

This symbol is the offset where the [fragmentFadeEnterAnimation](#) ([/reference/android/R.attr.html#fragmentFadeEnterAnimation](#)) attribute's value can be found in the [FragmentAnimation](#) ([/reference/android/R.styleable.html#FragmentAnimation](#)) array.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

Constant Value: 4 (0x00000004)

#### public static final int FragmentAnimation\_fragmentFadeExitAnimation

This symbol is the offset where the [fragmentFadeExitAnimation](#) ([/reference/android/R.attr.html#fragmentFadeExitAnimation](#)) attribute's value can be found in the [FragmentAnimation](#) ([/reference/android/R.styleable.html#FragmentAnimation](#)) array.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

Constant Value: 5 (0x00000005)

#### public static final int FragmentAnimation\_fragmentOpenEnterAnimation

This symbol is the offset where the [fragmentOpenEnterAnimation](#) ([/reference/android/R.attr.html#fragmentOpenEnterAnimation](#)) attribute's value can be found in the [FragmentAnimation](#) ([/reference/android/R.styleable.html#FragmentAnimation](#)) array.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

Constant Value: 0 (0x00000000)

#### public static final int FragmentAnimation\_fragmentOpenExitAnimation

This symbol is the offset where the [fragmentOpenExitAnimation](#) ([/reference/android/R.attr.html#fragmentOpenExitAnimation](#)) attribute's value can be found in the [FragmentAnimation](#) ([/reference/android/R.styleable.html#FragmentAnimation](#)) array.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

Constant Value: 1 (0x00000001)

#### public static final int FragmentBreadCrumbs\_gravity

Specifies how an object should position its content, on both the X and Y axes, within its own bounds.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
top	0x30	Push object to the top of its container, not changing its size.
bottom	0x50	Push object to the bottom of its container, not changing its size.
left	0x03	Push object to the left of its container, not changing its size.
right	0x05	Push object to the right of its container, not changing its size.
center_vertical	0x10	Place object in the vertical center of its container, not changing its size.
fill_vertical	0x70	Grow the vertical size of the object if needed so it completely fills

		its container.
center_horizontal	0x01	Place object in the horizontal center of its container, not changing its size.
fill_horizontal	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
center	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
fill	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
clip_vertical	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
clip_horizontal	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a right gravity will clip the left edge, and neither will clip both edges.
start	0x00800003	Push object to the beginning of its container, not changing its size.
end	0x00800005	Push object to the end of its container, not changing its size.

This corresponds to the global attribute resource symbol [gravity](#) ([/reference/android/R.attr.html#gravity](#)).

Constant Value: 0 (0x00000000)

#### public static final int Fragment\_id

Supply an identifier name for the top-level view, to later retrieve it with [View.findViewById\(\)](#) ([/reference/android/view/View.html#findViewById\(int\)](#)) or [Activity.findViewById\(\)](#) ([/reference/android/app/Activity.html#findViewById\(int\)](#)). This must be a resource reference; typically you set this using the @+ syntax to create a new ID resources. For example:  
 android:id="@+id/my\_id" which allows you to later retrieve the view with  
 findViewById(R.id.my\_id).

Must be a reference to another resource, in the form "@[+] [package:] type: name" or to a theme attribute in the form "?[package:] [type:] name".

This corresponds to the global attribute resource symbol [id](#) ([/reference/android/R.attr.html#id](#)).

Constant Value: 1 (0x00000001)

#### public static final int Fragment\_name

Supply the name of the fragment class to instantiate.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:] type: name") or theme attribute (in the form "?[package:] [type:] name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) ([/reference/android/R.attr.html#name](#)).

Constant Value: 0 (0x00000000)

#### public static final int Fragment\_tag

Supply a tag for the top-level view containing a String, to be retrieved later with [View.getTag\(\)](#) ([/reference/android/view/View.html#getTag\(\)](#)) or searched for with [View.findViewWithTag\(\)](#) ([/reference/android/view/View.html#findViewWithTag\(java.lang.Object\)](#)). It is generally preferable to use IDs (through the android:id attribute) instead of tags because they are faster and allow for compile-time type checking.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:] type: name") or theme attribute (in the form "?[package:] [type:] name") containing a value of this type.

This corresponds to the global attribute resource symbol [tag](#) ([/reference/android/R.attr.html#tag](#)).

Constant Value: 2 (0x00000002)

#### public static final int FrameLayout\_Layout\_layout\_gravity

Standard gravity constant that a child supplies to its parent. Defines how the child view should be positioned, on both the X and Y axes, within its enclosing layout.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
top	0x30	Push object to the top of its container, not changing its size.
bottom	0x50	Push object to the bottom of its container, not changing its size.
left	0x03	Push object to the left of its container, not changing its size.
right	0x05	Push object to the right of its container, not changing its size.
center_vertical	0x10	Place object in the vertical center of its container, not changing its size.
fill_vertical	0x70	Grow the vertical size of the object if needed so it completely fills its container.
center_horizontal	0x01	Place object in the horizontal center of its container, not changing its size.
fill_horizontal	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
center	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
fill	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
clip_vertical	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
clip_horizontal	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a right gravity will clip the left edge, and neither will clip both edges.
start	0x00800003	Push object to the beginning of its container, not changing its size.
end	0x00800005	Push object to the end of its container, not changing its size.

This corresponds to the global attribute resource symbol [layout\\_gravity](#)  
[\(/reference/android/R.attr.html#layout\\_gravity\)](#).

Constant Value: 0 (0x00000000)

#### public static final int FrameLayout\_foreground

Defines the drawable to draw over the content. This can be used as an overlay. The foreground drawable participates in the padding of the content if the gravity is set to fill.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [foreground](#)  
[\(/reference/android/R.attr.html#foreground\)](#).

Constant Value: 0 (0x00000000)

#### public static final int FrameLayout\_foregroundGravity

Defines the gravity to apply to the foreground drawable. The gravity defaults to fill.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
top	0x30	Push object to the top of its container, not changing its size.
bottom	0x50	Push object to the bottom of its container, not changing its size.
left	0x03	Push object to the left of its container, not changing its size.
right	0x05	Push object to the right of its container, not changing its size.
center_vertical	0x10	Place object in the vertical center of its container, not changing its size.
fill_vertical	0x70	Grow the vertical size of the object if needed so it completely fills its container.
center_horizontal	0x01	Place object in the horizontal center of its container, not changing its size.
fill_horizontal	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
center	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
fill	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.

<code>clip_vertical</code>	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
<code>clip_horizontal</code>	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a right gravity will clip the left edge, and neither will clip both edges.

This corresponds to the global attribute resource symbol [foregroundGravity](#)  
[\(/reference/android/R.attr.html#foregroundGravity\)](#).

Constant Value: 2 (0x00000002)

#### `public static final int FrameLayout_measureAllChildren`

Determines whether to measure all children or just those in the VISIBLE or INVISIBLE state when measuring. Defaults to false.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ] [ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [measureAllChildren](#)  
[\(/reference/android/R.attr.html#measureAllChildren\)](#).

Constant Value: 1 (0x00000001)

#### `public static final int Gallery_animationDuration`

Sets how long a transition animation should run (in milliseconds) when layout has changed. Only relevant if animation is turned on.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ] [ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [animationDuration](#)  
[\(/reference/android/R.attr.html#animationDuration\)](#).

Constant Value: 1 (0x00000001)

#### `public static final int Gallery_gravity`

Specifies how an object should position its content, on both the X and Y axes, within its own bounds.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
<code>top</code>	0x30	Push object to the top of its container, not changing its size.
<code>bottom</code>	0x50	Push object to the bottom of its container, not changing its size.
<code>left</code>	0x03	Push object to the left of its container, not changing its size.
<code>right</code>	0x05	Push object to the right of its container, not changing its size.
<code>center_vertical</code>	0x10	Place object in the vertical center of its container, not changing its size.
<code>fill_vertical</code>	0x70	Grow the vertical size of the object if needed so it completely fills its container.
<code>center_horizontal</code>	0x01	Place object in the horizontal center of its container, not changing its size.
<code>fill_horizontal</code>	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
<code>center</code>	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
<code>fill</code>	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
<code>clip_vertical</code>	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
<code>clip_horizontal</code>	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a

right gravity will clip the left edge, and neither will clip both edges.  
start            0x00800003 Push object to the beginning of its container, not changing its size.  
end            0x00800005 Push object to the end of its container, not changing its size.

This corresponds to the global attribute resource symbol [gravity](#)  
[\(/reference/android/R.attr.html#gravity\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int `Gallery_spacing`**

This symbol is the offset where the [spacing](#) ([\(/reference/android/R.attr.html#spacing\)](#)) attribute's value can be found in the [Gallery](#) ([\(/reference/android/R.styleable.html#Gallery\)](#)) array.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

Constant Value: 2 (0x00000002)

#### **public static final int `Gallery_unselectedAlpha`**

Sets the alpha on the items that are not selected.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [unselectedAlpha](#)  
[\(/reference/android/R.attr.html#unselectedAlpha\)](#).

Constant Value: 3 (0x00000003)

#### **public static final int `GestureOverlayView_eventsInterceptionEnabled`**

Defines whether the overlay should intercept the motion events when a gesture is recognized.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [eventsInterceptionEnabled](#)  
[\(/reference/android/R.attr.html#eventsInterceptionEnabled\)](#).

Constant Value: 10 (0x0000000a)

#### **public static final int `GestureOverlayView_fadeDuration`**

Duration, in milliseconds, of the fade out effect after the user is done drawing a gesture.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [fadeDuration](#)  
[\(/reference/android/R.attr.html#fadeDuration\)](#).

Constant Value: 5 (0x00000005)

#### **public static final int `GestureOverlayView_fadeEnabled`**

Defines whether the gesture will automatically fade out after being recognized.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [fadeEnabled](#)  
[\(/reference/android/R.attr.html#fadeEnabled\)](#).

Constant Value: 11 (0x0000000b)

**public static final int GestureOverlayView\_fadeOffset**

Time, in milliseconds, to wait before the gesture fades out after the user is done drawing it.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [fadeOffset](#)  
[\(/reference/android/R.attr.html#fadeOffset\)](#).

Constant Value: 4 (0x00000004)

**public static final int GestureOverlayView\_gestureColor**

Color used to draw a gesture.

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [gestureColor](#)  
[\(/reference/android/R.attr.html#gestureColor\)](#).

Constant Value: 2 (0x00000002)

**public static final int GestureOverlayView\_gestureStrokeAngleThreshold**

Minimum curve angle a stroke must contain before it is recognized as a gesture.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [gestureStrokeAngleThreshold](#)  
[\(/reference/android/R.attr.html#gestureStrokeAngleThreshold\)](#).

Constant Value: 9 (0x00000009)

**public static final int GestureOverlayView\_gestureStrokeLengthThreshold**

Minimum length of a stroke before it is recognized as a gesture.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [gestureStrokeLengthThreshold](#)  
[\(/reference/android/R.attr.html#gestureStrokeLengthThreshold\)](#).

Constant Value: 7 (0x00000007)

**public static final int GestureOverlayView\_gestureStrokeSquarenessThreshold**

Squareness threshold of a stroke before it is recognized as a gesture.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [gestureStrokeSquarenessThreshold](#)  
[\(/reference/android/R.attr.html#gestureStrokeSquarenessThreshold\)](#).

Constant Value: 8 (0x00000008)

**public static final int GestureOverlayView\_gestureStrokeType**

Defines the type of strokes that define a gesture.

Must be one of the following constant values.

Constant Value	Description
single 0	A gesture is made of only one stroke.

**multiple** 1 A gesture is made of multiple strokes.  
This corresponds to the global attribute resource symbol [gestureStrokeType](#)  
[\(/reference/android/R.attr.html#gestureStrokeType\)](#).

Constant Value: 6 (0x00000006)

#### **public static final int GestureOverlayView\_gestureStrokeWidth**

Width of the stroke used to draw the gesture.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [gestureStrokeWidth](#)  
[\(/reference/android/R.attr.html#gestureStrokeWidth\)](#).

Constant Value: 1 (0x00000001)

#### **public static final int GestureOverlayView\_orientation**

Indicates whether horizontal (when the orientation is vertical) or vertical (when orientation is horizontal) strokes automatically define a gesture.

Must be one of the following constant values.

Constant	Value	Description
horizontal	0	Defines an horizontal widget.
vertical	1	Defines a vertical widget.

This corresponds to the global attribute resource symbol [orientation](#)  
[\(/reference/android/R.attr.html#orientation\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int GestureOverlayView\_uncertainGestureColor**

Color used to draw the user's strokes until we are sure it's a gesture.

Must be a color value, in the form "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [uncertainGestureColor](#)  
[\(/reference/android/R.attr.html#uncertainGestureColor\)](#).

Constant Value: 3 (0x00000003)

#### **public static final int GlowPadView\_directionDescriptions**

Reference to an array resource that be used to announce the directions with targets around the circle.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [directionDescriptions](#)  
[\(/reference/android/R.attr.html#directionDescriptions\)](#).

Constant Value: 3 (0x00000003)

#### **public static final int GlowPadView\_gravity**

Specifies how an object should position its content, on both the X and Y axes, within its own bounds.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
top	0x30	Push object to the top of its container, not changing its size.
bottom	0x50	Push object to the bottom of its container, not changing its size.
left	0x03	Push object to the left of its container, not changing its size.
right	0x05	Push object to the right of its container, not changing its size.
center_vertical	0x10	Place object in the vertical center of its container, not changing its size.
fill_vertical	0x70	Grow the vertical size of the object if needed so it completely fills

		its container.
center_horizontal	0x01	Place object in the horizontal center of its container, not changing its size.
fill_horizontal	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
center	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
fill	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
clip_vertical	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
clip_horizontal	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a right gravity will clip the left edge, and neither will clip both edges.
start	0x00800003	Push object to the beginning of its container, not changing its size.
end	0x00800005	Push object to the end of its container, not changing its size.

This corresponds to the global attribute resource symbol [gravity](#)  
[\(/reference/android/R.attr.html#gravity\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **GlowPadView\_innerRadius**

Inner radius of glow area.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [innerRadius](#)  
[\(/reference/android/R.attr.html#innerRadius\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **GlowPadView\_targetDescriptions**

Reference to an array resource that be used as description for the targets around the circle.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [targetDescriptions](#)  
[\(/reference/android/R.attr.html#targetDescriptions\)](#).

Constant Value: 2 (0x00000002)

#### public static final int **GradientDrawableGradient\_angle**

Angle of the gradient.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [angle](#)  
[\(/reference/android/R.attr.html#angle\)](#).

Constant Value: 3 (0x00000003)

#### public static final int **GradientDrawableGradient\_centerColor**

Optional center color. For linear gradients, use centerX or centerY to place the center color.

Must be a color value, in the form "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [centerColor](#)  
(/reference/android/R.attr.html#centerColor).

Constant Value: 8 (0x00000008)

#### public static final int GradientDrawableGradient\_centerX

X coordinate of the origin of the gradient within the shape.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [centerX](#)  
(/reference/android/R.attr.html#centerX).

Constant Value: 5 (0x00000005)

#### public static final int GradientDrawableGradient\_centerY

Y coordinate of the origin of the gradient within the shape.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [centerY](#)  
(/reference/android/R.attr.html#centerY).

Constant Value: 6 (0x00000006)

#### public static final int GradientDrawableGradient\_endColor

End color of the gradient.

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [endColor](#)  
(/reference/android/R.attr.html#endColor).

Constant Value: 1 (0x00000001)

#### public static final int GradientDrawableGradient\_gradientRadius

Radius of the gradient, used only with radial gradient.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [gradientRadius](#)  
(/reference/android/R.attr.html#gradientRadius).

Constant Value: 7 (0x00000007)

#### public static final int GradientDrawableGradient\_startColor

Start color of the gradient.

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [startColor](#)  
(/reference/android/R.attr.html#startColor).

Constant Value: 0 (0x00000000)

#### public static final int GradientDrawableGradient\_type

Type of gradient. The default type is linear.

Must be one of the following constant values.

Constant Value	Description
linear 0	Linear gradient.
radial 1	Radial, or circular, gradient.
sweep 2	Sweep, or angled or diamond, gradient.

This corresponds to the global attribute resource symbol [type](#) (/reference/android/R.attr.html#type).

Constant Value: 4 (0x00000004)

#### public static final int GradientDrawableGradient\_useLevel

This symbol is the offset where the [useLevel](#) (/reference/android/R.attr.html#useLevel) attribute's value can be found in the [GradientDrawableGradient](#)  
(/reference/android/R.styleable.html#GradientDrawableGradient) array.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 2 (0x00000002)

#### public static final int GradientDrawablePadding\_bottom

Amount of bottom padding inside the gradient shape.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [bottom](#)  
(/reference/android/R.attr.html#bottom).

Constant Value: 3 (0x00000003)

#### public static final int GradientDrawablePadding\_left

Amount of left padding inside the gradient shape.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [left](#) (/reference/android/R.attr.html#left).

Constant Value: 0 (0x00000000)

#### public static final int GradientDrawablePadding\_right

Amount of right padding inside the gradient shape.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [right](#)  
[\(/reference/android/R.attr.html#right\)](#).

Constant Value: 2 (0x00000002)

#### public static final int **GradientDrawablePadding\_top**

Amount of top padding inside the gradient shape.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [top](#)  
[\(/reference/android/R.attr.html#top\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **GradientDrawableSize\_height**

Height of the gradient shape.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [height](#)  
[\(/reference/android/R.attr.html#height\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **GradientDrawableSize\_width**

Width of the gradient shape.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [width](#)  
[\(/reference/android/R.attr.html#width\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **GradientDrawableSolid\_color**

Solid color for the gradient shape.

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [color](#)  
[\(/reference/android/R.attr.html#color\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **GradientDrawableStroke\_color**

Color of the gradient shape's stroke.

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [color](#)  
[\(/reference/android/R.attr.html#color\)](#).

Constant Value: 1 (0x00000001)

## public static final int GradientDrawableStroke\_dashGap

Gap between dashes in the stroke.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [dashGap](#)  
[\(/reference/android/R.attr.html#dashGap\)](#).

Constant Value: 3 (0x00000003)

## public static final int GradientDrawableStroke\_dashWidth

Length of a dash in the stroke.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [dashWidth](#)  
[\(/reference/android/R.attr.html#dashWidth\)](#).

Constant Value: 2 (0x00000002)

## public static final int GradientDrawableStroke\_width

Width of the gradient shape's stroke.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [width](#)  
[\(/reference/android/R.attr.html#width\)](#).

Constant Value: 0 (0x00000000)

## public static final int GradientDrawable\_dither

Enables or disables dithering.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [dither](#)  
[\(/reference/android/R.attr.html#dither\)](#).

Constant Value: 0 (0x00000000)

## public static final int GradientDrawable\_innerRadius

Inner radius of the ring. When defined, innerRadiusRatio is ignored.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [innerRadius](#)  
[\(/reference/android/R.attr.html#innerRadius\)](#).

Constant Value: 6 (0x00000006)

## **public static final int GradientDrawable\_innerRadiusRatio**

Inner radius of the ring expressed as a ratio of the ring's width. For instance, if innerRadiusRatio=9, then the inner radius equals the ring's width divided by 9. This value is ignored if innerRadius is defined. Default value is 9.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [innerRadiusRatio](#)  
(/reference/android/R.attr.html#innerRadiusRatio).

Constant Value: 3 (0x00000003)

## **public static final int GradientDrawable\_shape**

Indicates what shape to fill with a gradient.

Must be one of the following constant values.

Constant	Value	Description
rectangle	0	Rectangle shape, with optional rounder corners.
oval	1	Oval shape.
line	2	Line shape.
ring	3	Ring shape.

This corresponds to the global attribute resource symbol [shape](#)  
(/reference/android/R.attr.html#shape).

Constant Value: 2 (0x00000002)

## **public static final int GradientDrawable\_thickness**

Thickness of the ring. When defined, thicknessRatio is ignored.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [thickness](#)  
(/reference/android/R.attr.html#thickness).

Constant Value: 7 (0x00000007)

## **public static final int GradientDrawable\_thicknessRatio**

Thickness of the ring expressed as a ratio of the ring's width. For instance, if thicknessRatio=3, then the thickness equals the ring's width divided by 3. This value is ignored if innerRadius is defined. Default value is 3.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [thicknessRatio](#)  
(/reference/android/R.attr.html#thicknessRatio).

Constant Value: 4 (0x00000004)

## **public static final int GradientDrawable\_useLevel**

Indicates whether the drawable's level affects the way the gradient is drawn.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [useLevel](#)  
(/reference/android/R.attr.html#useLevel).

Constant Value: 5 (0x00000005)

## public static final int GradientDrawable\_visible

Indicates whether the drawable should initially be visible.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [visible](#)  
(/reference/android/R.attr.html#visible).

Constant Value: 1 (0x00000001)

## public static final int GridLayoutAnimation\_columnDelay

Fraction of the animation duration used to delay the beginning of the animation of each column.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [columnDelay](#)  
(/reference/android/R.attr.html#columnDelay).

Constant Value: 0 (0x00000000)

## public static final int GridLayoutAnimation\_direction

Direction of the animation in the grid.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
left_to_right	0x0	Animates columns from left to right.
right_to_left	0x1	Animates columns from right to left.
top_to_bottom	0x0	Animates rows from top to bottom.
bottom_to_top	0x2	Animates rows from bottom to top.

This corresponds to the global attribute resource symbol [direction](#)  
(/reference/android/R.attr.html#direction).

Constant Value: 2 (0x00000002)

## public static final int GridLayoutAnimation\_directionPriority

Priority of the rows and columns. When the priority is none, both rows and columns have the same priority. When the priority is column, the animations will be applied on the columns first. The same goes for rows.

Must be one of the following constant values.

Constant	Value	Description
none	0	Rows and columns are animated at the same time.
column	1	Columns are animated first.
row	2	Rows are animated first.

This corresponds to the global attribute resource symbol [directionPriority](#)  
(/reference/android/R.attr.html#directionPriority).

Constant Value: 3 (0x00000003)

## public static final int GridLayoutAnimation\_rowDelay

Fraction of the animation duration used to delay the beginning of the animation of each row.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [rowDelay](#)  
(/reference/android/R.attr.html#rowDelay).

Constant Value: 1 (0x00000001)

#### public static final int GridLayout\_Layout\_layout\_column

The column boundary delimiting the left of the group of cells occupied by this view.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_column](#)  
(/reference/android/R.attr.html#layout\_column).

Constant Value: 1 (0x00000001)

#### public static final int GridLayout\_Layout\_layout\_columnSpan

The column span: the difference between the right and left boundaries delimiting the group of cells occupied by this view. The default is one. See [GridLayout.Spec](#)  
(/reference/android/widget/GridLayout.Spec.html).

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_columnSpan](#)  
(/reference/android/R.attr.html#layout\_columnSpan).

Constant Value: 4 (0x00000004)

#### public static final int GridLayout\_Layout\_layout\_gravity

Gravity specifies how a component should be placed in its group of cells. The default is LEFT | BASELINE. See [setGravity\(int\)](#) (/reference/android/widget/LayoutParams.html#setGravity(int)).

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
top	0x30	Push object to the top of its container, not changing its size.
bottom	0x50	Push object to the bottom of its container, not changing its size.
left	0x03	Push object to the left of its container, not changing its size.
right	0x05	Push object to the right of its container, not changing its size.
center_vertical	0x10	Place object in the vertical center of its container, not changing its size.
fill_vertical	0x70	Grow the vertical size of the object if needed so it completely fills its container.
center_horizontal	0x01	Place object in the horizontal center of its container, not changing its size.
fill_horizontal	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
center	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
fill	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
clip_vertical	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
clip_horizontal	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a right gravity will clip the left edge, and neither will clip both edges.
start	0x00800003	Push object to the beginning of its container, not changing its size.
end	0x00800005	Push object to the end of its container, not changing its size.

This corresponds to the global attribute resource symbol [layout\\_gravity](#)  
(/reference/android/R.attr.html#layout\_gravity).

Constant Value: 0 (0x00000000)

#### public static final int GridLayout\_Layout\_layout\_row

The row boundary delimiting the top of the group of cells occupied by this view.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_row](#)  
(/reference/android/R.attr.html#layout\_row).

Constant Value: 2 (0x00000002)

#### public static final int GridLayout\_Layout\_layout\_rowSpan

The row span: the difference between the bottom and top boundaries delimiting the group of cells occupied by this view. The default is one. See [GridLayout.Spec](#)  
(/reference/android/widget/GridLayout.Spec.html).

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_rowSpan](#)  
(/reference/android/R.attr.html#layout\_rowSpan).

Constant Value: 3 (0x00000003)

#### public static final int GridLayout\_alignmentMode

When set to alignMargins, causes alignment to take place between the outer boundary of a view, as defined by its margins. When set to alignBounds, causes alignment to take place between the edges of the view. The default is alignMargins. See [setAlignmentMode\(int\)](#)  
(/reference/android/widget/GridLayout.html#setAlignmentMode(int)).

Must be one of the following constant values.

Constant	Value	Description
alignBounds	0	Align the bounds of the children. See <a href="#">ALIGN_BOUNDS</a> .
alignMargins	1	Align the margins of the children. See <a href="#">ALIGN_MARGINS</a> .

This corresponds to the global attribute resource symbol [alignmentMode](#)  
(/reference/android/R.attr.html#alignmentMode).

Constant Value: 6 (0x00000006)

#### public static final int GridLayout\_columnCount

The maximum number of columns to create when automatically positioning children.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [columnCount](#)  
(/reference/android/R.attr.html#columnCount).

Constant Value: 3 (0x00000003)

#### public static final int GridLayout\_columnOrderPreserved

When set to true, forces column boundaries to appear in the same order as column indices. The default is true. See [setColumnOrderPreserved\(boolean\)](#)  
(/reference/android/widget/GridLayout.html#setColumnOrderPreserved(boolean)).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [columnOrderPreserved](#)  
(/reference/android/R.attr.html#columnOrderPreserved).

Constant Value: 4 (0x00000004)

#### public static final int **GridLayout\_orientation**

The orientation property is not used during layout. It is only used to allocate row and column parameters when they are not specified by its children's layout parameters. GridLayout works like LinearLayout in this case; putting all the components either in a single row or in a single column - depending on the value of this flag. In the horizontal case, a columnCount property may be additionally supplied to force new rows to be created when a row is full. The rowCount attribute may be used similarly in the vertical case. The default is horizontal.

Must be one of the following constant values.

Constant	Value	Description
horizontal	0	Defines an horizontal widget.
vertical	1	Defines a vertical widget.

This corresponds to the global attribute resource symbol [orientation](#)  
[\(/reference/android/R.attr.html#orientation\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **GridLayoutRowCount**

The maximum number of rows to create when automatically positioning children.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [rowCount](#)  
[\(/reference/android/R.attr.html#rowCount\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **GridLayout\_rowOrderPreserved**

When set to true, forces row boundaries to appear in the same order as row indices. The default is true. See [setRowOrderPreserved\(boolean\)](#)  
[\(/reference/android/widget/GridLayout.html#setRowOrderPreserved\(boolean\)\)](#).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [rowOrderPreserved](#)  
[\(/reference/android/R.attr.html#rowOrderPreserved\)](#).

Constant Value: 2 (0x00000002)

#### public static final int **GridLayout\_useDefaultMargins**

When set to true, tells GridLayout to use default margins when none are specified in a view's layout parameters. The default value is false. See [setUseDefaultMargins\(boolean\)](#)  
[\(/reference/android/widget/GridLayout.html#setUseDefaultMargins\(boolean\)\)](#).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [useDefaultMargins](#)  
[\(/reference/android/R.attr.html#useDefaultMargins\)](#).

Constant Value: 5 (0x00000005)

#### public static final int **GridView\_columnWidth**

Specifies the fixed width for each column.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [columnWidth](#)  
[\(/reference/android/R.attr.html#columnWidth\)](#).

Constant Value: 4 (0x00000004)

#### public static final int **GridView\_gravity**

Specifies the gravity within each cell.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
top	0x30	Push object to the top of its container, not changing its size.
bottom	0x50	Push object to the bottom of its container, not changing its size.
left	0x03	Push object to the left of its container, not changing its size.
right	0x05	Push object to the right of its container, not changing its size.
center_vertical	0x10	Place object in the vertical center of its container, not changing its size.
fill_vertical	0x70	Grow the vertical size of the object if needed so it completely fills its container.
center_horizontal	0x01	Place object in the horizontal center of its container, not changing its size.
fill_horizontal	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
center	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
fill	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
clip_vertical	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
clip_horizontal	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a right gravity will clip the left edge, and neither will clip both edges.
start	0x00800003	Push object to the beginning of its container, not changing its size.
end	0x00800005	Push object to the end of its container, not changing its size.

This corresponds to the global attribute resource symbol [gravity](#)  
[\(/reference/android/R.attr.html#gravity\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **GridView\_horizontalSpacing**

Defines the default horizontal spacing between columns.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [horizontalSpacing](#)  
[\(/reference/android/R.attr.html#horizontalSpacing\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **GridView\_numColumns**

Defines how many columns to show.

May be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

May be one of the following constant values.

Constant Value	Description
auto_fit -1	Display as many columns as possible to fill the available space.

This corresponds to the global attribute resource symbol [numColumns](#)  
(/reference/android/R.attr.html#numColumns).

Constant Value: 5 (0x00000005)

#### public static final int **GridView\_stretchMode**

Defines how columns should stretch to fill the available empty space, if any.

Must be one of the following constant values.

Constant	Value	Description
none	0	Stretching is disabled.
spacingWidth	1	The spacing between each column is stretched.
columnWidth	2	Each column is stretched equally.
spacingWidthUniform	3	The spacing between each column is uniformly stretched..

This corresponds to the global attribute resource symbol [stretchMode](#)  
(/reference/android/R.attr.html#stretchMode).

Constant Value: 3 (0x00000003)

#### public static final int **GridView\_verticalSpacing**

Defines the default vertical spacing between rows.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [verticalSpacing](#)  
(/reference/android/R.attr.html#verticalSpacing).

Constant Value: 2 (0x00000002)

#### public static final int **HorizontalScrollView\_fillViewport**

Defines whether the scrollview should stretch its content to fill the viewport.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [fillViewport](#)  
(/reference/android/R.attr.html#fillViewport).

Constant Value: 0 (0x00000000)

#### public static final int **IconDefault\_icon**

A Drawable resource providing a graphical representation of its associated item. Use with the application tag (to supply a default icon for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific icon for that component). It may also be used with the intent-filter tag to supply an icon to show to the user when an activity is being selected based on a particular Intent.

The given icon will be used to display to the user a graphical representation of its associated component; for example, as the icon for main activity that is displayed in the launcher. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [icon](#) (/reference/android/R.attr.html#icon).

Constant Value: 0 (0x00000000)

#### public static final int **IconMenuView\_maxItemsPerRow**

Defines the maximum number of items per row.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in

the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [maxItemsPerRow](#) ([/reference/android/R.attr.html#maxItemsPerRow](#)).

Constant Value: 2 (0x00000002)

#### public static final int **IconMenuView\_maxRows**

Defines the maximum number of rows displayed.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [maxRows](#) ([/reference/android/R.attr.html#maxRows](#)).

Constant Value: 1 (0x00000001)

#### public static final int **IconMenuView\_moreIcon**

'More' icon.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [moreIcon](#) ([/reference/android/R.attr.html#moreIcon](#)).

Constant Value: 3 (0x00000003)

#### public static final int **IconMenuView\_rowHeight**

Defines the height of each row.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [rowHeight](#) ([/reference/android/R.attr.html#rowHeight](#)).

Constant Value: 0 (0x00000000)

#### public static final int **Icon\_icon**

A Drawable resource providing a graphical representation of its associated item. Use with the application tag (to supply a default icon for all application components), or with the activity, receiver, service, or instrumentation tag (to supply a specific icon for that component). It may also be used with the intent-filter tag to supply an icon to show to the user when an activity is being selected based on a particular Intent.

The given icon will be used to display to the user a graphical representation of its associated component; for example, as the icon for main activity that is displayed in the launcher. This must be a reference to a Drawable resource containing the image definition.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [icon](#) ([/reference/android/R.attr.html#icon](#)).

Constant Value: 0 (0x00000000)

#### public static final int **Icon\_mimeType**

Specify a MIME type that is handled, as per [IntentFilter.addDataType\(\)](#) ([/reference/android/content/IntentFilter.html#addDataType\(java.lang.String\)](#)).

*Note: MIME type matching in the Android framework is case-sensitive, unlike formal RFC MIME types. As a result, MIME types here should always use lower case letters.*

Must be a string value, using '\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in

the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [mimeType](#) ([/reference/android/R.attr.html#mimeType](#)).

Constant Value: 1 (0x00000001)

#### **public static final int ImageView\_adjustViewBounds**

Set this to true if you want the ImageView to adjust its bounds to preserve the aspect ratio of its drawable.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [adjustViewBounds](#) ([/reference/android/R.attr.html#adjustViewBounds](#)).

Constant Value: 2 (0x00000002)

#### **public static final int ImageView\_baseline**

The offset of the baseline within this view. See {see android.view.View#getBaseline} for details

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [baseline](#) ([/reference/android/R.attr.html#baseline](#)).

Constant Value: 8 (0x00000008)

#### **public static final int ImageView\_baselineAlignBottom**

If true, the image view will be baseline aligned with based on its bottom edge.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [baselineAlignBottom](#) ([/reference/android/R.attr.html#baselineAlignBottom](#)).

Constant Value: 6 (0x00000006)

#### **public static final int ImageView\_cropToPadding**

If true, the image will be cropped to fit within its padding.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [cropToPadding](#) ([/reference/android/R.attr.html#cropToPadding](#)).

Constant Value: 7 (0x00000007)

#### **public static final int ImageView\_maxHeight**

An optional argument to supply a maximum height for this view. See {see android.widget.ImageView#setMaxHeight} for details.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [maxHeight](#) ([/reference/android/R.attr.html#maxHeight](#)).

Constant Value: 4 (0x00000004)

#### public static final int **ImageView\_maxWidth**

An optional argument to supply a maximum width for this view. See {see android.widget.ImageView#setMaxWidth} for details.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [maxWidth](#) ([/reference/android/R.attr.html#maxWidth](#)).

Constant Value: 3 (0x00000003)

#### public static final int **ImageView\_scaleType**

Controls how the image should be resized or moved to match the size of this ImageView.

Must be one of the following constant values.

Constant	Value	Description
matrix	0	
fitXY	1	
fitStart	2	
fitCenter	3	
fitEnd	4	
center	5	
centerCrop	6	
centerInside	7	

This corresponds to the global attribute resource symbol [scaleType](#) ([/reference/android/R.attr.html#scaleType](#)).

Constant Value: 1 (0x00000001)

#### public static final int **ImageView\_src**

Sets a drawable as the content of this ImageView.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [src](#) ([/reference/android/R.attr.html#src](#)).

Constant Value: 0 (0x00000000)

#### public static final int **ImageView\_tint**

Set a tinting color for the image.

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [tint](#) ([/reference/android/R.attr.html#tint](#)).

Constant Value: 5 (0x00000005)

#### public static final int **InputMethodService\_imeExtractEnterAnimation**

Animation to use when showing the fullscreen extract UI after it had previously been hidden.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [imeExtractEnterAnimation](#) ([/reference/android/R.attr.html#imeExtractEnterAnimation](#)).

Constant Value: 1 (0x00000001)

## **public static final int InputMethodService\_imeExtractExitAnimation**

Animation to use when hiding the fullscreen extract UI after it had previously been shown.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [imeExtractExitAnimation](#) ([/reference/android/R.attr.html#imeExtractExitAnimation](#)).

Constant Value: 2 (0x00000002)

## **public static final int InputMethodService\_imeFullscreenBackground**

Background to use for entire input method when it is being shown in fullscreen mode with the extract view, to ensure that it completely covers the application. This allows, for example, the candidate view to be hidden while in fullscreen mode without having the application show through behind it.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form "#rgb", "#argb", "#rrggb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [imeFullscreenBackground](#) ([/reference/android/R.attr.html#imeFullscreenBackground](#)).

Constant Value: 0 (0x00000000)

## **public static final int InputMethod\_Subtype\_icon**

The icon of the subtype.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [icon](#) ([/reference/android/R.attr.html#icon](#)).

Constant Value: 1 (0x00000001)

## **public static final int InputMethod\_Subtype\_imeSubtypeExtraValue**

The extra value of the subtype. This string can be any string and will be passed to the IME when the framework calls the IME with the subtype.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [imeSubtypeExtraValue](#) ([/reference/android/R.attr.html#imeSubtypeExtraValue](#)).

Constant Value: 4 (0x00000004)

## **public static final int InputMethod\_Subtype\_imeSubtypeLocale**

The locale of the subtype. This string should be a locale (e.g. en\_US, fr\_FR...) and will be passed to the IME when the framework calls the IME with the subtype. This is also used by the framework to know the supported locales of the IME.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [imeSubtypeLocale](#) ([/reference/android/R.attr.html#imeSubtypeLocale](#)).

Constant Value: 2 (0x00000002)

## **public static final int InputMethod\_Subtype\_imeSubtypeMode**

The mode of the subtype. This string can be a mode (e.g. voice, keyboard...) and this string will be passed to the IME when the framework calls the IME with the subtype.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [imeSubtypeMode](#)  
(/reference/android/R.attr.html#imeSubtypeMode).

Constant Value: 3 (0x00000003)

#### public static final int InputMethod\_Subtype\_isAuxiliary

Set true if the subtype is auxiliary. An auxiliary subtype won't be shown in the input method selection list in the settings app. InputMethodManager#switchToLastInputMethod will ignore auxiliary subtypes when it chooses a target subtype.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [isAuxiliary](#)  
(/reference/android/R.attr.html#isAuxiliary).

Constant Value: 5 (0x00000005)

#### public static final int InputMethod\_Subtype\_label

The name of the subtype.

May be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This corresponds to the global attribute resource symbol [label](#)  
(/reference/android/R.attr.html#label).

Constant Value: 0 (0x00000000)

#### public static final int InputMethod\_Subtype\_overridesImplicitlyEnabledSubtype

Set true when this subtype should be selected by default if no other subtypes are selected explicitly. Note that a subtype with this parameter being true will not be shown in the subtypes list.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [overridesImplicitlyEnabledSubtype](#)  
(/reference/android/R.attr.html#overridesImplicitlyEnabledSubtype).

Constant Value: 6 (0x00000006)

#### public static final int InputMethod\_Subtype\_subtypeId

The unique id for the subtype. The input method framework keeps track of enabled subtypes by ID. When the IME package gets upgraded, enabled IDs will stay enabled even if other attributes are different. If the ID is unspecified (by calling the other constructor or 0. Arrays.hashCode(new Object[] {locale, mode, extraValue, isAuxiliary, overridesImplicitlyEnabledSubtype})) will be used instead.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [subtypeId](#)  
(/reference/android/R.attr.html#subtypeId).

Constant Value: 7 (0x00000007)

#### public static final int InputMethod\_isDefault

Set to true in all of the configurations for which this input method should be considered an option as the default.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [isDefault](#)  
(/reference/android/R.attr.html#isDefault).

Constant Value: 0 (0x00000000)

#### public static final int **InputMethod\_settingsActivity**

Component name of an activity that allows the user to modify the settings for this service.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [settingsActivity](#)  
([/reference/android/R.attr.html#settingsActivity](#)).

Constant Value: 1 (0x00000001)

#### public static final int **InsetDrawable\_drawable**

Reference to a drawable resource to use for the frame. If not given, the drawable must be defined by the first child tag.

Must be a reference to another resource, in the form "[+][ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [drawable](#)  
([/reference/android/R.attr.html#drawable](#)).

Constant Value: 1 (0x00000001)

#### public static final int **InsetDrawable\_insetBottom**

This symbol is the offset where the [insetBottom](#) ([/reference/android/R.attr.html#insetBottom](#)) attribute's value can be found in the [InsetDrawable](#) ([/reference/android/R.styleable.html#InsetDrawable](#)) array.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 5 (0x00000005)

#### public static final int **InsetDrawable\_insetLeft**

This symbol is the offset where the [insetLeft](#) ([/reference/android/R.attr.html#insetLeft](#)) attribute's value can be found in the [InsetDrawable](#) ([/reference/android/R.styleable.html#InsetDrawable](#)) array.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 2 (0x00000002)

#### public static final int **InsetDrawable\_insetRight**

This symbol is the offset where the [insetRight](#) ([/reference/android/R.attr.html#insetRight](#)) attribute's value can be found in the [InsetDrawable](#) ([/reference/android/R.styleable.html#InsetDrawable](#)) array.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 3 (0x00000003)

#### public static final int **InsetDrawable\_insetTop**

This symbol is the offset where the [insetTop](#) ([/reference/android/R.attr.html#insetTop](#)) attribute's

value can be found in the [InsetDrawable](#) ([/reference/android/R.styleable.html#InsetDrawable](#)) array.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

Constant Value: 4 (0x00000004)

#### public static final int **InsetDrawable\_visible**

Provides initial visibility state of the drawable; the default value is false. See [setVisible\(boolean, boolean\)](#) ([/reference/android/graphics/drawable/Drawable.html#setVisible\(boolean, boolean\)](#)).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [visible](#) ([/reference/android/R.attr.html#visible](#)).

Constant Value: 0 (0x00000000)

#### public static final int **IntentCategory\_name**

Required name of the category.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) ([/reference/android/R.attr.html#name](#)).

Constant Value: 0 (0x00000000)

#### public static final int **Intent\_action**

The action name to assign to the Intent, as per [Intent.setAction\(\)](#) ([/reference/android/content/Intent.html#setAction\(java.lang.String\)](#)).

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [action](#) ([/reference/android/R.attr.html#action](#)).

Constant Value: 2 (0x00000002)

#### public static final int **Intent\_data**

The data URI to assign to the Intent, as per [Intent.setData\(\)](#) ([/reference/android/content/Intent.html#setData\(android.net.Uri\)](#)).

*Note: scheme and host name matching in the Android framework is case-sensitive, unlike the formal RFC. As a result, URIs here should always be normalized to use lower case letters for these elements (as well as other proper Uri normalization).*

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [data](#) ([/reference/android/R.attr.html#data](#)).

Constant Value: 3 (0x00000003)

#### public static final int **Intent\_mimeType**

The MIME type name to assign to the Intent, as per [Intent.setType\(\)](#) ([/reference/android/content/Intent.html#setType\(java.lang.String\)](#)).

*Note: MIME type matching in the Android framework is case-sensitive, unlike formal RFC MIME types. As a result,*

*MIME types here should always use lower case letters.*

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [mime\\_type](#)  
[\(/reference/android/R.attr.html#mime\\_type\)](#).

Constant Value: 1 (0x00000001)

#### **public static final int Intent\_targetClass**

The class part of the ComponentName to assign to the Intent, as per [Intent.setComponent\(\)](#)  
[\(/reference/android/content/Intent.html#setComponent\(android.content.ComponentName\)\)](#).

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [targetClass](#)  
[\(/reference/android/R.attr.html#targetClass\)](#).

Constant Value: 4 (0x00000004)

#### **public static final int Intent\_targetPackage**

The package part of the ComponentName to assign to the Intent, as per [Intent.setComponent\(\)](#)  
[\(/reference/android/content/Intent.html#setComponent\(android.content.ComponentName\)\)](#).

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [targetPackage](#)  
[\(/reference/android/R.attr.html#targetPackage\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int KeyboardLayout\_keyboardLayout**

The key character map file resource.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [keyboardLayout](#)  
[\(/reference/android/R.attr.html#keyboardLayout\)](#).

Constant Value: 2 (0x00000002)

#### **public static final int KeyboardLayout\_label**

The display label of the keyboard layout.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This corresponds to the global attribute resource symbol [label](#)  
[\(/reference/android/R.attr.html#label\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int KeyboardLayout\_name**

The name of the keyboard layout, must be unique in the receiver.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [name](#) [\(/reference/android/R.attr.html#name\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **KeyboardViewPreviewState\_state\_long\_pressable**

State for [KeyboardView](#) ([/reference/android/inputmethodservice/KeyboardView.html](#)) key preview background.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_long\\_pressable](#) ([/reference/android/R.attr.html#state\\_long\\_pressable](#)).

Constant Value: 0 (0x00000000)

#### public static final int **KeyboardView\_keyBackground**

Image for the key. This image needs to be a StateListDrawable, with the following possible states: normal, pressed, checkable, checkable+pressed, checkable+checked, checkable+checked+pressed.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [keyBackground](#) ([/reference/android/R.attr.html#keyBackground](#)).

Constant Value: 2 (0x00000002)

#### public static final int **KeyboardView\_keyPreviewHeight**

Height of the key press feedback popup.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [keyPreviewHeight](#) ([/reference/android/R.attr.html#keyPreviewHeight](#)).

Constant Value: 8 (0x00000008)

#### public static final int **KeyboardView\_keyPreviewLayout**

Layout resource for key press feedback.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [keyPreviewLayout](#) ([/reference/android/R.attr.html#keyPreviewLayout](#)).

Constant Value: 6 (0x00000006)

#### public static final int **KeyboardView\_keyPreviewOffset**

Vertical offset of the key press feedback from the key.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [keyPreviewOffset](#) ([/reference/android/R.attr.html#keyPreviewOffset](#)).

Constant Value: 7 (0x00000007)

#### public static final int **KeyboardView\_keyTextColor**

Color to use for the label in a key.

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [keyTextColor](#)  
(/reference/android/R.attr.html#keyTextColor).

Constant Value: 5 (0x00000005)

#### public static final int **KeyboardView\_keyTextSize**

Size of the text for character keys.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [keyTextSize](#)  
(/reference/android/R.attr.html#keyTextSize).

Constant Value: 3 (0x00000003)

#### public static final int **KeyboardView\_labelTextSize**

Size of the text for custom keys with some text and no icon.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [labelTextSize](#)  
(/reference/android/R.attr.html#labelTextSize).

Constant Value: 4 (0x00000004)

#### public static final int **KeyboardView\_popupLayout**

Layout resource for popup keyboards.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [popupLayout](#)  
(/reference/android/R.attr.html#popupLayout).

Constant Value: 10 (0x0000000a)

#### public static final int **KeyboardView\_shadowColor**

Place a shadow of the specified color behind the text.

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [shadowColor](#)  
(/reference/android/R.attr.html#shadowColor).

Constant Value: 0 (0x00000000)

#### public static final int **KeyboardView\_shadowRadius**

Radius of the shadow.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [shadowRadius](#)  
(/reference/android/R.attr.html#shadowRadius).

Constant Value: 1 (0x00000001)

#### public static final int **KeyboardView\_verticalCorrection**

Amount to offset the touch Y coordinate by, for bias correction.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [verticalCorrection](#)  
[\(/reference/android/R.attr.html#verticalCorrection\)](#).

Constant Value: 9 (0x00000009)

#### public static final int **Keyboard\_Key\_codes**

The unicode value or comma-separated values that this key outputs.

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

May be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [codes](#)  
[\(/reference/android/R.attr.html#codes\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **Keyboard\_Key\_iconPreview**

The icon to show in the popup preview.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [iconPreview](#)  
[\(/reference/android/R.attr.html#iconPreview\)](#).

Constant Value: 7 (0x00000007)

#### public static final int **Keyboard\_Key\_isModifier**

Whether this is a modifier key such as Alt or Shift.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [isModifier](#)  
[\(/reference/android/R.attr.html#isModifier\)](#).

Constant Value: 4 (0x00000004)

#### public static final int **Keyboard\_Key\_isRepeatable**

Whether long-pressing on this key will make it repeat.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [isRepeatable](#)  
[\(/reference/android/R.attr.html#isRepeatable\)](#).

Constant Value: 6 (0x00000006)

#### public static final int **Keyboard\_Key\_isSticky**

Whether this is a toggle key.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*] *type:****name***") or theme attribute (in the form "?[*package:*] [*type:*] *name*") containing a value of this type.

This corresponds to the global attribute resource symbol [isSticky](#)  
[\(/reference/android/R.attr.html#isSticky\)](#).

Constant Value: 5 (0x00000005)

#### **public static final int Keyboard.Key\_keyEdgeFlags**

Key edge flags.

Must be one or more (separated by '|') of the following constant values.

Constant Value	Description
left 1	Key is anchored to the left of the keyboard.
right 2	Key is anchored to the right of the keyboard.

This corresponds to the global attribute resource symbol [keyEdgeFlags](#)  
[\(/reference/android/R.attr.html#keyEdgeFlags\)](#).

Constant Value: 3 (0x00000003)

#### **public static final int Keyboard.Key\_keyIcon**

The icon to display on the key instead of the label.

Must be a reference to another resource, in the form "@[+] [*package:*] *type:****name***" or to a theme attribute in the form "?[*package:*] [*type:*] *name*".

This corresponds to the global attribute resource symbol [keyIcon](#)  
[\(/reference/android/R.attr.html#keyIcon\)](#).

Constant Value: 10 (0x0000000a)

#### **public static final int Keyboard.Key\_keyLabel**

The label to display on the key.

Must be a string value, using '\\.' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[*package:*] *type:****name***") or theme attribute (in the form "?[*package:*] [*type:*] *name*") containing a value of this type.

This corresponds to the global attribute resource symbol [keyLabel](#)  
[\(/reference/android/R.attr.html#keyLabel\)](#).

Constant Value: 9 (0x00000009)

#### **public static final int Keyboard.Key\_keyOutputText**

The string of characters to output when this key is pressed.

Must be a string value, using '\\.' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[*package:*] *type:****name***") or theme attribute (in the form "?[*package:*] [*type:*] *name*") containing a value of this type.

This corresponds to the global attribute resource symbol [keyOutputText](#)  
[\(/reference/android/R.attr.html#keyOutputText\)](#).

Constant Value: 8 (0x00000008)

#### **public static final int Keyboard.Key\_keyboardMode**

Mode of the keyboard. If the mode doesn't match the requested keyboard mode, the key will be skipped.

Must be a reference to another resource, in the form "@[+] [*package:*] *type:****name***" or to a theme attribute in the form "?[*package:*] [*type:*] *name*".

This corresponds to the global attribute resource symbol [keyboardMode](#)  
[\(/reference/android/R.attr.html#keyboardMode\)](#).

Constant Value: 11 (0x0000000b)

#### **public static final int Keyboard.Key\_popupCharacters**

The characters to display in the popup keyboard.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [popupCharacters](#)  
[\(/reference/android/R.attr.html#popupCharacters\)](#).

Constant Value: 2 (0x00000002)

#### public static final int Keyboard\_Key\_popupKeyboard

The XML keyboard layout of any popup keyboard.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [popupKeyboard](#)  
[\(/reference/android/R.attr.html#popupKeyboard\)](#).

Constant Value: 1 (0x00000001)

#### public static final int Keyboard\_Row\_keyboardMode

Mode of the keyboard. If the mode doesn't match the requested keyboard mode, the row will be skipped.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [keyboardMode](#)  
[\(/reference/android/R.attr.html#keyboardMode\)](#).

Constant Value: 1 (0x00000001)

#### public static final int Keyboard\_Row\_rowEdgeFlags

Row edge flags.

Must be one or more (separated by '|') of the following constant values.

Constant Value	Description
top 4	Row is anchored to the top of the keyboard.
bottom 8	Row is anchored to the bottom of the keyboard.

This corresponds to the global attribute resource symbol [rowEdgeFlags](#)  
[\(/reference/android/R.attr.html#rowEdgeFlags\)](#).

Constant Value: 0 (0x00000000)

#### public static final int Keyboard\_horizontalGap

Default horizontal gap between keys.

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [horizontalGap](#)  
[\(/reference/android/R.attr.html#horizontalGap\)](#).

Constant Value: 2 (0x00000002)

#### public static final int Keyboard\_keyHeight

Default height of a key, in pixels or percentage of display width.

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%".

The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [keyHeight](#)  
(/reference/android/R.attr.html#keyHeight).

Constant Value: 1 (0x00000001)

#### public static final int **Keyboard\_keyWidth**

Default width of a key, in pixels or percentage of display width.

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%".  
The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [keyWidth](#)  
(/reference/android/R.attr.html#keyWidth).

Constant Value: 0 (0x00000000)

#### public static final int **Keyboard\_verticalGap**

Default vertical gap between rows of keys.

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%".  
The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [verticalGap](#)  
(/reference/android/R.attr.html#verticalGap).

Constant Value: 3 (0x00000003)

#### public static final int **LayerDrawableItem\_bottom**

Bottom coordinate of the layer.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [bottom](#)  
(/reference/android/R.attr.html#bottom).

Constant Value: 5 (0x00000005)

#### public static final int **LayerDrawableItem\_drawable**

Drawable used to render the layer.

Must be a reference to another resource, in the form "@+[ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [drawable](#)  
(/reference/android/R.attr.html#drawable).

Constant Value: 1 (0x00000001)

## `public static final int LayerDrawableItem_id`

Identifier of the layer. This can be used to retrieve the layer from a drawable container.

Must be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ] [ type: ] name`".

This corresponds to the global attribute resource symbol [`id` \(/reference/android/R.attr.html#id\)](#).

Constant Value: 0 (0x00000000)

## `public static final int LayerDrawableItem_left`

Left coordinate of the layer.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ] [ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [`left` \(/reference/android/R.attr.html#left\)](#).

Constant Value: 2 (0x00000002)

## `public static final int LayerDrawableItem_right`

Right coordinate of the layer.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ] [ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [`right` \(/reference/android/R.attr.html#right\)](#).

Constant Value: 4 (0x00000004)

## `public static final int LayerDrawableItem_top`

Top coordinate of the layer.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ] [ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [`top` \(/reference/android/R.attr.html#top\)](#).

Constant Value: 3 (0x00000003)

## `public static final int LayerDrawable_opacity`

Indicates the opacity of the layer. This can be useful to allow the system to enable drawing optimizations. The default value is translucent.

Must be one of the following constant values.

Constant	Value	Description
opaque	-1	Indicates that the layer is opaque and contains no transparent nor translucent pixels.
transparent	-2	The layer is completely transparent (no pixel will be drawn.)
translucent	-3	The layer has translucent pixels.

This corresponds to the global attribute resource symbol [`opacity` \(/reference/android/R.attr.html#opacity\)](#).

Constant Value: 0 (0x00000000)

## `public static final int LayoutAnimation_animation`

Animation to use on each child.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [animation](#)  
[\(/reference/android/R.attr.html#animation\)](#).

Constant Value: 2 (0x00000002)

#### public static final int **LayoutAnimation\_animationOrder**

The order in which the animations will be started.

Must be one of the following constant values.

Constant Value	Description
normal 0	Animations are started in the natural order.
reverse 1	Animations are started in the reverse order.
random 2	Animations are started randomly.

This corresponds to the global attribute resource symbol [animationOrder](#)  
[\(/reference/android/R.attr.html#animationOrder\)](#).

Constant Value: 3 (0x00000003)

#### public static final int **LayoutAnimation\_delay**

Fraction of the animation duration used to delay the beginning of the animation of each child.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*][*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [delay](#)  
[\(/reference/android/R.attr.html#delay\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **LayoutAnimation\_interpolator**

Interpolator used to interpolate the delay between the start of each animation.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [interpolator](#)  
[\(/reference/android/R.attr.html#interpolator\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **LevelListDrawableItem\_drawable**

Reference to a drawable resource to use for the frame. If not given, the drawable must be defined by the first child tag.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [drawable](#)  
[\(/reference/android/R.attr.html#drawable\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **LevelListDrawableItem\_maxLevel**

The maximum level allowed for this item.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*][*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [maxLevel](#)  
[\(/reference/android/R.attr.html#maxLevel\)](#).

Constant Value: 2 (0x00000002)

#### public static final int **LevelListDrawableItem\_minLevel**

The minimum level allowed for this item.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [minLevel](#) ([/reference/android/R.attr.html#minLevel](#)).

Constant Value: 1 (0x00000001)

#### public static final int **LinearLayout\_Layout\_layout\_gravity**

Standard gravity constant that a child supplies to its parent. Defines how the child view should be positioned, on both the X and Y axes, within its enclosing layout.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
top	0x30	Push object to the top of its container, not changing its size.
bottom	0x50	Push object to the bottom of its container, not changing its size.
left	0x03	Push object to the left of its container, not changing its size.
right	0x05	Push object to the right of its container, not changing its size.
center_vertical	0x10	Place object in the vertical center of its container, not changing its size.
fill_vertical	0x70	Grow the vertical size of the object if needed so it completely fills its container.
center_horizontal	0x01	Place object in the horizontal center of its container, not changing its size.
fill_horizontal	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
center	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
fill	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
clip_vertical	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
clip_horizontal	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a right gravity will clip the left edge, and neither will clip both edges.
start	0x00800003	Push object to the beginning of its container, not changing its size.
end	0x00800005	Push object to the end of its container, not changing its size.

This corresponds to the global attribute resource symbol [layout\\_gravity](#) ([/reference/android/R.attr.html#layout\\_gravity](#)).

Constant Value: 0 (0x00000000)

#### public static final int **LinearLayout\_Layout\_layout\_height**

Specifies the basic height of the view. This is a required attribute for any view inside of a containing layout manager. Its value may be a dimension (such as "12dip") for a constant height or one of the special constants.

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

May be one of the following constant values.

Constant	Value	Description
fill_parent	-1	The view should be as big as its parent (minus padding). This constant is deprecated starting from API Level 8 and is replaced by <code>match_parent</code> .

`match_parent` -1 The view should be as big as its parent (minus padding). Introduced in API Level 8.

`wrap_content` -2 The view should be only big enough to enclose its content (plus padding).

This corresponds to the global attribute resource symbol [layout\\_height](#)  
[\(/reference/android/R.attr.html#layout\\_height\)](#).

Constant Value: 2 (0x00000002)

#### **public static final int `LinearLayout_Layout_layout_weight`**

This symbol is the offset where the [layout\\_weight](#) ([\(/reference/android/R.attr.html#layout\\_weight\)](#)) attribute's value can be found in the [LinearLayout\\_Layout](#)  
[\(/reference/android/R.styleable.html#LinearLayout\\_Layout\)](#) array.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 3 (0x00000003)

#### **public static final int `LinearLayout_Layout_layout_width`**

Specifies the basic width of the view. This is a required attribute for any view inside of a containing layout manager. Its value may be a dimension (such as "12dp") for a constant width or one of the special constants.

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

May be one of the following constant values.

Constant	Value	Description
<code>fill_parent</code>	-1	The view should be as big as its parent (minus padding). This constant is deprecated starting from API Level 8 and is replaced by <code>match_parent</code> .
<code>match_parent</code>	-1	The view should be as big as its parent (minus padding). Introduced in API Level 8.
<code>wrap_content</code>	-2	The view should be only big enough to enclose its content (plus padding).

This corresponds to the global attribute resource symbol [layout\\_width](#)  
[\(/reference/android/R.attr.html#layout\\_width\)](#).

Constant Value: 1 (0x00000001)

#### **public static final int `LinearLayout_baselineAligned`**

When set to false, prevents the layout from aligning its children's baselines. This attribute is particularly useful when the children use different values for gravity. The default value is true.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [baselineAligned](#)  
[\(/reference/android/R.attr.html#baselineAligned\)](#).

Constant Value: 2 (0x00000002)

#### **public static final int `LinearLayout_baselineAlignedChildIndex`**

When a linear layout is part of another layout that is baseline aligned, it can specify which of its children to baseline align to (that is, which child TextView).

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [baselineAlignedChildIndex](#)  
[\(/reference/android/R.attr.html#baselineAlignedChildIndex\)](#).

Constant Value: 3 (0x00000003)

### **public static final int LinearLayout\_divider**

Drawable to use as a vertical divider between buttons.

May be a reference to another resource, in the form "@[+][ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol divider  
[\(/reference/android/R.attr.html#divider\)](#).

Constant Value: 5 (0x00000005)

### **public static final int LinearLayout\_dividerPadding**

Size of padding on either end of a divider.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol dividerPadding  
[\(/reference/android/R.attr.html#dividerPadding\)](#).

Constant Value: 8 (0x00000008)

### **public static final int LinearLayout\_gravity**

Specifies how an object should position its content, on both the X and Y axes, within its own bounds.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
top	0x30	Push object to the top of its container, not changing its size.
bottom	0x50	Push object to the bottom of its container, not changing its size.
left	0x03	Push object to the left of its container, not changing its size.
right	0x05	Push object to the right of its container, not changing its size.
center_vertical	0x10	Place object in the vertical center of its container, not changing its size.
fill_vertical	0x70	Grow the vertical size of the object if needed so it completely fills its container.
center_horizontal	0x01	Place object in the horizontal center of its container, not changing its size.
fill_horizontal	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
center	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
fill	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
clip_vertical	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
clip_horizontal	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a right gravity will clip the left edge, and neither will clip both edges.
start	0x00800003	Push object to the beginning of its container, not changing its size.
end	0x00800005	Push object to the end of its container, not changing its size.

This corresponds to the global attribute resource symbol gravity  
[\(/reference/android/R.attr.html#gravity\)](#).

Constant Value: 0 (0x00000000)

### **public static final int LinearLayout\_measureWithLargestChild**

When set to true, all children with a weight will be considered having the minimum size of the largest child. If false, all children are measured normally.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [measureWithLargestChild](#)  
(/reference/android/R.attr.html#measureWithLargestChild).

Constant Value: 6 (0x00000006)

#### public static final int **LinearLayout\_orientation**

Should the layout be a column or a row? Use "horizontal" for a row, "vertical" for a column. The default is horizontal.

Must be one of the following constant values.

Constant	Value	Description
horizontal	0	Defines an horizontal widget.
vertical	1	Defines a vertical widget.

This corresponds to the global attribute resource symbol [orientation](#)  
(/reference/android/R.attr.html#orientation).

Constant Value: 1 (0x00000001)

#### public static final int **LinearLayout\_showDividers**

Setting for which dividers to show.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
none	0	
beginning	1	
middle	2	
end	4	

This corresponds to the global attribute resource symbol [showDividers](#)  
(/reference/android/R.attr.html#showDividers).

Constant Value: 7 (0x00000007)

#### public static final int **LinearLayout\_weightSum**

Defines the maximum weight sum. If unspecified, the sum is computed by adding the layout\_weight of all of the children. This can be used for instance to give a single child 50% of the total available space by giving it a layout\_weight of 0.5 and setting the weightSum to 1.0.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [weightSum](#)  
(/reference/android/R.attr.html#weightSum).

Constant Value: 4 (0x00000004)

#### public static final int **ListPreference\_entries**

The human-readable array to present as a list. Each entry must have a corresponding index in entryValues.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [entries](#)  
(/reference/android/R.attr.html#entries).

Constant Value: 0 (0x00000000)

#### public static final int **ListPreference\_entryValues**

The array to find the value to save for a preference when an entry from entries is selected. If a user clicks on the second item in entries, the second item in this array will be saved to the preference.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [entryValues](#)  
[\(/reference/android/R.attr.html#entryValues\)](#).

Constant Value: 1 (0x00000001)

#### **public static final int ListView\_divider**

Drawable or color to draw between list items.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [divider](#)  
[\(/reference/android/R.attr.html#divider\)](#).

Constant Value: 1 (0x00000001)

#### **public static final int ListView\_dividerHeight**

Height of the divider. Will use the intrinsic height of the divider if this is not specified.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [dividerHeight](#)  
[\(/reference/android/R.attr.html#dividerHeight\)](#).

Constant Value: 2 (0x00000002)

#### **public static final int ListView\_entries**

Reference to an array resource that will populate the ListView. For static content, this is simpler than populating the ListView programmatically.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [entries](#)  
[\(/reference/android/R.attr.html#entries\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int ListView\_footerDividersEnabled**

When set to false, the ListView will not draw the divider before each footer view. The default value is true.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [footerDividersEnabled](#)  
[\(/reference/android/R.attr.html#footerDividersEnabled\)](#).

Constant Value: 4 (0x00000004)

#### **public static final int ListView\_headerDividersEnabled**

When set to false, the ListView will not draw the divider after each header view. The default value is true.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [headerDividersEnabled](#)  
[\(/reference/android/R.attr.html#headerDividersEnabled\)](#).

Constant Value: 3 (0x00000003)

#### **public static final int ListView\_overScrollFooter**

Drawable to draw below list content.

May be a reference to another resource, in the form "`@[+][ package: ]type: name`" or to a theme attribute in the form "`?[ package: ][ type: ]name`".

May be a color value, in the form of "#`rgb`", "#`argb`", "#`rrggbb`", or "#`aarrggbb`".

This corresponds to the global attribute resource symbol [overScrollFooter](#)  
[\(/reference/android/R.attr.html#overScrollFooter\)](#).

Constant Value: 6 (0x00000006)

#### **public static final int ListView\_overScrollHeader**

Drawable to draw above list content.

May be a reference to another resource, in the form "`@[+][ package: ]type: name`" or to a theme attribute in the form "`?[ package: ][ type: ]name`".

May be a color value, in the form of "#`rgb`", "#`argb`", "#`rrggbb`", or "#`aarrggbb`".

This corresponds to the global attribute resource symbol [overScrollHeader](#)  
[\(/reference/android/R.attr.html#overScrollHeader\)](#).

Constant Value: 5 (0x00000005)

#### **public static final int MapView\_apiKey**

Value is a string that specifies the Maps API Key to use.

Must be a string value, using '\\.' to escape characters such as '\\n' or '\\uxxxx' for a unicode character.

This may also be a reference to a resource (in the form "`@[ package: ]type: name`") or theme attribute (in the form "`?[ package: ][ type: ]name`") containing a value of this type.

This corresponds to the global attribute resource symbol [apiKey](#)  
[\(/reference/android/R.attr.html#apiKey\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int MediaRouteButton\_mediaRouteTypes**

The types of media routes the button and its resulting chooser will filter by.

May be an integer value, such as "100".

This may also be a reference to a resource (in the form "`@[ package: ]type: name`") or theme attribute (in the form "`?[ package: ][ type: ]name`") containing a value of this type.

May be one of the following constant values.

Constant	Value	Description
liveAudio	0x1	Allow selection of live audio routes.
user	0x800000	Allow selection of user (app-specified) routes.

This corresponds to the global attribute resource symbol [mediaRouteTypes](#)  
[\(/reference/android/R.attr.html#mediaRouteTypes\)](#).

Constant Value: 2 (0x00000002)

#### **public static final int MediaRouteButton\_minHeight**

This symbol is the offset where the [minHeight](#) [\(/reference/android/R.attr.html#minHeight\)](#) attribute's value can be found in the [MediaRouteButton](#)  
[\(/reference/android/R.styleable.html#MediaRouteButton\)](#) array.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "`@[ package: ]type: name`") or theme attribute (in the form "`?[ package: ][ type: ]name`") containing a value of this type.

Constant Value: 1 (0x00000001)

#### **public static final int MediaRouteButton\_minWidth**

This symbol is the offset where the [minWidth](#) [\(/reference/android/R.attr.html#minWidth\)](#) attribute's

value can be found in the [MediaRouteButton](#)  
([/reference/android/R.styleable.html#MediaRouteButton](#)) array.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:]type:name") containing a value of this type.

Constant Value: 0 (0x00000000)

#### public static final int **MenuGroup\_checkableBehavior**

Whether the items are capable of displaying a check mark.

Must be one of the following constant values.

Constant	Value	Description
none	0	The items are not checkable.
all	1	The items are all checkable.
single	2	The items are checkable and there will only be a single checked item in this group.

This corresponds to the global attribute resource symbol [checkableBehavior](#)  
([/reference/android/R.attr.html#checkableBehavior](#)).

Constant Value: 5 (0x00000005)

#### public static final int **MenuGroup\_enabled**

Whether the items are enabled.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [enabled](#)  
([/reference/android/R.attr.html#enabled](#)).

Constant Value: 0 (0x00000000)

#### public static final int **MenuGroup\_id**

The ID of the group.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:]type:name".

This corresponds to the global attribute resource symbol [id](#) ([/reference/android/R.attr.html#id](#)).

Constant Value: 1 (0x00000001)

#### public static final int **MenuGroup\_menuCategory**

The category applied to all items within this group. (This will be or'ed with the orderInCategory attribute.)

Must be one of the following constant values.

Constant	Value	Description
container	0x00010000	Items are part of a container.
system	0x00020000	Items are provided by the system.
secondary	0x00030000	Items are user-supplied secondary (infrequently used).
alternative	0x00040000	Items are alternative actions.

This corresponds to the global attribute resource symbol [menuCategory](#)  
([/reference/android/R.attr.html#menuCategory](#)).

Constant Value: 3 (0x00000003)

#### public static final int **MenuGroup\_orderInCategory**

The order within the category applied to all items within this group. (This will be or'ed with the category attribute.)

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in

the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [orderInCategory](#) ([/reference/android/R.attr.html#orderInCategory](#)).

Constant Value: 4 (0x00000004)

#### **public static final int MenuGroup\_visible**

Whether the items are shown/visible.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "?[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [visible](#) ([/reference/android/R.attr.html#visible](#)).

Constant Value: 2 (0x00000002)

#### **public static final int MenuItemCheckedFocusedState\_state\_checkable**

State identifier indicating that the object *may* display a check mark. See [state\\_checked](#) ([/reference/android/R.attr.html#state\\_checked](#)) for the identifier that indicates whether it is actually checked.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "?[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_checkable](#) ([/reference/android/R.attr.html#state\\_checkable](#)).

Constant Value: 1 (0x00000001)

#### **public static final int MenuItemCheckedFocusedState\_state\_checked**

State identifier indicating that the object is currently checked. See [state\\_checkable](#) ([/reference/android/R.attr.html#state\\_checkable](#)) for an additional identifier that can indicate if any object may ever display a check, regardless of whether state\_checked is currently set.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "?[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_checked](#) ([/reference/android/R.attr.html#state\\_checked](#)).

Constant Value: 2 (0x00000002)

#### **public static final int MenuItemCheckedFocusedState\_state\_focused**

State value for [StateListDrawable](#) ([/reference/android/graphics/drawable/StateListDrawable.html](#)), set when a view has input focus.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "?[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_focused](#) ([/reference/android/R.attr.html#state\\_focused](#)).

Constant Value: 0 (0x00000000)

#### **public static final int MenuItemCheckedState\_state\_checkable**

State identifier indicating that the object *may* display a check mark. See [state\\_checked](#) ([/reference/android/R.attr.html#state\\_checked](#)) for the identifier that indicates whether it is actually checked.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "?[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_checked](#)  
[\(/reference/android/R.attr.html#state\\_checked\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int MenuItemCheckedState\_state\_checked**

State identifier indicating that the object is currently checked. See [state\\_checked](#)  
[\(/reference/android/R.attr.html#state\\_checked\)](#) for an additional identifier that can indicate if any object may ever display a check, regardless of whether state\_checked is currently set.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_checked](#)  
[\(/reference/android/R.attr.html#state\\_checked\)](#).

Constant Value: 1 (0x00000001)

#### **public static final int MenuItemUncheckedFocusedState\_state\_checkable**

State identifier indicating that the object *may* display a check mark. See [state\\_checked](#)  
[\(/reference/android/R.attr.html#state\\_checked\)](#) for the identifier that indicates whether it is actually checked.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_checked](#)  
[\(/reference/android/R.attr.html#state\\_checked\)](#).

Constant Value: 1 (0x00000001)

#### **public static final int MenuItemUncheckedFocusedState\_state\_focused**

State value for [StateListDrawable](#)  
[\(/reference/android/graphics/drawable/StateListDrawable.html\)](#), set when a view has input focus.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_focused](#)  
[\(/reference/android/R.attr.html#state\\_focused\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int MenuItemUncheckedState\_state\_checkable**

State identifier indicating that the object *may* display a check mark. See [state\\_checked](#)  
[\(/reference/android/R.attr.html#state\\_checked\)](#) for the identifier that indicates whether it is actually checked.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_checked](#)  
[\(/reference/android/R.attr.html#state\\_checked\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int MenuItem\_actionLayout**

An optional layout to be used as an action view. See [setActionView\(android.view.View\)](#)  
[\(/reference/android/view/MenuItem.html#setActionView\(android.view.View\)\)](#) for more info.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:]type:name".

This corresponds to the global attribute resource symbol [actionLayout](#)

[\(/reference/android/R.attr.html#actionLayout\)](#)

Constant Value: 14 (0x0000000e)

#### **public static final int MenuItem\_actionProviderClass**

The name of an optional ActionProvider class to instantiate an action view and perform operations such as default action for that menu item. See [setActionProvider\(android.view.ActionProvider\)](#)  
[\(/reference/android/view/MenuItem.html#setActionProvider\(android.view.ActionProvider\)\)](#) for more info.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [actionProviderClass](#)  
[\(/reference/android/R.attr.html#actionProviderClass\)](#).

Constant Value: 16 (0x00000010)

#### **public static final int MenuItem\_actionViewClass**

The name of an optional View class to instantiate and use as an action view. See [setActionView\(android.view.View\)](#)  
[\(/reference/android/view/MenuItem.html#setActionView\(android.view.View\)\)](#) for more info.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [actionViewClass](#)  
[\(/reference/android/R.attr.html#actionViewClass\)](#).

Constant Value: 15 (0x0000000f)

#### **public static final int MenuItem\_alphabeticShortcut**

The alphabetic shortcut key. This is the shortcut when using a keyboard with alphabetic keys.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [alphabeticShortcut](#)  
[\(/reference/android/R.attr.html#alphabeticShortcut\)](#).

Constant Value: 9 (0x00000009)

#### **public static final int MenuItem\_checkable**

Whether the item is capable of displaying a check mark.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [checkable](#)  
[\(/reference/android/R.attr.html#checkable\)](#).

Constant Value: 11 (0x0000000b)

#### **public static final int MenuItem\_checked**

Whether the item is checked. Note that you must first have enabled checking with the checkable attribute or else the check mark will not appear.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [checked](#)  
[\(/reference/android/R.attr.html#checked\)](#).

Constant Value: 3 (0x00000003)

## **public static final int MenuItem\_enabled**

Whether the item is enabled.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [enabled](#)  
(/reference/android/R.attr.html#enabled).

Constant Value: 1 (0x00000001)

## **public static final int MenuItem\_icon**

The icon associated with this item. This icon will not always be shown, so the title should be sufficient in describing this item.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [icon](#) (/reference/android/R.attr.html#icon).

Constant Value: 0 (0x00000000)

## **public static final int MenuItem\_id**

The ID of the item.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [id](#) (/reference/android/R.attr.html#id).

Constant Value: 2 (0x00000002)

## **public static final int MenuItem\_menuCategory**

The category applied to the item. (This will be or'd with the orderInCategory attribute.)

Must be one of the following constant values.

Constant	Value	Description
container	0x00010000	Items are part of a container.
system	0x00020000	Items are provided by the system.
secondary	0x00030000	Items are user-supplied secondary (infrequently used).
alternative	0x00040000	Items are alternative actions.

This corresponds to the global attribute resource symbol [menuCategory](#)  
(/reference/android/R.attr.html#menuCategory).

Constant Value: 5 (0x00000005)

## **public static final int MenuItem\_numericShortcut**

The numeric shortcut key. This is the shortcut when using a numeric (e.g., 12-key) keyboard.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [numericShortcut](#)  
(/reference/android/R.attr.html#numericShortcut).

Constant Value: 10 (0x0000000a)

## **public static final int MenuItem\_onClick**

Name of a method on the Context used to inflate the menu that will be called when the item is clicked.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [onClick](#)

[\(/reference/android/R.attr.html#onClick\)](#).

Constant Value: 12 (0x0000000c)

#### public static final int **MenuItem\_orderInCategory**

The order within the category applied to the item. (This will be or'ed with the category attribute.)

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [orderInCategory](#)  
[\(/reference/android/R.attr.html#orderInCategory\)](#).

Constant Value: 6 (0x00000006)

#### public static final int **MenuItem\_showAsAction**

How this item should display in the Action Bar, if present.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
never	0	Never show this item in an action bar, show it in the overflow menu instead. Mutually exclusive with "ifRoom" and "always".
ifRoom	1	Show this item in an action bar if there is room for it as determined by the system. Favor this option over "always" where possible. Mutually exclusive with "never" and "always".
always	2	Always show this item in an actionbar, even if it would override the system's limits of how much stuff to put there. This may make your action bar look bad on some screens. In most cases you should use "ifRoom" instead. Mutually exclusive with "ifRoom" and "never".
withText	4	When this item is shown as an action in the action bar, show a text label with it even if it has an icon representation.
collapseActionView	8	This item's action view collapses to a normal menu item. When expanded, the action view takes over a larger segment of its container.

This corresponds to the global attribute resource symbol [showAsAction](#)  
[\(/reference/android/R.attr.html#showAsAction\)](#).

Constant Value: 13 (0x0000000d)

#### public static final int **MenuItem\_title**

The title associated with the item.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [title](#)  
[\(/reference/android/R.attr.html#title\)](#).

Constant Value: 7 (0x00000007)

#### public static final int **MenuItem\_titleCondensed**

The condensed title associated with the item. This is used in situations where the normal title may be too long to be displayed.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [titleCondensed](#)  
[\(/reference/android/R.attr.html#titleCondensed\)](#).

Constant Value: 8 (0x00000008)

#### public static final int **MenuItem\_visible**

Whether the item is shown/visible.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*:]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [visible](#)  
[\(/reference/android/R.attr.html#visible\)](#).

Constant Value: 4 (0x00000004)

#### **public static final int MenuView\_headerBackground**

Default background for the menu header.

May be a reference to another resource, in the form "@[+][*package*: ]*type*:*name*" or to a theme attribute in the form "?[*package*: ][*type*:]*name*".

May be a color value, in the form "#*rgb*", "#*argb*", "#*rrggbb*", or "#*aarrggbb*".

This corresponds to the global attribute resource symbol [headerBackground](#)  
[\(/reference/android/R.attr.html#headerBackground\)](#).

Constant Value: 4 (0x00000004)

#### **public static final int MenuView\_horizontalDivider**

Default horizontal divider between rows of menu items.

Must be a reference to another resource, in the form "@[+][*package*: ]*type*:*name*" or to a theme attribute in the form "?[*package*: ][*type*:]*name*".

This corresponds to the global attribute resource symbol [horizontalDivider](#)  
[\(/reference/android/R.attr.html#horizontalDivider\)](#).

Constant Value: 2 (0x00000002)

#### **public static final int MenuView\_itemBackground**

Default background for each menu item.

May be a reference to another resource, in the form "@[+][*package*: ]*type*:*name*" or to a theme attribute in the form "?[*package*: ][*type*:]*name*".

May be a color value, in the form "#*rgb*", "#*argb*", "#*rrggbb*", or "#*aarrggbb*".

This corresponds to the global attribute resource symbol [itemBackground](#)  
[\(/reference/android/R.attr.html#itemBackground\)](#).

Constant Value: 5 (0x00000005)

#### **public static final int MenuView\_itemIconDisabledAlpha**

Default disabled icon alpha for each menu item that shows an icon.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*:]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [itemIconDisabledAlpha](#)  
[\(/reference/android/R.attr.html#itemIconDisabledAlpha\)](#).

Constant Value: 6 (0x00000006)

#### **public static final int MenuView\_itemTextAppearance**

Default appearance of menu item text.

Must be a reference to another resource, in the form "@[+][*package*: ]*type*:*name*" or to a theme attribute in the form "?[*package*: ][*type*:]*name*".

This corresponds to the global attribute resource symbol [itemTextAppearance](#)  
[\(/reference/android/R.attr.html#itemTextAppearance\)](#).

Constant Value: 1 (0x00000001)

#### **public static final int MenuView\_verticalDivider**

Default vertical divider between menu items.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [verticaldivider](#)  
(/reference/android/R.attr.html#verticaldivider).

Constant Value: 3 (0x00000003)

#### public static final int **MenuView\_windowAnimationStyle**

Default animations for the menu.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [windowAnimationStyle](#)  
(/reference/android/R.attr.html#windowAnimationStyle).

Constant Value: 0 (0x00000000)

#### public static final int **MipmapDrawableItem\_drawable**

Reference to a drawable resource to use for the frame. If not given, the drawable must be defined by the first child tag.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [drawable](#)  
(/reference/android/R.attr.html#drawable).

Constant Value: 0 (0x00000000)

#### public static final int **MultiPaneChallengeLayout\_Layout\_layout\_gravity**

Standard gravity constant that a child supplies to its parent. Defines how the child view should be positioned, on both the X and Y axes, within its enclosing layout.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
top	0x30	Push object to the top of its container, not changing its size.
bottom	0x50	Push object to the bottom of its container, not changing its size.
left	0x03	Push object to the left of its container, not changing its size.
right	0x05	Push object to the right of its container, not changing its size.
center_vertical	0x10	Place object in the vertical center of its container, not changing its size.
fill_vertical	0x70	Grow the vertical size of the object if needed so it completely fills its container.
center_horizontal	0x01	Place object in the horizontal center of its container, not changing its size.
fill_horizontal	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
center	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
fill	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
clip_vertical	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
clip_horizontal	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a right gravity will clip the left edge, and neither will clip both edges.
start	0x00800003	Push object to the beginning of its container, not changing its size.
end	0x00800005	Push object to the end of its container, not changing its size.

This corresponds to the global attribute resource symbol [layout\\_gravity](#)  
(/reference/android/R.attr.html#layout\_gravity).

Constant Value: 0 (0x00000000)

## **public static final int MultiPaneChallengeLayout\_orientation**

Influences how layout\_centerWithinArea behaves

Must be one of the following constant values.

Constant	Value	Description
horizontal	0	Defines an horizontal widget.
vertical	1	Defines a vertical widget.

This corresponds to the global attribute resource symbol [orientation](#)  
[\(/reference/android/R.attr.html#orientation\)](#).

Constant Value: 0 (0x00000000)

## **public static final int MultiSelectListPreference\_entries**

The human-readable array to present as a list. Each entry must have a corresponding index in entryValues.

Must be a reference to another resource, in the form "@[+][package: ]type: name" or to a theme attribute in the form "?[package: ][type: ]name".

This corresponds to the global attribute resource symbol [entries](#)  
[\(/reference/android/R.attr.html#entries\)](#).

Constant Value: 0 (0x00000000)

## **public static final int MultiSelectListPreference\_entryValues**

The array to find the value to save for a preference when an entry from entries is selected. If a user clicks the second item in entries, the second item in this array will be saved to the preference.

Must be a reference to another resource, in the form "@[+][package: ]type: name" or to a theme attribute in the form "?[package: ][type: ]name".

This corresponds to the global attribute resource symbol [entryValues](#)  
[\(/reference/android/R.attr.html#entryValues\)](#).

Constant Value: 1 (0x00000001)

## **public static final int MultiWaveView\_directionDescriptions**

Reference to an array resource that be used to announce the directions with targets around the circle.

Must be a reference to another resource, in the form "@[+][package: ]type: name" or to a theme attribute in the form "?[package: ][type: ]name".

This corresponds to the global attribute resource symbol [directionDescriptions](#)  
[\(/reference/android/R.attr.html#directionDescriptions\)](#).

Constant Value: 1 (0x00000001)

## **public static final int MultiWaveView\_targetDescriptions**

Reference to an array resource that be used as description for the targets around the circle.

Must be a reference to another resource, in the form "@[+][package: ]type: name" or to a theme attribute in the form "?[package: ][type: ]name".

This corresponds to the global attribute resource symbol [targetDescriptions](#)  
[\(/reference/android/R.attr.html#targetDescriptions\)](#).

Constant Value: 0 (0x00000000)

## **public static final int NinePatchDrawable\_dither**

Enables or disables dithering of the bitmap if the bitmap does not have the same pixel configuration as the screen (for instance: a ARGB 8888 bitmap with an RGB 565 screen).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package: ]type: name") or theme attribute (in the form "?[package: ][type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [dither](#)  
[\(/reference/android/R.attr.html#dither\)](#).

Constant Value: 1 (0x00000001)

## **public static final int NinePatchDrawable\_src**

Identifier of the bitmap file. This attribute is mandatory.

May be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ] [ type: ] name`".

May be a color value, in the form of "#`rgb`", "#`argb`", "#`rrggbb`", or "#`aarrggbb`".

This corresponds to the global attribute resource symbol [src](#) ([/reference/android/R.attr.html#src](#)).

Constant Value: 0 (0x00000000)

## **public static final int OvershootInterpolator\_tension**

This is the amount of tension.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ] [ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [tension](#) ([/reference/android/R.attr.html#tension](#)).

Constant Value: 0 (0x00000000)

## **public static final int PopupWindowBackgroundState\_state\_above\_anchor**

State identifier indicating the popup will be above the anchor.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ] [ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_above\\_anchor](#) ([/reference/android/R.attr.html#state\\_above\\_anchor](#)).

Constant Value: 0 (0x00000000)

## **public static final int PopupWindow\_popupAnimationStyle**

This symbol is the offset where the [popupAnimationStyle](#) ([/reference/android/R.attr.html#popupAnimationStyle](#)) attribute's value can be found in the [PopupWindow](#) ([/reference/android/R.styleable.html#PopupWindow](#)) array.

Must be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ] [ type: ] name`".

Constant Value: 1 (0x00000001)

## **public static final int PopupWindow\_popupBackground**

This symbol is the offset where the [popupBackground](#) ([/reference/android/R.attr.html#popupBackground](#)) attribute's value can be found in the [PopupWindow](#) ([/reference/android/R.styleable.html#PopupWindow](#)) array.

May be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ] [ type: ] name`".

May be a color value, in the form of "#`rgb`", "#`argb`", "#`rrggbb`", or "#`aarrggbb`".

Constant Value: 0 (0x00000000)

## **public static final int PreferenceGroup\_orderingFromXml**

Whether to order the Preference under this group as they appear in the XML file. If this is false, the ordering will follow the Preference order attribute and default to alphabetic for those without the order attribute.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ] [ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [orderingFromXml](#) ([/reference/android/R.attr.html#orderingFromXml](#)).

Constant Value: 0 (0x00000000)

#### **public static final int PreferenceHeader\_breadCrumbShortTitle**

The short title for the bread crumb of this item.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [breadCrumbShortTitle](#) ([/reference/android/R.attr.html#breadCrumbShortTitle](#)).

Constant Value: 6 (0x00000006)

#### **public static final int PreferenceHeader\_breadCrumbTitle**

The title for the bread crumb of this item.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [breadCrumbTitle](#) ([/reference/android/R.attr.html#breadCrumbTitle](#)).

Constant Value: 5 (0x00000005)

#### **public static final int PreferenceHeader\_fragment**

The fragment that is displayed when the user selects this item.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [fragment](#) ([/reference/android/R.attr.html#fragment](#)).

Constant Value: 4 (0x00000004)

#### **public static final int PreferenceHeader\_icon**

An icon for the item.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [icon](#) ([/reference/android/R.attr.html#icon](#)).

Constant Value: 0 (0x00000000)

#### **public static final int PreferenceHeader\_id**

Identifier value for the header.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [id](#) ([/reference/android/R.attr.html#id](#)).

Constant Value: 1 (0x00000001)

#### **public static final int PreferenceHeader\_summary**

The summary for the item.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [summary](#) ([/reference/android/R.attr.html#summary](#)).

Constant Value: 3 (0x00000003)

## **public static final int PreferenceHeader\_title**

The title of the item that is shown to the user.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [title](#)  
[\(/reference/android/R.attr.html#title\)](#).

Constant Value: 2 (0x00000002)

## **public static final int Preference\_defaultValue**

The default value for the preference, which will be set either if persistence is off or persistence is on and the preference is not found in the persistent storage.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

May be an integer value, such as "100".

May be a boolean value, either "true" or "false".

May be a floating point value, such as "1.2".

This corresponds to the global attribute resource symbol [defaultValue](#)  
[\(/reference/android/R.attr.html#defaultValue\)](#).

Constant Value: 11 (0x0000000b)

## **public static final int Preference\_dependency**

The key of another Preference that this Preference will depend on. If the other Preference is not set or is off, this Preference will be disabled.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [dependency](#)  
[\(/reference/android/R.attr.html#dependency\)](#).

Constant Value: 10 (0x0000000a)

## **public static final int Preference\_enabled**

Whether the Preference is enabled.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [enabled](#)  
[\(/reference/android/R.attr.html#enabled\)](#).

Constant Value: 2 (0x00000002)

## **public static final int Preference\_fragment**

When used inside of a modern PreferenceActivity, this declares a new PreferenceFragment to be shown when the user selects this item.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [fragment](#)  
[\(/reference/android/R.attr.html#fragment\)](#).

Constant Value: 13 (0x0000000d)

## public static final int Preference\_icon

The optional icon for the preference

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [icon](#) ([/reference/android/R.attr.html#icon](#)).

Constant Value: 0 (0x00000000)

## public static final int Preference\_key

The key to store the Preference value.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [key](#) ([/reference/android/R.attr.html#key](#)).

Constant Value: 6 (0x00000006)

## public static final int Preference\_layout

The layout for the Preference in a PreferenceActivity screen. This should rarely need to be changed, look at widgetLayout instead.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [layout](#) ([/reference/android/R.attr.html#layout](#)).

Constant Value: 3 (0x00000003)

## public static final int Preference\_order

The order for the Preference (lower values are to be ordered first). If this is not specified, the default order in will be alphabetic.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [order](#) ([/reference/android/R.attr.html#order](#)).

Constant Value: 8 (0x00000008)

## public static final int Preference\_persistent

Whether the Preference stores its value to the shared preferences.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [persistent](#) ([/reference/android/R.attr.html#persistent](#)).

Constant Value: 1 (0x00000001)

## public static final int Preference\_selectable

Whether the Preference is selectable.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [selectable](#) ([/reference/android/R.attr.html#selectable](#)).

Constant Value: 5 (0x00000005)

## **public static final int Preference\_shouldDisableView**

Whether the view of this Preference should be disabled when this Preference is disabled.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [shouldDisableView](#)  
(/reference/android/R.attr.html#shouldDisableView).

Constant Value: 12 (0x0000000c)

## **public static final int Preference\_summary**

The summary for the Preference in a PreferenceActivity screen.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [summary](#)  
(/reference/android/R.attr.html#summary).

Constant Value: 7 (0x00000007)

## **public static final int Preference\_title**

The title for the Preference in a PreferenceActivity screen.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [title](#)  
(/reference/android/R.attr.html#title).

Constant Value: 4 (0x00000004)

## **public static final int Preference\_widgetLayout**

The layout for the controllable widget portion of a Preference. This is inflated into the layout for a Preference and should be used more frequently than the layout attribute. For example, a checkbox preference would specify a custom layout (consisting of just the CheckBox) here.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [widgetLayout](#)  
(/reference/android/R.attr.html#widgetLayout).

Constant Value: 9 (0x00000009)

## **public static final int ProgressBar\_animationResolution**

Timeout between frames of animation in milliseconds

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [animationResolution](#)  
(/reference/android/R.attr.html#animationResolution).

Constant Value: 14 (0x0000000e)

## **public static final int ProgressBar\_ineterminate**

Allows to enable the indeterminate mode. In this mode the progress bar plays an infinite looping animation.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [indeterminate](#) ([/reference/android/R.attr.html#indeterminate](#)).

Constant Value: 5 (0x00000005)

#### public static final int **ProgressBar\_ineterminateBehavior**

Defines how the indeterminate mode should behave when the progress reaches max.

Must be one of the following constant values.

Constant Value	Description
repeat 1	Progress starts over from 0.
cycle 2	Progress keeps the current value and goes back to 0.

This corresponds to the global attribute resource symbol [indeterminateBehavior](#) ([/reference/android/R.attr.html#indeterminateBehavior](#)).

Constant Value: 10 (0x0000000a)

#### public static final int **ProgressBar\_ineterminateDrawable**

Drawable used for the indeterminate mode.

Must be a reference to another resource, in the form "@[+][ package: ] type: name" or to a theme attribute in the form "?[ package: ][ type: ] name".

This corresponds to the global attribute resource symbol [indeterminateDrawable](#) ([/reference/android/R.attr.html#indeterminateDrawable](#)).

Constant Value: 7 (0x00000007)

#### public static final int **ProgressBar\_ineterminateDuration**

Duration of the indeterminate animation.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ][ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [indeterminateDuration](#) ([/reference/android/R.attr.html#indeterminateDuration](#)).

Constant Value: 9 (0x00000009)

#### public static final int **ProgressBar\_ineterminateOnly**

Restricts to ONLY indeterminate mode (state-keeping progress mode will not work).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ][ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [indeterminateOnly](#) ([/reference/android/R.attr.html#indeterminateOnly](#)).

Constant Value: 6 (0x00000006)

#### public static final int **ProgressBar\_interpolator**

This symbol is the offset where the [interpolator](#) ([/reference/android/R.attr.html#interpolator](#)) attribute's value can be found in the [ProgressBar](#) ([/reference/android/R.styleable.html#ProgressBar](#)) array.

Must be a reference to another resource, in the form "@[+][ package: ] type: name" or to a theme attribute in the form "?[ package: ][ type: ] name".

Constant Value: 13 (0x0000000d)

#### public static final int **ProgressBar\_max**

Defines the maximum value the progress can take.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in

the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [max](#) ([/reference/android/R.attr.html#max](#)).

Constant Value: 2 (0x00000002)

#### public static final int **ProgressBar\_maxHeight**

An optional argument to supply a maximum height for this view. See {see [android.widget.ImageView#setMaxHeight](#)} for details.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "?[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [maxHeight](#) ([/reference/android/R.attr.html#maxHeight](#)).

Constant Value: 1 (0x00000001)

#### public static final int **ProgressBar\_maxWidth**

An optional argument to supply a maximum width for this view. See {see [android.widget.ImageView#setMaxWidth](#)} for details.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "?[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [maxWidth](#) ([/reference/android/R.attr.html#maxWidth](#)).

Constant Value: 0 (0x00000000)

#### public static final int **ProgressBar\_minHeight**

This symbol is the offset where the [minHeight](#) ([/reference/android/R.attr.html#minHeight](#)) attribute's value can be found in the [ProgressBar](#) ([/reference/android/R.styleable.html#ProgressBar](#)) array.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "?[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 12 (0x0000000c)

#### public static final int **ProgressBar\_minWidth**

This symbol is the offset where the [minWidth](#) ([/reference/android/R.attr.html#minWidth](#)) attribute's value can be found in the [ProgressBar](#) ([/reference/android/R.styleable.html#ProgressBar](#)) array.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "?[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 11 (0x0000000b)

#### public static final int **ProgressBar\_progress**

Defines the default progress value, between 0 and max.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "?[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [progress](#)

[\(/reference/android/R.attr.html#progress\)](#).

Constant Value: 3 (0x00000003)

#### public static final int **ProgressBar\_progressDrawable**

Drawable used for the progress mode.

Must be a reference to another resource, in the form "@[+][ package: ] type: name" or to a theme attribute in the form "?[ package: ][ type: ] name".

This corresponds to the global attribute resource symbol [progressDrawable](#)  
[\(/reference/android/R.attr.html#progressDrawable\)](#).

Constant Value: 8 (0x00000008)

#### public static final int **ProgressBar\_secondaryProgress**

Defines the secondary progress value, between 0 and max. This progress is drawn between the primary progress and the background. It can be ideal for media scenarios such as showing the buffering progress while the default progress shows the play progress.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ][ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [secondaryProgress](#)  
[\(/reference/android/R.attr.html#secondaryProgress\)](#).

Constant Value: 4 (0x00000004)

#### public static final int **PropertyAnimator\_propertyName**

Name of the property being animated.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ][ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [propertyName](#)  
[\(/reference/android/R.attr.html#propertyName\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **RadioGroup\_checkedButton**

The id of the child radio button that should be checked by default within this radio group.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ][ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [checkedButton](#)  
[\(/reference/android/R.attr.html#checkedButton\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **RadioGroup\_orientation**

Should the radio group be a column or a row? Use "horizontal" for a row, "vertical" for a column. The default is vertical.

Must be one of the following constant values.

Constant	Value	Description
horizontal	0	Defines an horizontal widget.
vertical	1	Defines a vertical widget.

This corresponds to the global attribute resource symbol [orientation](#)  
[\(/reference/android/R.attr.html#orientation\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **RatingBar\_isIndicator**

Whether this rating bar is an indicator (and non-changeable by the user).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [isIndicator](#)  
(/reference/android/R.attr.html#isIndicator).

Constant Value: 3 (0x00000003)

#### **public static final int RatingBar\_numStars**

The number of stars (or rating items) to show.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [numStars](#)  
(/reference/android/R.attr.html#numStars).

Constant Value: 0 (0x00000000)

#### **public static final int RatingBar\_rating**

The rating to set by default.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [rating](#)  
(/reference/android/R.attr.html#rating).

Constant Value: 1 (0x00000001)

#### **public static final int RatingBar\_stepSize**

The step size of the rating.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [stepSize](#)  
(/reference/android/R.attr.html#stepSize).

Constant Value: 2 (0x00000002)

#### **public static final int RecognitionService\_settingsActivity**

Component name of an activity that allows the user to modify the settings for this service.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [settingsActivity](#)  
(/reference/android/R.attr.html#settingsActivity).

Constant Value: 0 (0x00000000)

#### **public static final int RelativeLayout\_Layout\_layout\_above**

Positions the bottom edge of this view above the given anchor view ID. Accommodates bottom margin of this view and top margin of anchor view.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [layout\\_above](#)  
(/reference/android/R.attr.html#layout\_above).

Constant Value: 2 (0x00000002)

#### **public static final int RelativeLayout.LayoutParams\_alignBaseline**

Positions the baseline of this view on the baseline of the given anchor view ID.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [layout\\_alignBaseline](#) ([/reference/android/R.attr.html#layout\\_alignBaseline](#)).

Constant Value: 4 (0x00000004)

#### **public static final int RelativeLayout.LayoutParams\_alignBottom**

Makes the bottom edge of this view match the bottom edge of the given anchor view ID. Accommodates bottom margin.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [layout\\_alignBottom](#) ([/reference/android/R.attr.html#layout\\_alignBottom](#)).

Constant Value: 8 (0x00000008)

#### **public static final int RelativeLayout.LayoutParams\_alignEnd**

Makes the end edge of this view match the end edge of the given anchor view ID. Accommodates end margin.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [layout\\_alignEnd](#) ([/reference/android/R.attr.html#layout\\_alignEnd](#)).

Constant Value: 20 (0x00000014)

#### **public static final int RelativeLayout.LayoutParams\_alignLeft**

Makes the left edge of this view match the left edge of the given anchor view ID. Accommodates left margin.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [layout\\_alignLeft](#) ([/reference/android/R.attr.html#layout\\_alignLeft](#)).

Constant Value: 5 (0x00000005)

#### **public static final int RelativeLayout.LayoutParams\_alignParentBottom**

If true, makes the bottom edge of this view match the bottom edge of the parent. Accommodates bottom margin.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_alignParentBottom](#) ([/reference/android/R.attr.html#layout\\_alignParentBottom](#)).

Constant Value: 12 (0x0000000c)

#### **public static final int RelativeLayout.LayoutParams\_alignParentEnd**

If true, makes the end edge of this view match the end edge of the parent. Accommodates end margin.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_alignParentEnd](#)

[\(/reference/android/R.attr.html#layout\\_alignParentEnd\).](#)

Constant Value: 22 (0x00000016)

#### **public static final int RelativeLayout\_Layout\_layout\_alignParentLeft**

If true, makes the left edge of this view match the left edge of the parent. Accommodates left margin.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_alignParentLeft](#)

[\(/reference/android/R.attr.html#layout\\_alignParentLeft\).](#)

Constant Value: 9 (0x00000009)

#### **public static final int RelativeLayout\_Layout\_layout\_alignParentRight**

If true, makes the right edge of this view match the right edge of the parent. Accommodates right margin.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_alignParentRight](#)

[\(/reference/android/R.attr.html#layout\\_alignParentRight\).](#)

Constant Value: 11 (0x0000000b)

#### **public static final int RelativeLayout\_Layout\_layout\_alignParentStart**

If true, makes the start edge of this view match the start edge of the parent. Accommodates start margin.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_alignParentStart](#)

[\(/reference/android/R.attr.html#layout\\_alignParentStart\).](#)

Constant Value: 21 (0x00000015)

#### **public static final int RelativeLayout\_Layout\_layout\_alignParentTop**

If true, makes the top edge of this view match the top edge of the parent. Accommodates top margin.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_alignParentTop](#)

[\(/reference/android/R.attr.html#layout\\_alignParentTop\).](#)

Constant Value: 10 (0x0000000a)

#### **public static final int RelativeLayout\_Layout\_layout\_alignRight**

Makes the right edge of this view match the right edge of the given anchor view ID. Accommodates right margin.

Must be a reference to another resource, in the form "@[ + ][ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [layout\\_alignRight](#)

[\(/reference/android/R.attr.html#layout\\_alignRight\).](#)

Constant Value: 7 (0x00000007)

#### **public static final int RelativeLayout\_Layout\_layout\_alignStart**

Makes the start edge of this view match the start edge of the given anchor view ID. Accommodates start margin.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [layout\\_alignStart](#)  
(/reference/android/R.attr.html#layout\_alignStart).

Constant Value: 19 (0x00000013)

#### **public static final int RelativeLayout\_Layout\_layout\_alignTop**

Makes the top edge of this view match the top edge of the given anchor view ID. Accommodates top margin.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [layout\\_alignTop](#)  
(/reference/android/R.attr.html#layout\_alignTop).

Constant Value: 6 (0x00000006)

#### **public static final int RelativeLayout\_Layout\_layout\_alignWithParentIfMissing**

If set to true, the parent will be used as the anchor when the anchor cannot be found for layout\_toLeftOf, layout\_toRightOf, etc.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*][*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_alignWithParentIfMissing](#)  
(/reference/android/R.attr.html#layout\_alignWithParentIfMissing).

Constant Value: 16 (0x00000010)

#### **public static final int RelativeLayout\_Layout\_layout\_below**

Positions the top edge of this view below the given anchor view ID. Accommodates top margin of this view and bottom margin of anchor view.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [layout\\_below](#)  
(/reference/android/R.attr.html#layout\_below).

Constant Value: 3 (0x00000003)

#### **public static final int RelativeLayout\_Layout\_layout\_centerHorizontal**

If true, centers this child horizontally within its parent.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*][*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_centerHorizontal](#)  
(/reference/android/R.attr.html#layout\_centerHorizontal).

Constant Value: 14 (0x0000000e)

#### **public static final int RelativeLayout\_Layout\_layout\_centerInParent**

If true, centers this child horizontally and vertically within its parent.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*][*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_centerInParent](#)  
(/reference/android/R.attr.html#layout\_centerInParent).

Constant Value: 13 (0x0000000d)

#### **public static final int RelativeLayout\_Layout\_layout\_centerVertical**

If true, centers this child vertically within its parent.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_centerVertical](#)  
(/reference/android/R.attr.html#layout\_centerVertical).

Constant Value: 15 (0x0000000f)

#### public static final int **RelativeLayout\_Layout\_layout\_toEndOf**

Positions the start edge of this view to the end of the given anchor view ID. Accommodates start margin of this view and end margin of anchor view.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [layout\\_toEndOf](#)  
(/reference/android/R.attr.html#layout\_toEndOf).

Constant Value: 18 (0x00000012)

#### public static final int **RelativeLayout\_Layout\_layout\_toLeftOf**

Positions the right edge of this view to the left of the given anchor view ID. Accommodates right margin of this view and left margin of anchor view.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [layout\\_toLeftOf](#)  
(/reference/android/R.attr.html#layout\_toLeftOf).

Constant Value: 0 (0x00000000)

#### public static final int **RelativeLayout\_Layout\_layout\_toRightOf**

Positions the left edge of this view to the right of the given anchor view ID. Accommodates left margin of this view and right margin of anchor view.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [layout\\_toRightOf](#)  
(/reference/android/R.attr.html#layout\_toRightOf).

Constant Value: 1 (0x00000001)

#### public static final int **RelativeLayout\_Layout\_layout\_toStartOf**

Positions the end edge of this view to the start of the given anchor view ID. Accommodates end margin of this view and start margin of anchor view.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [layout\\_toStartOf](#)  
(/reference/android/R.attr.html#layout\_toStartOf).

Constant Value: 17 (0x00000011)

#### public static final int **RelativeLayout\_gravity**

Specifies how an object should position its content, on both the X and Y axes, within its own bounds.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
top	0x30	Push object to the top of its container, not changing its size.
bottom	0x50	Push object to the bottom of its container, not changing its size.
left	0x03	Push object to the left of its container, not changing its size.
right	0x05	Push object to the right of its container, not changing its size.
center_vertical	0x10	Place object in the vertical center of its container, not changing its

		size.
fill_vertical	0x70	Grow the vertical size of the object if needed so it completely fills its container.
center_horizontal	0x01	Place object in the horizontal center of its container, not changing its size.
fill_horizontal	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
center	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
fill	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
clip_vertical	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
clip_horizontal	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a right gravity will clip the left edge, and neither will clip both edges.
start	0x00800003	Push object to the beginning of its container, not changing its size.
end	0x00800005	Push object to the end of its container, not changing its size.

This corresponds to the global attribute resource symbol [gravity](#)  
[\(/reference/android/R.attr.html#gravity\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **RelativeLayout\_ignoreGravity**

Indicates what view should not be affected by gravity.

Must be a reference to another resource, in the form "@[+][ package: ] type: name" or to a theme attribute in the form "?[ package: ][ type: ] name".

This corresponds to the global attribute resource symbol [ignoreGravity](#)  
[\(/reference/android/R.attr.html#ignoreGravity\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **RingtonePreference\_ringtoneType**

Which ringtone type(s) to show in the picker.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
ringtone	1	Ringtones.
notification	2	Notification sounds.
alarm	4	Alarm sounds.
all	7	All available ringtone sounds.

This corresponds to the global attribute resource symbol [ringtoneType](#)  
[\(/reference/android/R.attr.html#ringtoneType\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **RingtonePreference\_showDefault**

Whether to show an item for a default sound.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ][ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [showDefault](#)  
[\(/reference/android/R.attr.html#showDefault\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **RingtonePreference\_showSilent**

Whether to show an item for 'Silent'.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*:]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [showSilent](#)  
(/reference/android/R.attr.html#showSilent).

Constant Value: 2 (0x00000002)

#### public static final int **RotarySelector\_orientation**

Use "horizontal" or "vertical". The default is horizontal.

Must be one of the following constant values.

Constant	Value	Description
horizontal	0	Defines an horizontal widget.
vertical	1	Defines a vertical widget.

This corresponds to the global attribute resource symbol [orientation](#)  
(/reference/android/R.attr.html#orientation).

Constant Value: 0 (0x00000000)

#### public static final int **RotateAnimation\_fromDegrees**

This symbol is the offset where the [fromDegrees](#) (/reference/android/R.attr.html#fromDegrees) attribute's value can be found in the [RotateAnimation](#) (/reference/android/R.styleable.html#RotateAnimation) array.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*:]*name*") containing a value of this type.

Constant Value: 0 (0x00000000)

#### public static final int **RotateAnimation\_pivotX**

This symbol is the offset where the [pivotX](#) (/reference/android/R.attr.html#pivotX) attribute's value can be found in the [RotateAnimation](#) (/reference/android/R.styleable.html#RotateAnimation) array.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*:]*name*") containing a value of this type.

Constant Value: 2 (0x00000002)

#### public static final int **RotateAnimation\_pivotY**

This symbol is the offset where the [pivotY](#) (/reference/android/R.attr.html#pivotY) attribute's value can be found in the [RotateAnimation](#) (/reference/android/R.styleable.html#RotateAnimation) array.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*:]*name*") containing a value of this type.

Constant Value: 3 (0x00000003)

#### public static final int **RotateAnimation\_toDegrees**

This symbol is the offset where the [toDegrees](#) (/reference/android/R.attr.html#toDegrees) attribute's value can be found in the [RotateAnimation](#) (/reference/android/R.styleable.html#RotateAnimation) array.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*:]*name*") containing a value of this type.

Constant Value: 1 (0x00000001)

#### public static final int **RotateDrawable\_drawable**

Reference to a drawable resource to use for the frame. If not given, the drawable must be defined by the first child tag.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [drawable](#)  
[\(/reference/android/R.attr.html#drawable\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **RotateDrawable\_fromDegrees**

This symbol is the offset where the [fromDegrees](#) [\(/reference/android/R.attr.html#fromDegrees\)](#) attribute's value can be found in the [RotateDrawable](#) [\(/reference/android/R.styleable.html#RotateDrawable\)](#) array.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

Constant Value: 2 (0x00000002)

#### public static final int **RotateDrawable\_pivotX**

This symbol is the offset where the [pivotX](#) [\(/reference/android/R.attr.html#pivotX\)](#) attribute's value can be found in the [RotateDrawable](#) [\(/reference/android/R.styleable.html#RotateDrawable\)](#) array.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

Constant Value: 4 (0x00000004)

#### public static final int **RotateDrawable\_pivotY**

This symbol is the offset where the [pivotY](#) [\(/reference/android/R.attr.html#pivotY\)](#) attribute's value can be found in the [RotateDrawable](#) [\(/reference/android/R.styleable.html#RotateDrawable\)](#) array.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

Constant Value: 5 (0x00000005)

#### public static final int **RotateDrawable\_toDegrees**

This symbol is the offset where the [toDegrees](#) [\(/reference/android/R.attr.html#toDegrees\)](#) attribute's value can be found in the [RotateDrawable](#) [\(/reference/android/R.styleable.html#RotateDrawable\)](#) array.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

Constant Value: 3 (0x00000003)

#### public static final int **RotateDrawable\_visible**

Provides initial visibility state of the drawable; the default value is false. See [setVisible\(boolean, boolean\)](#) [\(/reference/android/graphics/drawable/Drawable.html#setVisible\(boolean, boolean\)\)](#).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [visible](#)  
(/reference/android/R.attr.html#visible).

Constant Value: 0 (0x00000000)

#### public static final int **ScaleAnimation\_fromXScale**

This symbol is the offset where the [fromXScale](#) (/reference/android/R.attr.html#fromXScale) attribute's value can be found in the [ScaleAnimation](#) (/reference/android/R.styleable.html#ScaleAnimation) array.

May be a floating point value, such as "1.2".

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 2 (0x00000002)

#### public static final int **ScaleAnimation\_fromYScale**

This symbol is the offset where the [fromYScale](#) (/reference/android/R.attr.html#fromYScale) attribute's value can be found in the [ScaleAnimation](#) (/reference/android/R.styleable.html#ScaleAnimation) array.

May be a floating point value, such as "1.2".

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 4 (0x00000004)

#### public static final int **ScaleAnimation\_pivotX**

This symbol is the offset where the [pivotX](#) (/reference/android/R.attr.html#pivotX) attribute's value can be found in the [ScaleAnimation](#) (/reference/android/R.styleable.html#ScaleAnimation) array.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 0 (0x00000000)

#### public static final int **ScaleAnimation\_pivotY**

This symbol is the offset where the [pivotY](#) (/reference/android/R.attr.html#pivotY) attribute's value can be found in the [ScaleAnimation](#) (/reference/android/R.styleable.html#ScaleAnimation) array.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[*package:*] *type:*  
*name*") or theme attribute (in the form "?[*package:*] [*type:*] *name*") containing a value of this type.

Constant Value: 1 (0x00000001)

#### public static final int **ScaleAnimation\_toXScale**

This symbol is the offset where the [toXScale](#) (*/reference/android/R.attr.html#toXScale*) attribute's value can be found in the [ScaleAnimation](#) (*/reference/android/R.styleable.html#ScaleAnimation*) array.

May be a floating point value, such as "1.2".

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[*package:*] *type:*  
*name*") or theme attribute (in the form "?[*package:*] [*type:*] *name*") containing a value of this type.

Constant Value: 3 (0x00000003)

#### public static final int **ScaleAnimation\_toYScale**

This symbol is the offset where the [toYScale](#) (*/reference/android/R.attr.html#toYScale*) attribute's value can be found in the [ScaleAnimation](#) (*/reference/android/R.styleable.html#ScaleAnimation*) array.

May be a floating point value, such as "1.2".

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[*package:*] *type:*  
*name*") or theme attribute (in the form "?[*package:*] [*type:*] *name*") containing a value of this type.

Constant Value: 5 (0x00000005)

#### public static final int **ScaleDrawable\_drawable**

Reference to a drawable resource to draw with the specified scale.

Must be a reference to another resource, in the form "@[+][*package:*] *type:*  
*name*" or to a theme attribute in the form "?[*package:*] [*type:*] *name*".

This corresponds to the global attribute resource symbol [drawable](#) (*/reference/android/R.attr.html#drawable*).

Constant Value: 0 (0x00000000)

#### public static final int **ScaleDrawable\_scaleGravity**

Specifies where the drawable is positioned after scaling. The default value is left.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
top	0x30	Push object to the top of its container, not changing its size.
bottom	0x50	Push object to the bottom of its container, not changing its size.
left	0x03	Push object to the left of its container, not changing its size.
right	0x05	Push object to the right of its container, not changing its size.
center_vertical	0x10	Place object in the vertical center of its container, not changing its size.
fill_vertical	0x70	Grow the vertical size of the object if needed so it completely fills its container.
center_horizontal	0x01	Place object in the horizontal center of its container, not changing its size.

<code>fill_horizontal</code>	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
<code>center</code>	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
<code>fill</code>	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
<code>clip_vertical</code>	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
<code>clip_horizontal</code>	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a right gravity will clip the left edge, and neither will clip both edges.
<code>start</code>	0x00800003	Push object to the beginning of its container, not changing its size.
<code>end</code>	0x00800005	Push object to the end of its container, not changing its size.

This corresponds to the global attribute resource symbol [scaleGravity](#)  
[\(/reference/android/R.attr.html#scaleGravity\)](#).

Constant Value: 3 (0x00000003)

#### **public static final int ScaleDrawable\_scaleHeight**

Scale height, expressed as a percentage of the drawable's bound. The value's format is XX%. For instance: 100%, 12.5%, etc.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [scaleHeight](#)  
[\(/reference/android/R.attr.html#scaleHeight\)](#).

Constant Value: 2 (0x00000002)

#### **public static final int ScaleDrawable\_scaleWidth**

Scale width, expressed as a percentage of the drawable's bound. The value's format is XX%. For instance: 100%, 12.5%, etc.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [scaleWidth](#)  
[\(/reference/android/R.attr.html#scaleWidth\)](#).

Constant Value: 1 (0x00000001)

#### **public static final int ScaleDrawable\_useIntrinsicSizeAsMinimum**

Use the drawable's intrinsic width and height as minimum size values. Useful if the target drawable is a 9-patch or otherwise should not be scaled down beyond a minimum size.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [useIntrinsicSizeAsMinimum](#)  
[\(/reference/android/R.attr.html#useIntrinsicSizeAsMinimum\)](#).

Constant Value: 4 (0x00000004)

#### **public static final int ScrollView\_fillViewport**

Defines whether the scrollview should stretch its content to fill the viewport.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [fillViewport](#)

[\(/reference/android/R.attr.html#fillViewport\).](#)

Constant Value: 0 (0x00000000)

### public static final int **SearchView.iconifiedByDefault**

The default state of the SearchView. If true, it will be iconified when not in use and expanded when clicked.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [iconifiedByDefault](#)

[\(/reference/android/R.attr.html#iconifiedByDefault\).](#)

Constant Value: 3 (0x00000003)

### public static final int **SearchView.imeOptions**

The IME options to set on the query text field.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
normal	0x00000000	There are no special semantics associated with this editor.
actionUnspecified	0x00000000	There is no specific action associated with this editor, let the editor come up with its own if it can. Corresponds to <a href="#">IME_NULL</a> .
actionNone	0x00000001	This editor has no action associated with it. Corresponds to <a href="#">IME_ACTION_NONE</a> .
actionGo	0x00000002	The action key performs a "go" operation to take the user to the target of the text they typed. Typically used, for example, when entering a URL. Corresponds to <a href="#">IME_ACTION_GO</a> .
actionSearch	0x00000003	The action key performs a "search" operation, taking the user to the results of searching for the text they have typed (in whatever context is appropriate). Corresponds to <a href="#">IME_ACTION_SEARCH</a> .
actionSend	0x00000004	The action key performs a "send" operation, delivering the text to its target. This is typically used when composing a message. Corresponds to <a href="#">IME_ACTION_SEND</a> .
actionNext	0x00000005	The action key performs a "next" operation, taking the user to the next field that will accept text. Corresponds to <a href="#">IME_ACTION_NEXT</a> .
actionDone	0x00000006	The action key performs a "done" operation, closing the soft input method. Corresponds to <a href="#">IME_ACTION_DONE</a> .
actionPrevious	0x00000007	The action key performs a "previous" operation, taking the user to the previous field that will accept text. Corresponds to <a href="#">IME_ACTION_PREVIOUS</a> .
flagNoFullscreen	0x2000000	Used to request that the IME never go into fullscreen mode. Applications need to be aware that the flag is not a guarantee, and not all IMEs will respect it. Corresponds to <a href="#">IME_FLAG_NO_FULLSCREEN</a> <a href="#">(/reference/android/view/inputmethod/EditorInfo.html#IME_FLAG_NO_FULLSCREEN)</a> .
flagNavigatePrevious	0x4000000	Like flagNavigateNext, but specifies there is something interesting that a backward navigation can focus on. If the user selects the IME's facility to backward navigate, this will show up in the application as an actionPrevious at <a href="#">InputConnection.performEditorAction(int)</a> . Corresponds to <a href="#">IME_FLAG_NO_FULLSCREEN</a> <a href="#">(/reference/android/view/inputmethod/EditorInfo.html#IME_FLAG_NO_FULLSCREEN)</a> .
flagNavigateNext	0x8000000	Used to specify that there is something interesting that a forward navigation can focus on. This is like using actionNext, except it allows the IME to be multiline (with an enter key) as well as provide forward navigation. Note that some IMEs may not be able to do this, especially when running on a small screen where there is little space. In that case it does not need to present a UI for this option. Like actionNext, if the user selects the IME's facility to forward navigate, this will show up in the application at <a href="#">InputConnection.performEditorAction(int)</a> . Corresponds to <a href="#">IME_FLAG_NAVIGATE_NEXT</a> <a href="#">(/reference/android/view/inputmethod/EditorInfo.html#IME_FLAG_NAVIGATE_NEXT)</a> .
flagNoExtractUi	0x10000000	Used to specify that the IME does not need to show its extracted text UI. For input methods that may be fullscreen, often when in landscape mode, this allows them to be smaller and let part of the application be shown behind. Though there will likely be limited access to the application available from the user, it can make the experience of a (mostly) fullscreen IME less jarring. Note that when this flag is specified the IME may <i>not</i> be set up to be able to display text, so it should only be used in situations where this is not needed. Corresponds to <a href="#">IME_FLAG_NO_EXTRACT_UI</a> <a href="#">(/reference/android/view/inputmethod/EditorInfo.html#IME_FLAG_NO_EXTRACT_UI)</a> .

Used in conjunction with a custom action, this indicates that the action should not be

available as an accessory button when the input method is full-screen. Note that by setting this flag, there can be cases where the action is simply never available to the user. Setting this flagNoAccessoryAction 0x20000000 generally means that you think showing text being edited is more important than the action you have supplied.

Corresponds to [IME\\_FLAG\\_NO\\_ACCESSORY\\_ACTION](#)  
[\(/reference/android/view/inputmethod/EditorInfo.html#IME\\_FLAG\\_NO\\_ACCESSORY\\_ACTION\)](#).

flagNoEnterAction 0x40000000 Used in conjunction with a custom action, this indicates that the action should not be available in-line as a replacement for the "enter" key. Typically this is because the action has such a significant impact or is not recoverable enough that accidentally hitting it should be avoided, such as sending a message. Note that [TextView](#) will automatically set this flag for you on multi-line text views.

Corresponds to [IME\\_FLAG\\_NO\\_ENTER\\_ACTION](#)  
[\(/reference/android/view/inputmethod/EditorInfo.html#IME\\_FLAG\\_NO\\_ENTER\\_ACTION\)](#).

flagForceAscii 0x80000000 Used to request that the IME should be capable of inputting ASCII characters. The intention of this flag is to ensure that the user can type Roman alphabet characters in a [TextView](#) used for, typically, account ID or password input. It is expected that IMEs normally are able to input ASCII even without being told so (such IMEs already respect this flag in a sense), but there could be some cases they aren't when, for instance, only non-ASCII input languages like Arabic, Greek, Hebrew, Russian are enabled in the IME. Applications need to be aware that the flag is not a guarantee, and not all IMEs will respect it. However, it is strongly recommended for IME authors to respect this flag especially when their IME could end up with a state that has only non-ASCII input languages enabled.

Corresponds to [IME\\_FLAG\\_FORCE\\_ASCII](#)  
[\(/reference/android/view/inputmethod/EditorInfo.html#IME\\_FLAG\\_FORCE\\_ASCII\)](#).

This corresponds to the global attribute resource symbol [imeOptions](#)  
[\(/reference/android/R.attr.html#imeOptions\)](#).

Constant Value: 2 (0x00000002)

## public static final int SearchView\_inputType

The input type to set on the query text field.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
none	0x00000000	There is no content type. The text is not editable.
text	0x00000001	Just plain old text. Corresponds to <a href="#">TYPE_CLASS_TEXT</a>   <a href="#">TYPE_TEXT_VARIATION_NORMAL</a> .
textCapCharacters	0x00001001	Can be combined with <a href="#">text</a> and its variations to request capitalization of all characters. Corresponds to <a href="#">TYPE_TEXT_FLAG_CAP_CHARACTERS</a> .
textCapWords	0x00002001	Can be combined with <a href="#">text</a> and its variations to request capitalization of the first character of every word. Corresponds to <a href="#">TYPE_TEXT_FLAG_CAP_WORDS</a> .
textCapSentences	0x00004001	Can be combined with <a href="#">text</a> and its variations to request capitalization of the first character of every sentence. Corresponds to <a href="#">TYPE_TEXT_FLAG_CAP_SENTENCES</a> .
textAutoCorrect	0x00008001	Can be combined with <a href="#">text</a> and its variations to request auto-correction of text being input. Corresponds to <a href="#">TYPE_TEXT_FLAG_AUTO_CORRECT</a> .
textAutoComplete	0x00010001	Can be combined with <a href="#">text</a> and its variations to specify that this field will be doing its own auto-completion and talking with the input method appropriately. Corresponds to <a href="#">TYPE_TEXT_FLAG_AUTO_COMPLETE</a> .
textMultiLine	0x00020001	Can be combined with <a href="#">text</a> and its variations to allow multiple lines of text in the field. If this flag is not set, the text field will be constrained to a single line. Corresponds to <a href="#">TYPE_TEXT_FLAG_MULTI_LINE</a> .
textImeMultiLine	0x00040001	Can be combined with <a href="#">text</a> and its variations to indicate that though the regular text view should not be multiple lines, the IME should provide multiple lines if it can. Corresponds to <a href="#">TYPE_TEXT_FLAG_IME_MULTI_LINE</a> .
textNoSuggestions	0x00080001	Can be combined with <a href="#">text</a> and its variations to indicate that the IME should not show any dictionary-based word suggestions. Corresponds to <a href="#">TYPE_TEXT_FLAG_NO_SUGGESTIONS</a> .
textUri	0x00000011	Text that will be used as a URI. Corresponds to <a href="#">TYPE_CLASS_TEXT</a>   <a href="#">TYPE_TEXT_VARIATION_URI</a> . Text that will be used as an e-mail address. Corresponds to

textEmailAddress	0x00000021	<u>TYPE_CLASS_TEXT   TYPE_TEXT_VARIATION_EMAIL_ADDRESS</u> .
		Text that is being supplied as the subject of an e-mail.
textEmailSubject	0x00000031	Corresponds to <u>TYPE_CLASS_TEXT   TYPE_TEXT_VARIATION_EMAIL_SUBJECT</u> .
textShortMessage	0x00000041	Text that is the content of a short message. Corresponds to <u>TYPE_CLASS_TEXT   TYPE_TEXT_VARIATION_SHORT_MESSAGE</u> .
textLongMessage	0x00000051	Text that is the content of a long message. Corresponds to <u>TYPE_CLASS_TEXT   TYPE_TEXT_VARIATION_LONG_MESSAGE</u> .
textPersonName	0x00000061	Text that is the name of a person. Corresponds to <u>TYPE_CLASS_TEXT   TYPE_TEXT_VARIATION_PERSON_NAME</u> .
		Text that is being supplied as a postal mailing address.
textPostalAddress	0x00000071	Corresponds to <u>TYPE_CLASS_TEXT   TYPE_TEXT_VARIATION_POSTAL_ADDRESS</u> .
textPassword	0x00000081	Text that is a password. Corresponds to <u>TYPE_CLASS_TEXT   TYPE_TEXT_VARIATION_PASSWORD</u> .
		Text that is a password that should be visible. Corresponds to <u>TYPE_CLASS_TEXT   TYPE_TEXT_VARIATION_VISIBLE_PASSWORD</u> .
textVisiblePassword	0x00000091	
textWebEditText	0x000000a1	Text that is being supplied as text in a web form. Corresponds to <u>TYPE_CLASS_TEXT   TYPE_TEXT_VARIATION_WEB_EDIT_TEXT</u> .
textFilter	0x000000b1	Text that is filtering some other data. Corresponds to <u>TYPE_CLASS_TEXT   TYPE_TEXT_VARIATION_FILTER</u> .
textPhonetic	0x000000c1	Text that is for phonetic pronunciation, such as a phonetic name field in a contact entry. Corresponds to <u>TYPE_CLASS_TEXT   TYPE_TEXT_VARIATION_PHONETIC</u> .
		Text that will be used as an e-mail address on a web form.
textWebEmailAddress	0x000000d1	Corresponds to <u>TYPE_CLASS_TEXT   TYPE_TEXT_VARIATION_WEB_EMAIL_ADDRESS</u> .
textWebPassword	0x000000e1	Text that will be used as a password on a web form. Corresponds to <u>TYPE_CLASS_TEXT   TYPE_TEXT_VARIATION_WEB_PASSWORD</u> .
number	0x00000002	A numeric only field. Corresponds to <u>TYPE_CLASS_NUMBER   TYPE_NUMBER_VARIATION_NORMAL</u> .
numberSigned	0x00001002	Can be combined with <i>number</i> and its other options to allow a signed number. Corresponds to <u>TYPE_CLASS_NUMBER   TYPE_NUMBER_FLAG_SIGNED</u> .
numberDecimal	0x00002002	Can be combined with <i>number</i> and its other options to allow a decimal (fractional) number. Corresponds to <u>TYPE_CLASS_NUMBER   TYPE_NUMBER_FLAG_DECIMAL</u> .
numberPassword	0x00000012	A numeric password field. Corresponds to <u>TYPE_CLASS_NUMBER   TYPE_NUMBER_VARIATION_PASSWORD</u> .
phone	0x00000003	For entering a phone number. Corresponds to <u>TYPE_CLASS_PHONE</u> .
datetime	0x00000004	For entering a date and time. Corresponds to <u>TYPE_CLASS_DATETIME   TYPE_DATETIME_VARIATION_NORMAL</u> .
date	0x00000014	For entering a date. Corresponds to <u>TYPE_CLASS_DATETIME   TYPE_DATETIME_VARIATION_DATE</u> .
time	0x00000024	For entering a time. Corresponds to <u>TYPE_CLASS_DATETIME   TYPE_DATETIME_VARIATION_TIME</u> .

This corresponds to the global attribute resource symbol [inputType](#)  
[\(/reference/android/R.attr.html#inputType\)](#).

Constant Value: 1 (0x00000001)

#### public static final int SearchView\_maxWidth

An optional maximum width of the SearchView.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[*package:*] *type:name*") or theme attribute (in the form "?[*package:*] [*type:*] *name*") containing a value of this type.

This corresponds to the global attribute resource symbol [maxWidth](#)  
[\(/reference/android/R.attr.html#maxWidth\)](#).

Constant Value: 0 (0x00000000)

## `public static final int SearchView_queryHint`

An optional query hint string to be displayed in the empty query field.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [queryHint](#)  
[\(/reference/android/R.attr.html#queryHint\)](#).

Constant Value: 4 (0x00000004)

## `public static final int SearchableActionKey_keycode`

This attribute denotes the action key you wish to respond to. Note that not all action keys are actually supported using this mechanism, as many of them are used for typing, navigation, or system functions.

This will be added to the [ACTION\\_SEARCH](#) ([\(/reference/android/content/Intent.html#ACTION\\_SEARCH\)](#)) intent that is passed to your searchable activity. To examine the key code, use  
[getIntExtra\(SearchManager.ACTION\\_KEY\)](#)  
[\(/reference/android/content/Intent.html#getIntExtra\(java.lang.String, int\)\)](#).

Note, in addition to the keycode, you must also provide one or more of the action specifier attributes.  
*Required attribute.*

Must be one of the following constant values.

Constant	Value Description
KEYCODE_UNKNOWN	0
KEYCODE_SOFT_LEFT	1
KEYCODE_SOFT_RIGHT	2
KEYCODE_HOME	3
KEYCODE_BACK	4
KEYCODE_CALL	5
KEYCODE_ENDCALL	6
KEYCODE_0	7
KEYCODE_1	8
KEYCODE_2	9
KEYCODE_3	10
KEYCODE_4	11
KEYCODE_5	12
KEYCODE_6	13
KEYCODE_7	14
KEYCODE_8	15
KEYCODE_9	16
KEYCODE_STAR	17
KEYCODE_POUND	18
KEYCODE_DPAD_UP	19
KEYCODE_DPAD_DOWN	20
KEYCODE_DPAD_LEFT	21
KEYCODE_DPAD_RIGHT	22
KEYCODE_DPAD_CENTER	23
KEYCODE_VOLUME_UP	24
KEYCODE_VOLUME_DOWN	25
KEYCODE_POWER	26
KEYCODE_CAMERA	27
KEYCODE_CLEAR	28
KEYCODE_A	29
KEYCODE_B	30
KEYCODE_C	31
KEYCODE_D	32
KEYCODE_E	33
KEYCODE_F	34
KEYCODE_G	35
KEYCODE_H	36
KEYCODE_I	37

KEYCODE_J	38
KEYCODE_K	39
KEYCODE_L	40
KEYCODE_M	41
KEYCODE_N	42
KEYCODE_O	43
KEYCODE_P	44
KEYCODE_Q	45
KEYCODE_R	46
KEYCODE_S	47
KEYCODE_T	48
KEYCODE_U	49
KEYCODE_V	50
KEYCODE_W	51
KEYCODE_X	52
KEYCODE_Y	53
KEYCODE_Z	54
KEYCODE_COMMA	55
KEYCODE_PERIOD	56
KEYCODE_ALT_LEFT	57
KEYCODE_ALT_RIGHT	58
KEYCODE_SHIFT_LEFT	59
KEYCODE_SHIFT_RIGHT	60
KEYCODE_TAB	61
KEYCODE_SPACE	62
KEYCODE_SYM	63
KEYCODE_EXPLORER	64
KEYCODE_ENVELOPE	65
KEYCODE_ENTER	66
KEYCODE_DEL	67
KEYCODE_GRAVE	68
KEYCODE_MINUS	69
KEYCODE_EQUALS	70
KEYCODE_LEFT_BRACKET	71
KEYCODE_RIGHT_BRACKET	72
KEYCODE_BACKSLASH	73
KEYCODE_SEMICOLON	74
KEYCODE_APOSTROPHE	75
KEYCODE_SLASH	76
KEYCODE_AT	77
KEYCODE_NUM	78
KEYCODE_HEADSETHOOK	79
KEYCODE_FOCUS	80
KEYCODE_PLUS	81
KEYCODE_MENU	82
KEYCODE_NOTIFICATION	83
KEYCODE_SEARCH	84
KEYCODE_MEDIA_PLAY_PAUSE	85
KEYCODE_MEDIA_STOP	86
KEYCODE_MEDIA_NEXT	87
KEYCODE_MEDIA_PREVIOUS	88
KEYCODE_MEDIA_REWIND	89
KEYCODE_MEDIA_FAST_FORWARD	90
KEYCODE_MUTE	91
KEYCODE_PAGE_UP	92
KEYCODE_PAGE_DOWN	93
KEYCODE_PICTSYMBOLS	94
KEYCODE_SWITCH_CHARSET	95
KEYCODE_BUTTON_A	96

KEYCODE_BUTTON_B	97
KEYCODE_BUTTON_C	98
KEYCODE_BUTTON_X	99
KEYCODE_BUTTON_Y	100
KEYCODE_BUTTON_Z	101
KEYCODE_BUTTON_L1	102
KEYCODE_BUTTON_R1	103
KEYCODE_BUTTON_L2	104
KEYCODE_BUTTON_R2	105
KEYCODE_BUTTON_THUMBL	106
KEYCODE_BUTTON_THUMBR	107
KEYCODE_BUTTON_START	108
KEYCODE_BUTTON_SELECT	109
KEYCODE_BUTTON_MODE	110
KEYCODE_ESCAPE	111
KEYCODE_FORWARD_DEL	112
KEYCODE_CTRL_LEFT	113
KEYCODE_CTRL_RIGHT	114
KEYCODE_CAPS_LOCK	115
KEYCODE_SCROLL_LOCK	116
KEYCODE_META_LEFT	117
KEYCODE_META_RIGHT	118
KEYCODE_FUNCTION	119
KEYCODE_SYSRQ	120
KEYCODE_BREAK	121
KEYCODE_MOVE_HOME	122
KEYCODE_MOVE_END	123
KEYCODE_INSERT	124
KEYCODE_FORWARD	125
KEYCODE_MEDIA_PLAY	126
KEYCODE_MEDIA_PAUSE	127
KEYCODE_MEDIA_CLOSE	128
KEYCODE_MEDIA_EJECT	129
KEYCODE_MEDIA_RECORD	130
KEYCODE_F1	131
KEYCODE_F2	132
KEYCODE_F3	133
KEYCODE_F4	134
KEYCODE_F5	135
KEYCODE_F6	136
KEYCODE_F7	137
KEYCODE_F8	138
KEYCODE_F9	139
KEYCODE_F10	140
KEYCODE_F11	141
KEYCODE_F12	142
KEYCODE_NUM_LOCK	143
KEYCODE_NUMPAD_0	144
KEYCODE_NUMPAD_1	145
KEYCODE_NUMPAD_2	146
KEYCODE_NUMPAD_3	147
KEYCODE_NUMPAD_4	148
KEYCODE_NUMPAD_5	149
KEYCODE_NUMPAD_6	150
KEYCODE_NUMPAD_7	151
KEYCODE_NUMPAD_8	152
KEYCODE_NUMPAD_9	153
KEYCODE_NUMPAD_DIVIDE	154
KEYCODE_NUMPAD_MULTIPLY	155
KEYCODE_NUMPAD_SUBTRACT	156

KEYCODE_NUMPAD_ADD	157
KEYCODE_NUMPAD_DOT	158
KEYCODE_NUMPAD_COMMMA	159
KEYCODE_NUMPAD_ENTER	160
KEYCODE_NUMPAD_EQUALS	161
KEYCODE_NUMPAD_LEFT_PAREN	162
KEYCODE_NUMPAD_RIGHT_PAREN	163
KEYCODE_VOLUME_MUTE	164
KEYCODE_INFO	165
KEYCODE_CHANNEL_UP	166
KEYCODE_CHANNEL_DOWN	167
KEYCODE_ZOOM_IN	168
KEYCODE_ZOOM_OUT	169
KEYCODE_TV	170
KEYCODE_WINDOW	171
KEYCODE_GUIDE	172
KEYCODE_DVR	173
KEYCODE_BOOKMARK	174
KEYCODE_CAPTIONS	175
KEYCODE_SETTINGS	176
KEYCODE_TV_POWER	177
KEYCODE_TV_INPUT	178
KEYCODE_STB_POWER	179
KEYCODE_STB_INPUT	180
KEYCODE_AVR_POWER	181
KEYCODE_AVR_INPUT	182
KEYCODE_PROG_GRED	183
KEYCODE_PROG_GREEN	184
KEYCODE_PROG_YELLOW	185
KEYCODE_PROG_BLUE	186
KEYCODE_APP_SWITCH	187
KEYCODE_BUTTON_1	188
KEYCODE_BUTTON_2	189
KEYCODE_BUTTON_3	190
KEYCODE_BUTTON_4	191
KEYCODE_BUTTON_5	192
KEYCODE_BUTTON_6	193
KEYCODE_BUTTON_7	194
KEYCODE_BUTTON_8	195
KEYCODE_BUTTON_9	196
KEYCODE_BUTTON_10	197
KEYCODE_BUTTON_11	198
KEYCODE_BUTTON_12	199
KEYCODE_BUTTON_13	200
KEYCODE_BUTTON_14	201
KEYCODE_BUTTON_15	202
KEYCODE_BUTTON_16	203
KEYCODE_LANGUAGE_SWITCH	204
KEYCODE_MANNER_MODE	205
KEYCODE_3D_MODE	206
KEYCODE_CONTACTS	207
KEYCODE_CALENDAR	208
KEYCODE_MUSIC	209
KEYCODE_CALCULATOR	210
KEYCODE_ZENKAKU_HANKAKU	211
KEYCODE_EISU	212
KEYCODE_MUHENKAN	213
KEYCODE_HENKAN	214
KEYCODE_KATAKANA_HIRAGANA	215

KEYCODE_YEN	216
KEYCODE_RO	217
KEYCODE_KANA	218

This corresponds to the global attribute resource symbol [keycode](#)  
[\(/reference/android/R.attr.html#keycode\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **SearchableActionKey\_queryActionMsg**

If you wish to handle an action key during normal search query entry, you must define an action string here. This will be added to the [ACTION\\_SEARCH](#)  
[\(/reference/android/content/Intent.html#ACTION\\_SEARCH\)](#) intent that is passed to your searchable activity. To examine the string, use [getStringExtra\(SearchManager.ACTION\\_MSG\)](#)  
[\(/reference/android/content/Intent.html#getStringExtra\(java.lang.String\)\)](#). *Optional attribute.*

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [queryActionMsg](#)  
[\(/reference/android/R.attr.html#queryActionMsg\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **SearchableActionKey\_suggestActionMsg**

If you wish to handle an action key while a suggestion is being displayed *and selected*, there are two ways to handle this. If *all* of your suggestions can handle the action key, you can simply define the action message using this attribute. This will be added to the [ACTION\\_SEARCH](#)  
[\(/reference/android/content/Intent.html#ACTION\\_SEARCH\)](#) intent that is passed to your searchable activity. To examine the string, use [getStringExtra\(SearchManager.ACTION\\_MSG\)](#)  
[\(/reference/android/content/Intent.html#getStringExtra\(java.lang.String\)\)](#). *Optional attribute.*

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [suggestActionMsg](#)  
[\(/reference/android/R.attr.html#suggestActionMsg\)](#).

Constant Value: 2 (0x00000002)

#### public static final int **SearchableActionKey\_suggestActionMsgColumn**

If you wish to handle an action key while a suggestion is being displayed *and selected*, but you do not wish to enable this action key for every suggestion, then you can use this attribute to control it on a suggestion-by-suggestion basis. First, you must define a column (and name it here) where your suggestions will include the action string. Then, in your content provider, you must provide this column, and when desired, provide data in this column. The search manager will look at your suggestion cursor, using the string provided here in order to select a column, and will use that to select a string from the cursor. That string will be added to the [ACTION\\_SEARCH](#)  
[\(/reference/android/content/Intent.html#ACTION\\_SEARCH\)](#) intent that is passed to your searchable activity. To examine the string, use [getStringExtra\(SearchManager.ACTION\\_MSG\)](#)  
[\(/reference/android/content/Intent.html#getStringExtra\(java.lang.String\)\)](#). *If the data does not exist for the selection suggestion, the action key will be ignored.* *Optional attribute.*

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [suggestActionMsgColumn](#)  
[\(/reference/android/R.attr.html#suggestActionMsgColumn\)](#).

Constant Value: 3 (0x00000003)

#### public static final int **Searchable\_autoUrlDetect**

If provided and true, URLs entered in the search dialog while searching within this activity would be detected and treated as URLs (show a 'go' button in the keyboard and invoke the browser directly when user launches the URL instead of passing the URL to the activity). If set to false any URLs entered are treated as normal query text. The default value is false. *Optional attribute..*

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [autoUrlDetect](#)  
(/reference/android/R.attr.html#autoUrlDetect).

Constant Value: 21 (0x00000015)

#### public static final int Searchable\_hint

If supplied, this string will be displayed as a hint to the user. *Optional attribute*.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [hint](#) (/reference/android/R.attr.html#hint).

Constant Value: 2 (0x00000002)

#### public static final int Searchable\_icon

This is deprecated.

The default application icon is now always used, so this attribute is obsolete.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [icon](#) (/reference/android/R.attr.html#icon).

Constant Value: 1 (0x00000001)

#### public static final int Searchable imeOptions

Additional features you can enable in an IME associated with an editor to improve the integration with your application. The constants here correspond to those defined by [imeOptions](#)  
(/reference/android/view/inputmethod/EditorInfo.html#imeOptions).

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
normal	0x00000000	There are no special semantics associated with this editor.
actionUnspecified	0x00000000	There is no specific action associated with this editor, let the editor come up with its own if it can. Corresponds to <a href="#">IME_NULL</a> .
actionNone	0x00000001	This editor has no action associated with it. Corresponds to <a href="#">IME_ACTION_NONE</a> .
actionGo	0x00000002	The action key performs a "go" operation to take the user to the target of the text they typed. Typically used, for example, when entering a URL. Corresponds to <a href="#">IME_ACTION_GO</a> .
actionSearch	0x00000003	The action key performs a "search" operation, taking the user to the results of searching for the text they have typed (in whatever context is appropriate). Corresponds to <a href="#">IME_ACTION_SEARCH</a> .
actionSend	0x00000004	The action key performs a "send" operation, delivering the text to its target. This is typically used when composing a message. Corresponds to <a href="#">IME_ACTION_SEND</a> .
actionNext	0x00000005	The action key performs a "next" operation, taking the user to the next field that will accept text. Corresponds to <a href="#">IME_ACTION_NEXT</a> .
actionDone	0x00000006	The action key performs a "done" operation, closing the soft input method. Corresponds to <a href="#">IME_ACTION_DONE</a> .
actionPrevious	0x00000007	The action key performs a "previous" operation, taking the user to the previous field that will accept text. Corresponds to <a href="#">IME_ACTION_PREVIOUS</a> .
flagNoFullscreen	0x2000000	Used to request that the IME never go into fullscreen mode. Applications need to be aware that the flag is not a guarantee, and not all IMEs will respect it.  Corresponds to <a href="#">IME_FLAG_NO_FULLSCREEN</a> (/reference/android/view/inputmethod/EditorInfo.html#IME_FLAG_NO_FULLSCREEN).
flagNavigateNext	0x4000000	Like flagNavigateNext, but specifies there is something interesting that a backward navigation can focus on. If the user selects the IME's facility to backward navigate, this will show up in the application as an actionPrevious at <a href="#">InputConnection.performEditorAction(int)</a> .  Corresponds to <a href="#">IME_FLAG_NO_FULLSCREEN</a> (/reference/android/view/inputmethod/EditorInfo.html#IME_FLAG_NO_FULLSCREEN).
flagNavigatePrevious	0x4000000	Used to specify that there is something interesting that a forward navigation can focus on. This is like using actionNext, except allows the IME to be multiline (with an enter key) as well as provide forward navigation. Note that some IMEs may not be able to do this, especially

flagNavigateNext	0x8000000	when running on a small screen where there is little space. In that case it does not need to present a UI for this option. Like actionNext, if the user selects the IME's facility to forward navigate, this will show up in the application at <u><a href="#">InputConnection.performEditorAction(int)</a></u> . Corresponds to <u><a href="#">IME_FLAG_NAVIGATE_NEXT</a></u> <u><a href="#">(/reference/android/view/inputmethod/EditorInfo.html#IME_FLAG_NAVIGATE_NEXT)</a></u> .
flagNoExtractUi	0x10000000	Used to specify that the IME does not need to show its extracted text UI. For input methods that may be fullscreen, often when in landscape mode, this allows them to be smaller and let part of the application be shown behind. Though there will likely be limited access to the application available from the user, it can make the experience of a (mostly) fullscreen IME less jarring. Note that when this flag is specified the IME may <i>not</i> be set up to be able to display text, so it should only be used in situations where this is not needed. Corresponds to <u><a href="#">IME_FLAG_NO_EXTRACT_UI</a></u> <u><a href="#">(/reference/android/view/inputmethod/EditorInfo.html#IME_FLAG_NO_EXTRACT_UI)</a></u> .
flagNoAccessoryAction	0x20000000	Used in conjunction with a custom action, this indicates that the action should not be available as an accessory button when the input method is full-screen. Note that by setting this flag, there can be cases where the action is simply never available to the user. Setting this generally means that you think showing text being edited is more important than the action you have supplied. Corresponds to <u><a href="#">IME_FLAG_NO_ACCESSORY_ACTION</a></u> <u><a href="#">(/reference/android/view/inputmethod/EditorInfo.html#IME_FLAG_NO_ACCESSORY_ACTION)</a></u> .
flagNoEnterAction	0x40000000	Used in conjunction with a custom action, this indicates that the action should not be available in-line as a replacement for the "enter" key. Typically this is because the action has such a significant impact or is not recoverable enough that accidentally hitting it should be avoided, such as sending a message. Note that <u><a href="#">TextView</a></u> will automatically set this flag for you on multi-line text views. Corresponds to <u><a href="#">IME_FLAG_NO_ENTER_ACTION</a></u> <u><a href="#">(/reference/android/view/inputmethod/EditorInfo.html#IME_FLAG_NO_ENTER_ACTION)</a></u> .
flagForceAscii	0x80000000	Used to request that the IME should be capable of inputting ASCII characters. The intention of this flag is to ensure that the user can type Roman alphabet characters in a <u><a href="#">TextView</a></u> used for, typically, account ID or password input. It is expected that IMEs normally are able to input ASCII even without being told so (such IMEs already respect this flag in a sense), but there could be some cases they aren't when, for instance, only non-ASCII input languages like Arabic, Greek, Hebrew, Russian are enabled in the IME. Applications need to be aware that the flag is not a guarantee, and not all IMEs will respect it. However, it is strongly recommended for IME authors to respect this flag especially when their IME could end up with a state that has only non-ASCII input languages enabled. Corresponds to <u><a href="#">IME_FLAG_FORCE_ASCII</a></u> <u><a href="#">(/reference/android/view/inputmethod/EditorInfo.html#IME_FLAG_FORCE_ASCII)</a></u> .

This corresponds to the global attribute resource symbol [imeOptions](#)  
[\(/reference/android/R.attr.html#imeOptions\)](#).

Constant Value: 16 (0x00000010)

#### public static final int Searchable\_includeInGlobalSearch

If provided and true, this searchable activity will be included in any global lists of search targets. The default value is false. *Optional attribute..*

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [includeInGlobalSearch](#)  
[\(/reference/android/R.attr.html#includeInGlobalSearch\)](#).

Constant Value: 18 (0x00000012)

#### public static final int Searchable\_inputType

The type of data being placed in a text field, used to help an input method decide how to let the user enter text. The constants here correspond to those defined by [InputType](#)  
[\(/reference/android/text/InputType.html\)](#). Generally you can select a single value, though some can be combined together as indicated. Setting this attribute to anything besides *none* also implies that the text is editable.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
----------	-------	-------------

none	0x00000000	There is no content type. The text is not editable.
text	0x00000001	Just plain old text. Corresponds to <u>TYPE_CLASS_TEXT</u>   <u>TYPE_TEXT_VARIATION_NORMAL</u> .
textCapCharacters	0x00001001	Can be combined with <i>text</i> and its variations to request capitalization of all characters. Corresponds to <u>TYPE_TEXT_FLAG_CAP_CHARACTERS</u> .
textCapWords	0x00002001	Can be combined with <i>text</i> and its variations to request capitalization of the first character of every word. Corresponds to <u>TYPE_TEXT_FLAG_CAP_WORDS</u> .
textCapSentences	0x00004001	Can be combined with <i>text</i> and its variations to request capitalization of the first character of every sentence. Corresponds to <u>TYPE_TEXT_FLAG_CAP_SENTENCES</u> .
textAutoCorrect	0x00008001	Can be combined with <i>text</i> and its variations to request auto-correction of text being input. Corresponds to <u>TYPE_TEXT_FLAG_AUTO_CORRECT</u> .
textAutoComplete	0x00010001	Can be combined with <i>text</i> and its variations to specify that this field will be doing its own auto-completion and talking with the input method appropriately. Corresponds to <u>TYPE_TEXT_FLAG_AUTO_COMPLETE</u> .
textMultiLine	0x00020001	Can be combined with <i>text</i> and its variations to allow multiple lines of text in the field. If this flag is not set, the text field will be constrained to a single line. Corresponds to <u>TYPE_TEXT_FLAG_MULTI_LINE</u> .
textImeMultiLine	0x00040001	Can be combined with <i>text</i> and its variations to indicate that though the regular text view should not be multiple lines, the IME should provide multiple lines if it can. Corresponds to <u>TYPE_TEXT_FLAG_IME_MULTI_LINE</u> .
textNoSuggestions	0x00080001	Can be combined with <i>text</i> and its variations to indicate that the IME should not show any dictionary-based word suggestions. Corresponds to <u>TYPE_TEXT_FLAG_NO_SUGGESTIONS</u> .
textUri	0x00000011	Text that will be used as a URI. Corresponds to <u>TYPE_CLASS_TEXT</u>   <u>TYPE_TEXT_VARIATION_URI</u> .
textEmailAddress	0x00000021	Text that will be used as an e-mail address. Corresponds to <u>TYPE_CLASS_TEXT</u>   <u>TYPE_TEXT_VARIATION_EMAIL_ADDRESS</u> .
textEmailSubject	0x00000031	Text that is being supplied as the subject of an e-mail. Corresponds to <u>TYPE_CLASS_TEXT</u>   <u>TYPE_TEXT_VARIATION_EMAIL_SUBJECT</u> .
textShortMessage	0x00000041	Text that is the content of a short message. Corresponds to <u>TYPE_CLASS_TEXT</u>   <u>TYPE_TEXT_VARIATION_SHORT_MESSAGE</u> .
textLongMessage	0x00000051	Text that is the content of a long message. Corresponds to <u>TYPE_CLASS_TEXT</u>   <u>TYPE_TEXT_VARIATION_LONG_MESSAGE</u> .
textPersonName	0x00000061	Text that is the name of a person. Corresponds to <u>TYPE_CLASS_TEXT</u>   <u>TYPE_TEXT_VARIATION_PERSON_NAME</u> .
textPostalAddress	0x00000071	Text that is being supplied as a postal mailing address. Corresponds to <u>TYPE_CLASS_TEXT</u>   <u>TYPE_TEXT_VARIATION_POSTAL_ADDRESS</u> .
textPassword	0x00000081	Text that is a password. Corresponds to <u>TYPE_CLASS_TEXT</u>   <u>TYPE_TEXT_VARIATION_PASSWORD</u> .
textVisiblePassword	0x00000091	Text that is a password that should be visible. Corresponds to <u>TYPE_CLASS_TEXT</u>   <u>TYPE_TEXT_VARIATION_VISIBLE_PASSWORD</u> .
textWebEditText	0x000000a1	Text that is being supplied as text in a web form. Corresponds to <u>TYPE_CLASS_TEXT</u>   <u>TYPE_TEXT_VARIATION_WEB_EDIT_TEXT</u> .
textFilter	0x000000b1	Text that is filtering some other data. Corresponds to <u>TYPE_CLASS_TEXT</u>   <u>TYPE_TEXT_VARIATION_FILTER</u> .
textPhonetic	0x000000c1	Text that is for phonetic pronunciation, such as a phonetic name field in a contact entry. Corresponds to <u>TYPE_CLASS_TEXT</u>   <u>TYPE_TEXT_VARIATION_PHONETIC</u> .
textWebEmailAddress	0x000000d1	Text that will be used as an e-mail address on a web form. Corresponds to <u>TYPE_CLASS_TEXT</u>   <u>TYPE_TEXT_VARIATION_WEB_EMAIL_ADDRESS</u> .
textWebPassword	0x000000e1	Text that will be used as a password on a web form. Corresponds to <u>TYPE_CLASS_TEXT</u>   <u>TYPE_TEXT_VARIATION_WEB_PASSWORD</u> .
number	0x00000002	A numeric only field. Corresponds to <u>TYPE_CLASS_NUMBER</u>   <u>TYPE_NUMBER_VARIATION_NORMAL</u> .
numberSigned	0x00001002	Can be combined with <i>number</i> and its other options to allow a signed number. Corresponds to <u>TYPE_CLASS_NUMBER</u>   <u>TYPE_NUMBER_FLAG_SIGNED</u> .

numberDecimal	0x00002002	Can be combined with <i>number</i> and its other options to allow a decimal (fractional) number. Corresponds to <a href="#">TYPE_CLASS_NUMBER   TYPE_NUMBER_FLAG_DECIMAL</a> .
numberPassword	0x00000012	A numeric password field. Corresponds to <a href="#">TYPE_CLASS_NUMBER   TYPE_NUMBER_VARIATION_PASSWORD</a> .
phone	0x00000003	For entering a phone number. Corresponds to <a href="#">TYPE_CLASS_PHONE</a> .
datetime	0x00000004	For entering a date and time. Corresponds to <a href="#">TYPE_CLASS_DATETIME   TYPE_DATETIME_VARIATION_NORMAL</a> .
date	0x00000014	For entering a date. Corresponds to <a href="#">TYPE_CLASS_DATETIME   TYPE_DATETIME_VARIATION_DATE</a> .
time	0x00000024	For entering a time. Corresponds to <a href="#">TYPE_CLASS_DATETIME   TYPE_DATETIME_VARIATION_TIME</a> .

This corresponds to the global attribute resource symbol [inputType](#)  
[\(/reference/android/R.attr.html#inputType\)](#).

Constant Value: 10 (0x0000000a)

#### public static final int Searchable\_label

This is the user-displayed name of the searchable activity. *Required attribute*.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This corresponds to the global attribute resource symbol [label](#)  
[\(/reference/android/R.attr.html#label\)](#).

Constant Value: 0 (0x00000000)

#### public static final int Searchable\_queryAfterZeroResults

If provided and true, this searchable activity will be invoked for all queries in a particular session. If set to false and the activity returned zero results for a query, it will not be invoked again in that session for supersets of that zero-results query. For example, if the activity returned zero results for "bo", it would not be queried again for "bob". The default value is false. *Optional attribute*.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [queryAfterZeroResults](#)  
[\(/reference/android/R.attr.html#queryAfterZeroResults\)](#).

Constant Value: 19 (0x00000013)

#### public static final int Searchable\_searchButtonText

If supplied, this string will be displayed as the text of the "Search" button. *Optional attribute*.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [searchButtonText](#)  
[\(/reference/android/R.attr.html#searchButtonText\)](#).

Constant Value: 9 (0x00000009)

#### public static final int Searchable\_searchMode

Additional features are controlled by mode bits in this field. Omitting this field, or setting to zero, provides default behavior. *Optional attribute*.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
showSearchLabelAsBadge	0x04	If set, this flag enables the display of the search target (label) within the search bar. If neither bad mode is selected, no badge will be shown.  This is deprecated.
showSearchIconAsBadge	0x08	The default application icon is now always used, so this option is

obsolete.  
If set, this flag causes the suggestion column SUGGEST\_COLUMN\_INTENT\_DATA to be considered as the text for queryRewriteFromData 0x10 suggestion query rewriting. This should only be used when the values in SUGGEST\_COLUMN\_INTENT\_DATA are suitable for user inspection and editing - typically, HTTP/HTTPS Uri's.  
If set, this flag causes the suggestion column SUGGEST\_COLUMN\_TEXT\_1 to be considered as the text for suggestion query rewriting. This should be used for suggestions in queryRewriteFromText 0x20 which no query text is provided and the SUGGEST\_COLUMN\_INTENT\_DATA values are not suitable for user inspection and editing.

This corresponds to the global attribute resource symbol [searchMode](#) ([/reference/android/R.attr.html#searchMode](#)).

Constant Value: 3 (0x00000003)

#### public static final int **Searchable\_searchSettingsDescription**

If provided, this string will be used to describe the searchable item in the searchable items settings within system search settings. *Optional attribute*.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [searchSettingsDescription](#) ([/reference/android/R.attr.html#searchSettingsDescription](#)).

Constant Value: 20 (0x00000014)

#### public static final int **Searchable\_searchSuggestAuthority**

If provided, this is the trigger indicating that the searchable activity provides suggestions as well. The value must be a fully-qualified content provider authority (e.g. "com.example.android.apis.SuggestionProvider") and should match the "android:authorities" tag in your content provider's manifest entry. *Optional attribute*.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [searchSuggestAuthority](#) ([/reference/android/R.attr.html#searchSuggestAuthority](#)).

Constant Value: 4 (0x00000004)

#### public static final int **Searchable\_searchSuggestIntentAction**

If provided, and not overridden by an action in the selected suggestion, this string will be placed in the action field of the [Intent](#) ([/reference/android/content/Intent.html](#)) when the user clicks a suggestion. *Optional attribute*.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [searchSuggestIntentAction](#) ([/reference/android/R.attr.html#searchSuggestIntentAction](#)).

Constant Value: 7 (0x00000007)

#### public static final int **Searchable\_searchSuggestIntentData**

If provided, and not overridden by an action in the selected suggestion, this string will be placed in the data field of the [Intent](#) ([/reference/android/content/Intent.html](#)) when the user clicks a suggestion. *Optional attribute*.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [searchSuggestIntentData](#)

[\(/reference/android/R.attr.html#searchSuggestIntentData\)](#).

Constant Value: 8 (0x00000008)

#### public static final int **Searchable\_searchSuggestPath**

If provided, this will be inserted in the suggestions query Uri, after the authority you have provide but before the standard suggestions path. *Optional attribute.*

Must be a string value, using '\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [searchSuggestPath](#)

[\(/reference/android/R.attr.html#searchSuggestPath\)](#).

Constant Value: 5 (0x00000005)

#### public static final int **Searchable\_searchSuggestSelection**

If provided, suggestion queries will be passed into your query function as the *selection* parameter.

Typically this will be a WHERE clause for your database, and will contain a single question mark, which represents the actual query string that has been typed by the user. If not provided, then the user query text will be appended to the query Uri (after an additional "/"). *Optional attribute.*

Must be a string value, using '\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [searchSuggestSelection](#)

[\(/reference/android/R.attr.html#searchSuggestSelection\)](#).

Constant Value: 6 (0x00000006)

#### public static final int **Searchable\_searchSuggestThreshold**

If provided, this is the minimum number of characters needed to trigger search suggestions. The default value is 0. *Optional attribute.*

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [searchSuggestThreshold](#)

[\(/reference/android/R.attr.html#searchSuggestThreshold\)](#).

Constant Value: 17 (0x00000011)

#### public static final int **Searchable\_voiceLanguage**

If provided, this specifies the spoken language to be expected, and that it will be different than the one set in the [getDefault\(\)](#) ([\(/reference/java/util/Locale.html#getDefault\(\)\)](#)).

Must be a string value, using '\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [voiceLanguage](#)

[\(/reference/android/R.attr.html#voiceLanguage\)](#).

Constant Value: 14 (0x0000000e)

#### public static final int **Searchable\_voiceLanguageModel**

If provided, this specifies the language model that should be used by the voice recognition system. See [EXTRA\\_LANGUAGE\\_MODEL](#) ([\(/reference/android/speech/RecognizerIntent.html#EXTRA\\_LANGUAGE\\_MODEL\)](#)) for more information. If not provided, the default value [LANGUAGE\\_MODEL\\_FREE\\_FORM](#) ([\(/reference/android/speech/RecognizerIntent.html#LANGUAGE\\_MODEL\\_FREE\\_FORM\)](#)) will be used.

Must be a string value, using '\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [voiceLanguageModel](#)  
(/reference/android/R.attr.html#voiceLanguageModel).

Constant Value: 12 (0x0000000c)

#### public static final int **Searchable\_voiceMaxResults**

If provided, enforces the maximum number of results to return, including the "best" result which will always be provided as the SEARCH intent's primary query. Must be one or greater. If not provided, the recognizer will choose how many results to return.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [voiceMaxResults](#)  
(/reference/android/R.attr.html#voiceMaxResults).

Constant Value: 15 (0x0000000f)

#### public static final int **Searchable\_voicePromptText**

If provided, this specifies a prompt that will be displayed during voice input.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [voicePromptText](#)  
(/reference/android/R.attr.html#voicePromptText).

Constant Value: 13 (0x0000000d)

#### public static final int **Searchable\_voiceSearchMode**

Voice search features are controlled by mode bits in this field. Omitting this field, or setting to zero, provides default behavior. If showVoiceSearchButton is set, then launchWebSearch or launchRecognizer must also be set. *Optional attribute*.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
showVoiceSearchButton	0x01	If set, display a voice search button. This only takes effect if voice search is available on the device.
launchWebSearch	0x02	If set, the voice search button will take the user directly to a built-in voice web search activity. Most applications will not use this flag, as it will take the user away from the activity in which search was invoked.
launchRecognizer	0x04	If set, the voice search button will take the user directly to a built-in voice recording activity. This activity will prompt the user to speak, transcribe the spoken text, and forward the resulting query text to the searchable activity, just as if the user had typed it into the search UI and clicked the search button.

This corresponds to the global attribute resource symbol [voiceSearchMode](#)  
(/reference/android/R.attr.html#voiceSearchMode).

Constant Value: 11 (0x0000000b)

#### public static final int **SeekBar\_thumb**

Draws the thumb on a seekbar.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [thumb](#)  
(/reference/android/R.attr.html#thumb).

Constant Value: 0 (0x00000000)

#### public static final int **SeekBar\_thumbOffset**

An offset for the thumb that allows it to extend out of the range of the track.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [thumbOffset](#)  
[\(/reference/android/R.attr.html#thumbOffset\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **SelectionModeDrawables\_actionModeCopyDrawable**

Drawable to use for the Copy action button in Contextual Action Bar

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [actionModeCopyDrawable](#)  
[\(/reference/android/R.attr.html#actionModeCopyDrawable\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **SelectionModeDrawables\_actionModeCutDrawable**

Drawable to use for the Cut action button in Contextual Action Bar

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [actionModeCutDrawable](#)  
[\(/reference/android/R.attr.html#actionModeCutDrawable\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **SelectionModeDrawables\_actionModePasteDrawable**

Drawable to use for the Paste action button in Contextual Action Bar

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [actionModePasteDrawable](#)  
[\(/reference/android/R.attr.html#actionModePasteDrawable\)](#).

Constant Value: 2 (0x00000002)

#### public static final int **SelectionModeDrawables\_actionModeSelectAllDrawable**

Drawable to use for the Select all action button in Contextual Action Bar

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [actionModeSelectAllDrawable](#)  
[\(/reference/android/R.attr.html#actionModeSelectAllDrawable\)](#).

Constant Value: 3 (0x00000003)

#### public static final int **ShapeDrawablePadding\_bottom**

Bottom padding.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [bottom](#)  
[\(/reference/android/R.attr.html#bottom\)](#).

Constant Value: 3 (0x00000003)

#### public static final int **ShapeDrawablePadding\_left**

Left padding.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [left](#) ([/reference/android/R.attr.html#left](#)).

Constant Value: 0 (0x00000000)

#### public static final int **ShapeDrawablePadding\_right**

Right padding.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [right](#) ([/reference/android/R.attr.html#right](#)).

Constant Value: 2 (0x00000002)

#### public static final int **ShapeDrawablePadding\_top**

Top padding.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [top](#) ([/reference/android/R.attr.html#top](#)).

Constant Value: 1 (0x00000001)

#### public static final int **ShapeDrawable\_color**

Defines the color of the shape.

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [color](#) ([/reference/android/R.attr.html#color](#)).

Constant Value: 3 (0x00000003)

#### public static final int **ShapeDrawable\_dither**

Enables or disables dithering.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [dither](#) ([/reference/android/R.attr.html#dither](#)).

Constant Value: 0 (0x00000000)

#### public static final int **ShapeDrawable\_height**

Defines the height of the shape.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in

the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [height](#)  
(/reference/android/R.attr.html#height).

Constant Value: 1 (0x00000001)

#### public static final int **ShapeDrawable\_width**

Defines the width of the shape.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [width](#)  
(/reference/android/R.attr.html#width).

Constant Value: 2 (0x00000002)

#### public static final int **SlidingDrawer\_allowSingleTap**

Indicates whether the drawer can be opened/closed by a single tap on the handle. (If false, the user must drag or fling, or click using the trackball, to open/close the drawer.) Default is true.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [allowSingleTap](#)  
(/reference/android/R.attr.html#allowSingleTap).

Constant Value: 3 (0x00000003)

#### public static final int **SlidingDrawer\_animateOnClick**

Indicates whether the drawer should be opened/closed with an animation when the user clicks the handle. Default is true.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [animateOnClick](#)  
(/reference/android/R.attr.html#animateOnClick).

Constant Value: 6 (0x00000006)

#### public static final int **SlidingDrawer\_bottomOffset**

Extra offset for the handle at the bottom of the SlidingDrawer.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [bottomOffset](#)  
(/reference/android/R.attr.html#bottomOffset).

Constant Value: 1 (0x00000001)

#### public static final int **SlidingDrawer\_content**

Identifier for the child that represents the drawer's content.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [content](#)  
(/reference/android/R.attr.html#content).

Constant Value: 5 (0x00000005)

#### public static final int SlidingDrawer\_handle

Identifier for the child that represents the drawer's handle.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [handle](#)  
[\(/reference/android/R.attr.html#handle\)](#).

Constant Value: 4 (0x00000004)

#### public static final int SlidingDrawer\_orientation

Orientation of the SlidingDrawer.

Must be one of the following constant values.

Constant	Value	Description
horizontal	0	Defines an horizontal widget.
vertical	1	Defines a vertical widget.

This corresponds to the global attribute resource symbol [orientation](#)  
[\(/reference/android/R.attr.html#orientation\)](#).

Constant Value: 0 (0x00000000)

#### public static final int SlidingDrawer\_topOffset

Extra offset for the handle at the top of the SlidingDrawer.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [topOffset](#)  
[\(/reference/android/R.attr.html#topOffset\)](#).

Constant Value: 2 (0x00000002)

#### public static final int SlidingTab\_orientation

Use "horizontal" for a row, "vertical" for a column. The default is horizontal.

Must be one of the following constant values.

Constant	Value	Description
horizontal	0	Defines an horizontal widget.
vertical	1	Defines a vertical widget.

This corresponds to the global attribute resource symbol [orientation](#)  
[\(/reference/android/R.attr.html#orientation\)](#).

Constant Value: 0 (0x00000000)

#### public static final int SpellChecker\_Subtype\_label

The name of the subtype.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This corresponds to the global attribute resource symbol [label](#)  
[\(/reference/android/R.attr.html#label\)](#).

Constant Value: 0 (0x00000000)

#### public static final int SpellChecker\_Subtype\_subtypeExtraValue

The extra value of the subtype. This string can be any string and will be passed to the SpellChecker.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [subtypeExtraValue](#)  
(/reference/android/R.attr.html#subtypeExtraValue).

Constant Value: 2 (0x00000002)

#### public static final int SpellChecker\_Subtype\_subtypeLocale

The locale of the subtype. This string should be a locale (e.g. en\_US, fr\_FR...) This is also used by the framework to know the supported locales of the spell checker.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [subtypeLocale](#)  
(/reference/android/R.attr.html#subtypeLocale).

Constant Value: 1 (0x00000001)

#### public static final int SpellChecker\_label

The name of the spell checker.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This corresponds to the global attribute resource symbol [label](#)  
(/reference/android/R.attr.html#label).

Constant Value: 0 (0x00000000)

#### public static final int SpellChecker\_settingsActivity

Component name of an activity that allows the user to modify the settings for this service.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [settingsActivity](#)  
(/reference/android/R.attr.html#settingsActivity).

Constant Value: 1 (0x00000001)

#### public static final int Spinner\_dropDownHorizontalOffset

Horizontal offset from the spinner widget for positioning the dropdown in spinnerMode="dropdown".

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [dropDownHorizontalOffset](#)  
(/reference/android/R.attr.html#dropDownHorizontalOffset).

Constant Value: 5 (0x00000005)

#### public static final int Spinner\_dropDownSelector

List selector to use for spinnerMode="dropdown" display.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [dropDownSelector](#)

[\(/reference/android/R.attr.html#dropDownSelector\).](#)

Constant Value: 1 (0x00000001)

#### public static final int **Spinner\_dropdownVerticalOffset**

Vertical offset from the spinner widget for positioning the dropdown in spinnerMode="dropdown".

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [dropDownVerticalOffset](#)

[\(/reference/android/R.attr.html#dropDownVerticalOffset\).](#)

Constant Value: 6 (0x00000006)

#### public static final int **Spinner\_dropdownWidth**

Width of the dropdown in spinnerMode="dropdown".

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

May be one of the following constant values.

Constant	Value	Description
fill_parent	-1	The dropdown should fill the width of the screen. This constant is deprecated starting from API Level 8 and is replaced by match_parent.
match_parent	-1	The dropdown should fit the width of the screen. Introduced in API Level 8.
wrap_content	-2	The dropdown should fit the width of its anchor.

This corresponds to the global attribute resource symbol [dropDownWidth](#)  
[\(/reference/android/R.attr.html#dropDownWidth\).](#)

Constant Value: 4 (0x00000004)

#### public static final int **Spinner\_gravity**

Gravity setting for positioning the currently selected item.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
top	0x30	Push object to the top of its container, not changing its size.
bottom	0x50	Push object to the bottom of its container, not changing its size.
left	0x03	Push object to the left of its container, not changing its size.
right	0x05	Push object to the right of its container, not changing its size.
center_vertical	0x10	Place object in the vertical center of its container, not changing its size.
fill_vertical	0x70	Grow the vertical size of the object if needed so it completely fills its container.
center_horizontal	0x01	Place object in the horizontal center of its container, not changing its size.
fill_horizontal	0x07	Grow the horizontal size of the object if needed so it completely fills its container.
center	0x11	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
fill	0x77	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
clip_vertical	0x80	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
clip_horizontal	0x08	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a

right gravity will clip the left edge, and neither will clip both edges.  
start            0x00800003 Push object to the beginning of its container, not changing its size.  
end            0x00800005 Push object to the end of its container, not changing its size.

This corresponds to the global attribute resource symbol [gravity](#)  
[\(/reference/android/R.attr.html#gravity\)](#).

Constant Value: 0 (0x00000000)

#### public static final int Spinner\_popupBackground

Background drawable to use for the dropdown in spinnerMode="dropdown".

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [popupBackground](#)  
[\(/reference/android/R.attr.html#popupBackground\)](#).

Constant Value: 2 (0x00000002)

#### public static final int Spinner\_prompt

The prompt to display when the spinner's dialog is shown.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [prompt](#)  
[\(/reference/android/R.attr.html#prompt\)](#).

Constant Value: 3 (0x00000003)

#### public static final int Spinner\_spinnerMode

Display mode for spinner options.

Must be one of the following constant values.

Constant Value	Description
dialog 0	Spinner options will be presented to the user as a dialog window.
dropdown 1	Spinner options will be presented to the user as an inline dropdown anchored to the spinner widget itself.

This corresponds to the global attribute resource symbol [spinnerMode](#)  
[\(/reference/android/R.attr.html#spinnerMode\)](#).

Constant Value: 7 (0x00000007)

#### public static final int StateListDrawable\_constantSize

If true, the drawable's reported internal size will remain constant as the state changes; the size is the maximum of all of the states. If false, the size will vary based on the current state.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [constantSize](#)  
[\(/reference/android/R.attr.html#constantSize\)](#).

Constant Value: 3 (0x00000003)

#### public static final int StateListDrawable\_dither

Enables or disables dithering of the bitmap if the bitmap does not have the same pixel configuration as the screen (for instance: a ARGB 8888 bitmap with an RGB 565 screen).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [dither](#)  
[\(/reference/android/R.attr.html#dither\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **StateListDrawable\_enterFadeDuration**

Amount of time (in milliseconds) to fade in a new state drawable.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [enterFadeDuration](#)  
(/reference/android/R.attr.html#enterFadeDuration).

Constant Value: 4 (0x00000004)

#### public static final int **StateListDrawable\_exitFadeDuration**

Amount of time (in milliseconds) to fade out an old state drawable.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [exitFadeDuration](#)  
(/reference/android/R.attr.html#exitFadeDuration).

Constant Value: 5 (0x00000005)

#### public static final int **StateListDrawable\_variablePadding**

If true, allows the drawable's padding to change based on the current state that is selected. If false, the padding will stay the same (based on the maximum padding of all the states). Enabling this feature requires that the owner of the drawable deal with performing layout when the state changes, which is often not supported.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [variablePadding](#)  
(/reference/android/R.attr.html#variablePadding).

Constant Value: 2 (0x00000002)

#### public static final int **StateListDrawable\_visible**

Indicates whether the drawable should be initially visible.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [visible](#)  
(/reference/android/R.attr.html#visible).

Constant Value: 1 (0x00000001)

#### public static final int **SwitchPreference\_disableDependentsState**

The state (true for on, or false for off) that causes dependents to be disabled. By default, dependents will be disabled when this is unchecked, so the value of this preference is false.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [disableDependentsState](#)  
(/reference/android/R.attr.html#disableDependentsState).

Constant Value: 2 (0x00000002)

#### public static final int **SwitchPreference\_summaryOff**

The summary for the Preference in a PreferenceActivity screen when the SwitchPreference is unchecked. If separate on/off summaries are not needed, the summary attribute can be used instead.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [summaryOff](#)  
(/reference/android/R.attr.html#summaryOff).

Constant Value: 1 (0x00000001)

#### public static final int **SwitchPreference\_summaryOn**

The summary for the Preference in a PreferenceActivity screen when the SwitchPreference is checked. If separate on/off summaries are not needed, the summary attribute can be used instead.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [summaryOn](#)  
(/reference/android/R.attr.html#summaryOn).

Constant Value: 0 (0x00000000)

#### public static final int **SwitchPreference\_switchTextOff**

The text used on the switch itself when in the "off" state. This should be a very SHORT string, as it appears in a small space.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [switchTextOff](#)  
(/reference/android/R.attr.html#switchTextOff).

Constant Value: 4 (0x00000004)

#### public static final int **SwitchPreference\_switchTextOn**

The text used on the switch itself when in the "on" state. This should be a very SHORT string, as it appears in a small space.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [switchTextOn](#)  
(/reference/android/R.attr.html#switchTextOn).

Constant Value: 3 (0x00000003)

#### public static final int **Switch\_switchMinWidth**

Minimum width for the switch component

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [switchMinWidth](#)  
(/reference/android/R.attr.html#switchMinWidth).

Constant Value: 5 (0x00000005)

#### public static final int **Switch\_switchPadding**

Minimum space between the switch and caption text

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[*package*:] type:*name*") or theme attribute (in the form "?[*package*:] [type:*name*]" containing a value of this type.

This corresponds to the global attribute resource symbol [switchPadding](#)  
(/reference/android/R.attr.html#switchPadding).

Constant Value: 6 (0x00000006)

#### public static final int **Switch\_switchTextAppearance**

TextAppearance style for text displayed on the switch thumb.

Must be a reference to another resource, in the form "@[+][*package*:] type:*name*" or to a theme attribute in the form "?[*package*:] [type:*name*]".

This corresponds to the global attribute resource symbol [switchTextAppearance](#)  
(/reference/android/R.attr.html#switchTextAppearance).

Constant Value: 3 (0x00000003)

#### public static final int **Switch\_textOff**

Text to use when the switch is in the unchecked/"off" state.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[*package*:] type:*name*") or theme attribute (in the form "?[*package*:] [type:*name*]" containing a value of this type.

This corresponds to the global attribute resource symbol [textOff](#)  
(/reference/android/R.attr.html#textOff).

Constant Value: 1 (0x00000001)

#### public static final int **Switch\_textOn**

Text to use when the switch is in the checked/"on" state.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[*package*:] type:*name*") or theme attribute (in the form "?[*package*:] [type:*name*]" containing a value of this type.

This corresponds to the global attribute resource symbol [textOn](#)  
(/reference/android/R.attr.html#textOn).

Constant Value: 0 (0x00000000)

#### public static final int **Switch\_thumb**

Drawable to use as the "thumb" that switches back and forth.

Must be a reference to another resource, in the form "@[+][*package*:] type:*name*" or to a theme attribute in the form "?[*package*:] [type:*name*]".

This corresponds to the global attribute resource symbol [thumb](#)  
(/reference/android/R.attr.html#thumb).

Constant Value: 2 (0x00000002)

#### public static final int **Switch\_thumbTextPadding**

Amount of padding on either side of text within the switch thumb.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[*package*:] type:*name*") or theme attribute (in the form "?[*package*:] [type:*name*]" containing a value of this type.

This corresponds to the global attribute resource symbol [thumbTextPadding](#)  
(/reference/android/R.attr.html#thumbTextPadding).

Constant Value: 7 (0x00000007)

## public static final int **Switch\_track**

Drawable to use as the "track" that the switch thumb slides within.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*] [*type:*]*name*".

This corresponds to the global attribute resource symbol [track](#)  
[\(/reference/android/R.attr.html#track\)](#).

Constant Value: 4 (0x00000004)

## public static final int **SyncAdapter\_accountType**

The account type this authenticator handles.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*] [*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [accountType](#)  
[\(/reference/android/R.attr.html#accountType\)](#).

Constant Value: 1 (0x00000001)

## public static final int **SyncAdapter\_allowParallelSyncs**

Set to true to tell the SyncManager that this SyncAdapter supports multiple simultaneous syncs for the same account type and authority. Otherwise the SyncManager will be sure not to issue a start sync request to this SyncAdapter if the SyncAdapter is already syncing another account. Defaults to false.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*] [*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [allowParallelSyncs](#)  
[\(/reference/android/R.attr.html#allowParallelSyncs\)](#).

Constant Value: 5 (0x00000005)

## public static final int **SyncAdapter\_contentAuthority**

the authority of a content provider.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*] [*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [contentAuthority](#)  
[\(/reference/android/R.attr.html#contentAuthority\)](#).

Constant Value: 2 (0x00000002)

## public static final int **SyncAdapter\_isAlwaysSyncable**

Set to true to tell the SyncManager to automatically call setSyncable(..., ..., 1) for the SyncAdapter instead of issuing an initialization sync to the SyncAdapter. Defaults to false.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*] [*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [isAlwaysSyncable](#)  
[\(/reference/android/R.attr.html#isAlwaysSyncable\)](#).

Constant Value: 6 (0x00000006)

## public static final int **SyncAdapter\_settingsActivity**

If provided, specifies the action of the settings activity for this SyncAdapter.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*] [*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [settingsActivity](#)  
(/reference/android/R.attr.html#settingsActivity).

Constant Value: 0 (0x00000000)

#### **public static final int SyncAdapter\_supportsUploading**

This symbol is the offset where the [supportsUploading](#)  
(/reference/android/R.attr.html#supportsUploading) attribute's value can be found in the  
[SyncAdapter](#) (/reference/android/R.styleable.html#SyncAdapter) array.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 4 (0x00000004)

#### **public static final int SyncAdapter\_userVisible**

This symbol is the offset where the [userVisible](#) (/reference/android/R.attr.html#userVisible)  
attribute's value can be found in the [SyncAdapter](#) (/reference/android/R.styleable.html#SyncAdapter)  
array.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 3 (0x00000003)

#### **public static final int TabWidget\_divider**

Drawable used to draw the divider between tabs.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [divider](#)  
(/reference/android/R.attr.html#divider).

Constant Value: 0 (0x00000000)

#### **public static final int TabWidget\_tabStripEnabled**

Determines whether the strip under the tab indicators is drawn or not.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [tabStripEnabled](#)  
(/reference/android/R.attr.html#tabStripEnabled).

Constant Value: 3 (0x00000003)

#### **public static final int TabWidget\_tabStripLeft**

Drawable used to draw the left part of the strip underneath the tabs.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [tabStripLeft](#)  
(/reference/android/R.attr.html#tabStripLeft).

Constant Value: 1 (0x00000001)

#### **public static final int TabWidget\_tabStripRight**

Drawable used to draw the right part of the strip underneath the tabs.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [tabStripRight](#)  
(/reference/android/R.attr.html#tabStripRight).

Constant Value: 2 (0x00000002)

#### public static final int **TableLayout\_collapseColumns**

The zero-based index of the columns to collapse. The column indices must be separated by a comma: 1, 2, 5. Illegal and duplicate indices are ignored.

Must be a string value, using '\\,' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [collapseColumns](#)  
(/reference/android/R.attr.html#collapseColumns).

Constant Value: 2 (0x00000002)

#### public static final int **TableLayout\_shrinkColumns**

The zero-based index of the columns to shrink. The column indices must be separated by a comma: 1, 2, 5. Illegal and duplicate indices are ignored. You can shrink all columns by using the value "\*" instead. Note that a column can be marked stretchable and shrinkable at the same time.

Must be a string value, using '\\,' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [shrinkColumns](#)  
(/reference/android/R.attr.html#shrinkColumns).

Constant Value: 1 (0x00000001)

#### public static final int **TableLayout\_stretchColumns**

The zero-based index of the columns to stretch. The column indices must be separated by a comma: 1, 2, 5. Illegal and duplicate indices are ignored. You can stretch all columns by using the value "\*" instead. Note that a column can be marked stretchable and shrinkable at the same time.

Must be a string value, using '\\,' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [stretchColumns](#)  
(/reference/android/R.attr.html#stretchColumns).

Constant Value: 0 (0x00000000)

#### public static final int **TableRow\_Cell\_layout\_column**

The index of the column in which this child should be.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_column](#)  
(/reference/android/R.attr.html#layout\_column).

Constant Value: 0 (0x00000000)

#### public static final int **TableRow\_Cell\_layout\_span**

Defines how many columns this child should span. Must be >= 1.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_span](#)  
(/reference/android/R.attr.html#layout\_span).

Constant Value: 1 (0x00000001)

## **public static final int TextAppearance\_fontFamily**

Font family (named by string) for the text.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [fontFamily](#)  
(/reference/android/R.attr.html#fontFamily).

Constant Value: 8 (0x00000008)

## **public static final int TextAppearance\_textAllCaps**

Present the text in ALL CAPS. This may use a small-caps form when available.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [textAllCaps](#)  
(/reference/android/R.attr.html#textAllCaps).

Constant Value: 7 (0x00000007)

## **public static final int TextAppearance\_textColor**

Text color.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColor](#)  
(/reference/android/R.attr.html#textColor).

Constant Value: 3 (0x00000003)

## **public static final int TextAppearance\_textColorHighlight**

Color of the text selection highlight.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorHighlight](#)  
(/reference/android/R.attr.html#textColorHighlight).

Constant Value: 4 (0x00000004)

## **public static final int TextAppearance\_textColorHint**

Color of the hint text.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorHint](#)  
(/reference/android/R.attr.html#textColorHint).

Constant Value: 5 (0x00000005)

## **public static final int TextAppearance\_textColorLink**

Color of the links.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorLink](#)  
(/reference/android/R.attr.html#textColorLink).

Constant Value: 6 (0x00000006)

#### public static final int **TextAppearance\_textSize**

Size of the text. Recommended dimension type for text is "sp" for scaled-pixels (example: 15sp).

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [textSize](#)  
(/reference/android/R.attr.html#textSize).

Constant Value: 0 (0x00000000)

#### public static final int **TextAppearance textStyle**

Style (bold, italic, bolditalic) for the text.

Must be one or more (separated by '|') of the following constant values.

##### Constant Value Description

normal	0
bold	1
italic	2

This corresponds to the global attribute resource symbol [textStyle](#)  
(/reference/android/R.attr.html#textStyle).

Constant Value: 2 (0x00000002)

#### public static final int **TextAppearance\_typeface**

Typeface (normal, sans, serif, monospace) for the text.

Must be one of the following constant values.

##### Constant Value Description

normal	0
sans	1
serif	2
monospace	3

This corresponds to the global attribute resource symbol [typeface](#)  
(/reference/android/R.attr.html#typeface).

Constant Value: 1 (0x00000001)

#### public static final int **TextClock\_format12Hour**

Specifies the formatting pattern used to show the time and/or date in 12-hour mode. Please refer to [DateFormat](#) (/reference/android/text/format/DateFormat.html) for a complete description of accepted formatting patterns. The default pattern is "h:mm aa".

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [format12Hour](#)  
(/reference/android/R.attr.html#format12Hour).

Constant Value: 0 (0x00000000)

#### public static final int **TextClock\_format24Hour**

Specifies the formatting pattern used to show the time and/or date in 24-hour mode. Please refer to [DateFormat](#) (/reference/android/text/format/DateFormat.html) for a complete description of accepted formatting patterns. The default pattern is "k:mm".

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [format24Hour](#)  
(/reference/android/R.attr.html#format24Hour).

Constant Value: 1 (0x00000001)

#### public static final int **TextClock\_timeZone**

Specifies the time zone to use. When this attribute is specified, the TextClock will ignore the time zone of the system. To use the user's time zone, do not specify this attribute. The default value is the user's time zone. Please refer to [TimeZone](#) (/reference/java/util/TimeZone.html) for more information about time zone ids.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [timeZone](#)  
(/reference/android/R.attr.html#timeZone).

Constant Value: 2 (0x00000002)

#### public static final int **TextToSpeechEngine\_settingsActivity**

Component name of an activity that allows the user to modify the settings for this service.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [settingsActivity](#)  
(/reference/android/R.attr.html#settingsActivity).

Constant Value: 0 (0x00000000)

#### public static final int **TextViewAppearance\_textAppearance**

Base text color, typeface, size, and style.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [textAppearance](#)  
(/reference/android/R.attr.html#textAppearance).

Constant Value: 0 (0x00000000)

#### public static final int **TextViewMultiLineBackgroundState\_state\_multiline**

State identifier indicating a TextView has a multi-line layout.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_multiline](#)  
(/reference/android/R.attr.html#state\_multiline).

Constant Value: 0 (0x00000000)

#### public static final int **TextView\_autoLink**

Controls whether links such as urls and email addresses are automatically found and converted to clickable links. The default value is "none", disabling this feature.

Must be one or more (separated by '|') of the following constant values.

Constant Value	Description
none	0x00 Match no patterns (default).
web	0x01 Match Web URLs.
email	0x02 Match email addresses.
phone	0x04 Match phone numbers.

map 0x08 Match map addresses.

all 0x0f Match all patterns (equivalent to weblemail|phone|map).

This corresponds to the global attribute resource symbol [autoLink](#)

[\(/reference/android/R.attr.html#autoLink\)](#).

Constant Value: 11 (0x0000000b)

#### public static final int **TextView\_autoText**

If set, specifies that this TextView has a textual input method and automatically corrects some common spelling errors. The default is "false".

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [autoText](#)

[\(/reference/android/R.attr.html#autoText\)](#).

Constant Value: 45 (0x0000002d)

#### public static final int **TextView\_bufferType**

Determines the minimum type that getText() will return. The default is "normal". Note that EditText and LogTextBox always return Editable, even if you specify something less powerful here.

Must be one of the following constant values.

Constant	Value	Description
normal	0	Can return any CharSequence, possibly a Spanned one if the source text was Spanned.
spannable	1	Can only return Spannable.
editable	2	Can only return Spannable and Editable.

This corresponds to the global attribute resource symbol [bufferType](#)

[\(/reference/android/R.attr.html#bufferType\)](#).

Constant Value: 17 (0x00000011)

#### public static final int **TextView\_capitalize**

If set, specifies that this TextView has a textual input method and should automatically capitalize what the user types. The default is "none".

Must be one of the following constant values.

Constant	Value	Description
none	0	Don't automatically capitalize anything.
sentences	1	Capitalize the first word of each sentence.
words	2	Capitalize the first letter of every word.
characters	3	Capitalize every character.

This corresponds to the global attribute resource symbol [capitalize](#)

[\(/reference/android/R.attr.html#capitalize\)](#).

Constant Value: 44 (0x0000002c)

#### public static final int **TextView\_cursorVisible**

Makes the cursor visible (the default) or invisible.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [cursorVisible](#)

[\(/reference/android/R.attr.html#cursorVisible\)](#).

Constant Value: 21 (0x00000015)

#### public static final int **TextView\_digits**

If set, specifies that this TextView has a numeric input method and that these specific characters are the ones that it will accept. If this is set, numeric is implied to be true. The default is false.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [digits](#)  
[\(/reference/android/R.attr.html#digits\)](#).

Constant Value: 41 (0x00000029)

#### public static final int **TextView\_drawableBottom**

The drawable to be drawn below the text.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [drawableBottom](#)  
[\(/reference/android/R.attr.html#drawableBottom\)](#).

Constant Value: 49 (0x00000031)

#### public static final int **TextView\_drawableEnd**

The drawable to be drawn to the end of the text.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [drawableEnd](#)  
[\(/reference/android/R.attr.html#drawableEnd\)](#).

Constant Value: 74 (0x0000004a)

#### public static final int **TextView\_drawableLeft**

The drawable to be drawn to the left of the text.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [drawableLeft](#)  
[\(/reference/android/R.attr.html#drawableLeft\)](#).

Constant Value: 50 (0x00000032)

#### public static final int **TextView\_drawablePadding**

The padding between the drawables and the text.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [drawablePadding](#)  
[\(/reference/android/R.attr.html#drawablePadding\)](#).

Constant Value: 52 (0x00000034)

#### public static final int **TextView\_drawableRight**

The drawable to be drawn to the right of the text.

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [drawableRight](#)  
[\(/reference/android/R.attr.html#drawableRight\)](#).

Constant Value: 51 (0x00000033)

#### public static final int **TextView\_drawableStart**

The drawable to be drawn to the start of the text.

May be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ][ type: ] name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [drawableStart](#)  
[\(/reference/android/R.attr.html#drawableStart\)](#).

Constant Value: 73 (0x00000049)

#### public static final int **TextView\_drawableTop**

The drawable to be drawn above the text.

May be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ][ type: ] name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [drawableTop](#)  
[\(/reference/android/R.attr.html#drawableTop\)](#).

Constant Value: 48 (0x00000030)

#### public static final int **TextView\_editable**

If set, specifies that this TextView has an input method. It will be a textual one unless it has otherwise been specified. For TextView, this is false by default. For EditText, it is true by default.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ][ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [editable](#)  
[\(/reference/android/R.attr.html#editable\)](#).

Constant Value: 46 (0x0000002e)

#### public static final int **TextView\_editorExtras**

Reference to an [<input\\_extras>](#) [\(/reference/android/R.styleable.html#InputExtras\)](#) XML resource containing additional data to supply to an input method, which is private to the implementation of the input method. This simply fills in the [EditorInfo.extras](#) [\(/reference/android/view/inputmethod/EditorInfo.html#extras\)](#) field when the input method is connected.

Must be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ][ type: ] name`".

This corresponds to the global attribute resource symbol [editorExtras](#)  
[\(/reference/android/R.attr.html#editorExtras\)](#).

Constant Value: 58 (0x0000003a)

#### public static final int **TextView\_ellipsize**

If set, causes words that are longer than the view is wide to be ellipsized instead of broken in the middle. You will often also want to set scrollHorizontally or singleLine as well so that the text as a whole is also constrained to a single line instead of still allowed to be broken onto multiple lines.

Must be one of the following constant values.

##### **Constant Value Description**

none 0

start 1

middle 2

end 3

marquee 4

This corresponds to the global attribute resource symbol [ellipsize](#)

[\(/reference/android/R.attr.html#ellipsize\).](#)

Constant Value: 9 (0x00000009)

#### public static final int **TextView\_ems**

Makes the TextView be exactly this many ems wide.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [ems](#) [\(/reference/android/R.attr.html#ems\).](#)

Constant Value: 27 (0x0000001b)

#### public static final int **TextView\_enabled**

Specifies whether the widget is enabled. The interpretation of the enabled state varies by subclass. For example, a non-enabled EditText prevents the user from editing the contained text, and a non-enabled Button prevents the user from tapping the button. The appearance of enabled and non-enabled widgets may differ, if the drawables referenced from evaluating state\_enabled differ.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [enabled](#) [\(/reference/android/R.attr.html#enabled\).](#)

Constant Value: 0 (0x00000000)

#### public static final int **TextView\_fontFamily**

Font family (named by string) for the text.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [fontFamily](#) [\(/reference/android/R.attr.html#fontFamily\).](#)

Constant Value: 75 (0x00000004b)

#### public static final int **TextView\_freezesText**

If set, the text view will include its current complete text inside of its frozen icicle in addition to meta-data such as the current cursor position. By default this is disabled; it can be useful when the contents of a text view is not stored in a persistent place such as a content provider.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [freezesText](#) [\(/reference/android/R.attr.html#freezesText\).](#)

Constant Value: 47 (0x0000002f)

#### public static final int **TextView\_gravity**

Specifies how to align the text by the view's x- and/or y-axis when the text is smaller than the view.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
top	0x30	Push object to the top of its container, not changing its size.
bottom	0x50	Push object to the bottom of its container, not changing its size.
left	0x03	Push object to the left of its container, not changing its size.
right	0x05	Push object to the right of its container, not changing its size.
center_vertical	0x10	Place object in the vertical center of its container, not changing its size.

<code>fill_vertical</code>	<code>0x70</code>	Grow the vertical size of the object if needed so it completely fills its container.
<code>center_horizontal</code>	<code>0x01</code>	Place object in the horizontal center of its container, not changing its size.
<code>fill_horizontal</code>	<code>0x07</code>	Grow the horizontal size of the object if needed so it completely fills its container.
<code>center</code>	<code>0x11</code>	Place the object in the center of its container in both the vertical and horizontal axis, not changing its size.
<code>fill</code>	<code>0x77</code>	Grow the horizontal and vertical size of the object if needed so it completely fills its container.
<code>clip_vertical</code>	<code>0x80</code>	Additional option that can be set to have the top and/or bottom edges of the child clipped to its container's bounds. The clip will be based on the vertical gravity: a top gravity will clip the bottom edge, a bottom gravity will clip the top edge, and neither will clip both edges.
<code>clip_horizontal</code>	<code>0x08</code>	Additional option that can be set to have the left and/or right edges of the child clipped to its container's bounds. The clip will be based on the horizontal gravity: a left gravity will clip the right edge, a right gravity will clip the left edge, and neither will clip both edges.
<code>start</code>	<code>0x00800003</code>	Push object to the beginning of its container, not changing its size.
<code>end</code>	<code>0x00800005</code>	Push object to the end of its container, not changing its size.

This corresponds to the global attribute resource symbol [gravity](#)  
[\(/reference/android/R.attr.html#gravity\)](#).

Constant Value: 10 (0x0000000a)

#### **public static final int TextView\_height**

Makes the TextView be exactly this many pixels tall. You could get the same effect by specifying this number in the layout parameters.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [height](#)  
[\(/reference/android/R.attr.html#height\)](#).

Constant Value: 24 (0x00000018)

#### **public static final int TextView\_hint**

Hint text to display when the text is empty.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [hint](#) [\(/reference/android/R.attr.html#hint\)](#).

Constant Value: 19 (0x00000013)

#### **public static final int TextView imeActionId**

Supply a value for [EditorInfo.actionId](#)  
[\(/reference/android/view/inputmethod/EditorInfo.html#actionId\)](#) used when an input method is connected to the text view.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?@[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [imeActionId](#)  
[\(/reference/android/R.attr.html#imeActionId\)](#).

Constant Value: 61 (0x0000003d)

#### **public static final int TextView\_imeActionButton**

Supply a value for [EditorInfo.actionLabel](#)

[\(/reference/android/view/inputmethod/EditorInfo.html#actionLabel\)](#) used when an input method is connected to the text view.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [imeActionButton](#)  
[\(/reference/android/R.attr.html#imeActionButton\)](#).

Constant Value: 60 (0x0000003c)

#### public static final int **TextView\_imeOptions**

Additional features you can enable in an IME associated with an editor to improve the integration with your application. The constants here correspond to those defined by [imeOptions](#)  
[\(/reference/android/view/inputmethod/EditorInfo.html#imeOptions\)](#).

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
normal	0x00000000	There are no special semantics associated with this editor.
actionUnspecified	0x00000000	There is no specific action associated with this editor, let the editor come up with its own if it can. Corresponds to <a href="#">IME_NULL</a> .
actionNone	0x00000001	This editor has no action associated with it. Corresponds to <a href="#">IME_ACTION_NONE</a> .
actionGo	0x00000002	The action key performs a "go" operation to take the user to the target of the text they typed. Typically used, for example, when entering a URL. Corresponds to <a href="#">IME_ACTION_GO</a> .
actionSearch	0x00000003	The action key performs a "search" operation, taking the user to the results of searching for the text they have typed (in whatever context is appropriate). Corresponds to <a href="#">IME_ACTION_SEARCH</a> .
actionSend	0x00000004	The action key performs a "send" operation, delivering the text to its target. This is typically used when composing a message. Corresponds to <a href="#">IME_ACTION_SEND</a> .
actionNext	0x00000005	The action key performs a "next" operation, taking the user to the next field that will accept text. Corresponds to <a href="#">IME_ACTION_NEXT</a> .
actionDone	0x00000006	The action key performs a "done" operation, closing the soft input method. Corresponds to <a href="#">IME_ACTION_DONE</a> .
actionPrevious	0x00000007	The action key performs a "previous" operation, taking the user to the previous field that will accept text. Corresponds to <a href="#">IME_ACTION_PREVIOUS</a> .
flagNoFullscreen	0x2000000	Used to request that the IME never go into fullscreen mode. Applications need to be aware that the flag is not a guarantee, and not all IMEs will respect it. Corresponds to <a href="#">IME_FLAG_NO_FULLSCREEN</a> <a href="#">(/reference/android/view/inputmethod/EditorInfo.html#IME_FLAG_NO_FULLSCREEN)</a> .
flagNavigatePrevious	0x4000000	Like flagNavigateNext, but specifies there is something interesting that a backward navigation can focus on. If the user selects the IME's facility to backward navigate, this will show up in the application as an actionPrevious at <a href="#">InputConnection.performEditorAction(int)</a> . Corresponds to <a href="#">IME_FLAG_NO_FULLSCREEN</a> <a href="#">(/reference/android/view/inputmethod/EditorInfo.html#IME_FLAG_NO_FULLSCREEN)</a> .
flagNavigateNext	0x8000000	Used to specify that there is something interesting that a forward navigation can focus on. This is like using actionNext, except allows the IME to be multiline (with an enter key) as well as provide forward navigation. Note that some IMEs may not be able to do this, especially when running on a small screen where there is little space. In that case it does not need to present a UI for this option. Like actionNext, if the user selects the IME's facility to forward navigate, this will show up in the application at <a href="#">InputConnection.performEditorAction(int)</a> . Corresponds to <a href="#">IME_FLAG_NAVIGATE_NEXT</a> <a href="#">(/reference/android/view/inputmethod/EditorInfo.html#IME_FLAG_NAVIGATE_NEXT)</a> .
flagNoExtractUi	0x10000000	Used to specify that the IME does not need to show its extracted text UI. For input methods that may be fullscreen, often when in landscape mode, this allows them to be smaller and let part of the application be shown behind. Though there will likely be limited access to the application available from the user, it can make the experience of a (mostly) fullscreen IME less jarring. Note that when this flag is specified the IME may <i>not</i> be set up to be able to display text, so it should only be used in situations where this is not needed. Corresponds to <a href="#">IME_FLAG_NO_EXTRACT_UI</a> <a href="#">(/reference/android/view/inputmethod/EditorInfo.html#IME_FLAG_NO_EXTRACT_UI)</a> .
flagNoAccessoryAction	0x20000000	Used in conjunction with a custom action, this indicates that the action should not be available as an accessory button when the input method is full-screen. Note that by setting this flag, there can be cases where the action is simply never available to the user. Setting this generally means that you think showing text being edited is more important than the action you have supplied.

Corresponds to [IME\\_FLAG\\_NO\\_ACCESSORY\\_ACTION](#)

([/reference/android/view/inputmethod/EditorInfo.html#IME\\_FLAG\\_NO\\_ACCESSORY\\_ACTION](#)).

Used in conjunction with a custom action, this indicates that the action should not be available in-line as a replacement for the "enter" key. Typically this is because the action has such a significant impact or is not recoverable enough that accidentally hitting it should be avoided, such as sending a message. Note that [TextView](#) will automatically set this flag for you on multi-line text views.

Corresponds to [IME\\_FLAG\\_NO\\_ENTER\\_ACTION](#)

([/reference/android/view/inputmethod/EditorInfo.html#IME\\_FLAG\\_NO\\_ENTER\\_ACTION](#)).

Used to request that the IME should be capable of inputting ASCII characters. The intention of this flag is to ensure that the user can type Roman alphabet characters in a [TextView](#) used for, typically, account ID or password input. It is expected that IMEs normally are able to input ASCII even without being told so (such IMEs already respect this flag in a sense), but there could be some cases they aren't when, for instance, only non-ASCII input languages like Arabic, Greek, Hebrew, Russian are enabled in the IME. Applications need to be aware that the flag is not a guarantee, and not all IMEs will respect it. However, it is strongly recommended for IME authors to respect this flag especially when their IME could end up with a state that has only non-ASCII input languages enabled.

Corresponds to [IME\\_FLAG\\_FORCE\\_ASCII](#)

([/reference/android/view/inputmethod/EditorInfo.html#IME\\_FLAG\\_FORCE\\_ASCII](#)).

This corresponds to the global attribute resource symbol [imeOptions](#)

([/reference/android/R.attr.html#imeOptions](#)).

Constant Value: 59 (0x0000003b)

#### public static final int [TextView\\_includeFontPadding](#)

Leave enough room for ascenders and descenders instead of using the font ascent and descent strictly.  
(Normally true).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [includeFontPadding](#)

([/reference/android/R.attr.html#includeFontPadding](#)).

Constant Value: 34 (0x00000022)

#### public static final int [TextView\\_inputMethod](#)

If set, specifies that this [TextView](#) should use the specified input method (specified by fully-qualified class name).

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [inputMethod](#)

([/reference/android/R.attr.html#inputMethod](#)).

Constant Value: 43 (0x0000002b)

#### public static final int [TextView\\_inputType](#)

The type of data being placed in a text field, used to help an input method decide how to let the user enter text. The constants here correspond to those defined by [InputType](#)  
([/reference/android/text/InputType.html](#)). Generally you can select a single value, though some can be combined together as indicated. Setting this attribute to anything besides *none* also implies that the text is editable.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
none	0x00000000	There is no content type. The text is not editable.
text	0x00000001	Just plain old text. Corresponds to <a href="#">TYPE_CLASS_TEXT</a>   <a href="#">TYPE_TEXT_VARIATION_NORMAL</a> .
textCapCharacters	0x00001001	Can be combined with <i>text</i> and its variations to request capitalization of all characters. Corresponds to <a href="#">TYPE_TEXT_FLAG_CAP_CHARACTERS</a> .

textCapWords	0x00002001	Can be combined with <i>text</i> and its variations to request capitalization of the first character of every word. Corresponds to <u><a href="#">TYPE_TEXT_FLAG_CAP_WORDS</a></u> .
textCapSentences	0x00004001	Can be combined with <i>text</i> and its variations to request capitalization of the first character of every sentence. Corresponds to <u><a href="#">TYPE_TEXT_FLAG_CAP_SENTENCES</a></u> .
textAutoCorrect	0x00008001	Can be combined with <i>text</i> and its variations to request auto-correction of text being input. Corresponds to <u><a href="#">TYPE_TEXT_FLAG_AUTO_CORRECT</a></u> .
textAutoComplete	0x00010001	Can be combined with <i>text</i> and its variations to specify that this field will be doing its own auto-completion and talking with the input method appropriately. Corresponds to <u><a href="#">TYPE_TEXT_FLAG_AUTO_COMPLETE</a></u> .
textMultiLine	0x00020001	Can be combined with <i>text</i> and its variations to allow multiple lines of text in the field. If this flag is not set, the text field will be constrained to a single line. Corresponds to <u><a href="#">TYPE_TEXT_FLAG_MULTI_LINE</a></u> .
textImeMultiLine	0x00040001	Can be combined with <i>text</i> and its variations to indicate that though the regular text view should not be multiple lines, the IME should provide multiple lines if it can. Corresponds to <u><a href="#">TYPE_TEXT_FLAG_IME_MULTI_LINE</a></u> .
textNoSuggestions	0x00080001	Can be combined with <i>text</i> and its variations to indicate that the IME should not show any dictionary-based word suggestions. Corresponds to <u><a href="#">TYPE_TEXT_FLAG_NO_SUGGESTIONS</a></u> .
textUri	0x00000011	Text that will be used as a URI. Corresponds to <u><a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_URI</a></u> .
textEmailAddress	0x00000021	Text that will be used as an e-mail address. Corresponds to <u><a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_EMAIL_ADDRESS</a></u> .
textEmailSubject	0x00000031	Text that is being supplied as the subject of an e-mail. Corresponds to <u><a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_EMAIL_SUBJECT</a></u> .
textShortMessage	0x00000041	Text that is the content of a short message. Corresponds to <u><a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_SHORT_MESSAGE</a></u> .
textLongMessage	0x00000051	Text that is the content of a long message. Corresponds to <u><a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_LONG_MESSAGE</a></u> .
textPersonName	0x00000061	Text that is the name of a person. Corresponds to <u><a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_PERSON_NAME</a></u> .
textPostalAddress	0x00000071	Text that is being supplied as a postal mailing address. Corresponds to <u><a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_POSTAL_ADDRESS</a></u> .
textPassword	0x00000081	Text that is a password. Corresponds to <u><a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_PASSWORD</a></u> .
textVisiblePassword	0x00000091	Text that is a password that should be visible. Corresponds to <u><a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_VISIBLE_PASSWORD</a></u> .
textWebEditText	0x000000a1	Text that is being supplied as text in a web form. Corresponds to <u><a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_WEB_EDIT_TEXT</a></u> .
textFilter	0x000000b1	Text that is filtering some other data. Corresponds to <u><a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_FILTER</a></u> .
textPhonetic	0x000000c1	Text that is for phonetic pronunciation, such as a phonetic name field in a contact entry. Corresponds to <u><a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_PHONETIC</a></u> .
textWebEmailAddress	0x000000d1	Text that will be used as an e-mail address on a web form. Corresponds to <u><a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_WEB_EMAIL_ADDRESS</a></u> .
textWebPassword	0x000000e1	Text that will be used as a password on a web form. Corresponds to <u><a href="#">TYPE_CLASS_TEXT TYPE_TEXT_VARIATION_WEB_PASSWORD</a></u> .
number	0x00000002	A numeric only field. Corresponds to <u><a href="#">TYPE_CLASS_NUMBER TYPE_NUMBER_VARIATION_NORMAL</a></u> .
numberSigned	0x00001002	Can be combined with <i>number</i> and its other options to allow a signed number. Corresponds to <u><a href="#">TYPE_CLASS_NUMBER TYPE_NUMBER_FLAG_SIGNED</a></u> .
numberDecimal	0x00002002	Can be combined with <i>number</i> and its other options to allow a decimal (fractional) number. Corresponds to <u><a href="#">TYPE_CLASS_NUMBER TYPE_NUMBER_FLAG_DECIMAL</a></u> .
numberPassword	0x00000012	A numeric password field. Corresponds to <u><a href="#">TYPE_CLASS_NUMBER TYPE_NUMBER_VARIATION_PASSWORD</a></u> .
phone	0x00000003	For entering a phone number. Corresponds to <u><a href="#">TYPE_CLASS_PHONE</a></u> .

<code>datetime</code>	0x00000004	For entering a date and time. Corresponds to <code>TYPE_CLASS_DATETIME   TYPE_DATETIME_VARIATION_NORMAL</code> .
<code>date</code>	0x00000014	For entering a date. Corresponds to <code>TYPE_CLASS_DATETIME   TYPE_DATETIME_VARIATION_DATE</code> .
<code>time</code>	0x00000024	For entering a time. Corresponds to <code>TYPE_CLASS_DATETIME   TYPE_DATETIME_VARIATION_TIME</code> .

This corresponds to the global attribute resource symbol `inputType`  
`(/reference/android/R.attr.html#inputType)`.

Constant Value: 56 (0x00000038)

#### `public static final int TextView_lineSpacingExtra`

Extra spacing between lines of text.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol `lineSpacingExtra`  
`(/reference/android/R.attr.html#lineSpacingExtra)`.

Constant Value: 53 (0x00000035)

#### `public static final int TextView_lineSpacingMultiplier`

Extra spacing between lines of text, as a multiplier.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol `lineSpacingMultiplier`  
`(/reference/android/R.attr.html#lineSpacingMultiplier)`.

Constant Value: 54 (0x00000036)

#### `public static final int TextView_lines`

Makes the TextView be exactly this many lines tall.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol `lines`  
`(/reference/android/R.attr.html#lines)`.

Constant Value: 23 (0x00000017)

#### `public static final int TextView_linksClickable`

If set to false, keeps the movement method from being set to the link movement method even if autoLink causes links to be found.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol `linksClickable`  
`(/reference/android/R.attr.html#linksClickable)`.

Constant Value: 12 (0x0000000c)

#### `public static final int TextView_marqueeRepeatLimit`

The number of times to repeat the marquee animation. Only applied if the TextView has marquee enabled.

May be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in

the form "?[ package: ] [ type: ] name") containing a value of this type.

May be one of the following constant values.

Constant	Value	Description
marquee_forever	-1	Indicates that marquee should repeat indefinitely. This corresponds to the global attribute resource symbol <a href="#">marqueeRepeatLimit</a> ( <a href="#">/reference/android/R.attr.html#marqueeRepeatLimit</a> ).

Constant Value: 55 (0x00000037)

#### **public static final int TextView\_maxEms**

Makes the TextView be at most this many ems wide.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "?[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [maxEms](#) ([/reference/android/R.attr.html#maxEms](#)).

Constant Value: 26 (0x0000001a)

#### **public static final int TextView\_maxHeight**

Makes the TextView be at most this many pixels tall.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "?[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [maxHeight](#) ([/reference/android/R.attr.html#maxHeight](#)).

Constant Value: 14 (0x0000000e)

#### **public static final int TextView\_maxLength**

Set an input filter to constrain the text length to the specified number.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "?[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [maxLength](#) ([/reference/android/R.attr.html#maxLength](#)).

Constant Value: 35 (0x00000023)

#### **public static final int TextView\_maxLines**

Makes the TextView be at most this many lines tall. When used on an editable text, the `inputType` attribute's value must be combined with the `textMultiLine` flag for the `maxLines` attribute to apply.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "?[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [maxLines](#) ([/reference/android/R.attr.html#maxLines](#)).

Constant Value: 22 (0x00000016)

#### **public static final int TextView\_maxWidth**

Makes the TextView be at most this many pixels wide.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [maxWidth](#)  
(/reference/android/R.attr.html#maxWidth).

Constant Value: 13 (0x0000000d)

#### public static final int **TextView\_minEms**

Makes the TextView be at least this many ems wide.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [minEms](#)  
(/reference/android/R.attr.html#minEms).

Constant Value: 29 (0x0000001d)

#### public static final int **TextView\_minHeight**

Makes the TextView be at least this many pixels tall.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [minHeight](#)  
(/reference/android/R.attr.html#minHeight).

Constant Value: 16 (0x00000010)

#### public static final int **TextView\_minLines**

Makes the TextView be at least this many lines tall. When used on an editable text, the `inputType` attribute's value must be combined with the `textMultiLine` flag for the `minLines` attribute to apply.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [minLines](#)  
(/reference/android/R.attr.html#minLines).

Constant Value: 25 (0x00000019)

#### public static final int **TextView\_minWidth**

Makes the TextView be at least this many pixels wide.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [minWidth](#)  
(/reference/android/R.attr.html#minWidth).

Constant Value: 15 (0x0000000f)

#### public static final int **TextView\_numeric**

If set, specifies that this TextView has a numeric input method. The default is false.

Must be one or more (separated by '|') of the following constant values.

Constant Value	Description
integer 0x01	Input is numeric.
signed 0x03	Input is numeric, with sign allowed.

decimal 0x05 Input is numeric, with decimals allowed.

This corresponds to the global attribute resource symbol [numeric](#)  
[\(/reference/android/R.attr.html#numeric\)](#).

Constant Value: 40 (0x00000028)

#### public static final int **TextView\_password**

Whether the characters of the field are displayed as password dots instead of themselves.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [password](#)  
[\(/reference/android/R.attr.html#password\)](#).

Constant Value: 31 (0x0000001f)

#### public static final int **TextView\_phoneNumber**

If set, specifies that this TextView has a phone number input method. The default is false.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [phoneNumber](#)  
[\(/reference/android/R.attr.html#phoneNumber\)](#).

Constant Value: 42 (0x0000002a)

#### public static final int **TextView\_privateImeOptions**

An addition content type description to supply to the input method attached to the text view, which is private to the implementation of the input method. This simply fills in the [EditorInfo.privateImeOptions](#)

[\(/reference/android/view/inputmethod/EditorInfo.html#privateImeOptions\)](#) field when the input method is connected.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [privateImeOptions](#)  
[\(/reference/android/R.attr.html#privateImeOptions\)](#).

Constant Value: 57 (0x00000039)

#### public static final int **TextView\_scrollHorizontally**

Whether the text is allowed to be wider than the view (and therefore can be scrolled horizontally).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [scrollHorizontally](#)  
[\(/reference/android/R.attr.html#scrollHorizontally\)](#).

Constant Value: 30 (0x0000001e)

#### public static final int **TextView\_selectAllOnFocus**

If the text is selectable, select it all when the view takes focus.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [selectAllOnFocus](#)  
[\(/reference/android/R.attr.html#selectAllOnFocus\)](#).

Constant Value: 33 (0x00000021)

**public static final int TextView\_shadowColor**

Place a shadow of the specified color behind the text.

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [shadowColor](#)  
([/reference/android/R.attr.html#shadowColor](#)).

Constant Value: 36 (0x00000024)

**public static final int TextView\_shadowDx**

Horizontal offset of the shadow.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [shadowDx](#)  
([/reference/android/R.attr.html#shadowDx](#)).

Constant Value: 37 (0x00000025)

**public static final int TextView\_shadowDy**

Vertical offset of the shadow.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [shadowDy](#)  
([/reference/android/R.attr.html#shadowDy](#)).

Constant Value: 38 (0x00000026)

**public static final int TextView\_shadowRadius**

Radius of the shadow.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [shadowRadius](#)  
([/reference/android/R.attr.html#shadowRadius](#)).

Constant Value: 39 (0x00000027)

**public static final int TextView\_singleLine**

Constrains the text to a single horizontally scrolling line instead of letting it wrap onto multiple lines, and advances focus instead of inserting a newline when you press the enter key. The default value is false (multi-line wrapped text mode) for non-editable text, but if you specify any value for inputType, the default is true (single-line input field mode).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [singleLine](#)  
([/reference/android/R.attr.html#singleLine](#)).

Constant Value: 32 (0x00000020)

**public static final int TextView\_text**

Text to display.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[*package*:]type:*name*") or theme attribute (in the form "?[*package*:][type:*name*]" containing a value of this type.

This corresponds to the global attribute resource symbol [text \(/reference/android/R.attr.html#text\)](#).

Constant Value: 18 (0x00000012)

#### public static final int **TextView\_textAllCaps**

Present the text in ALL CAPS. This may use a small-caps form when available.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package*:]type:*name*") or theme attribute (in the form "?[*package*:][type:*name*]" containing a value of this type.

This corresponds to the global attribute resource symbol [textAllCaps \(/reference/android/R.attr.html#textAllCaps\)](#).

Constant Value: 72 (0x00000048)

#### public static final int **TextView\_textAppearance**

Base text color, typeface, size, and style.

Must be a reference to another resource, in the form "@[+][*package*:]type:*name*" or to a theme attribute in the form "?[*package*:][type:*name*]".

This corresponds to the global attribute resource symbol [textAppearance \(/reference/android/R.attr.html#textAppearance\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **TextView\_textColor**

Text color.

May be a reference to another resource, in the form "@[+][*package*:]type:*name*" or to a theme attribute in the form "?[*package*:][type:*name*]".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColor \(/reference/android/R.attr.html#textColor\)](#).

Constant Value: 5 (0x00000005)

#### public static final int **TextView\_textColorHighlight**

Color of the text selection highlight.

May be a reference to another resource, in the form "@[+][*package*:]type:*name*" or to a theme attribute in the form "?[*package*:][type:*name*]".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorHighlight \(/reference/android/R.attr.html#textColorHighlight\)](#).

Constant Value: 6 (0x00000006)

#### public static final int **TextView\_textColorHint**

Color of the hint text.

May be a reference to another resource, in the form "@[+][*package*:]type:*name*" or to a theme attribute in the form "?[*package*:][type:*name*]".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorHint \(/reference/android/R.attr.html#textColorHint\)](#).

Constant Value: 7 (0x00000007)

#### public static final int **TextView\_textColorLink**

Text color for links.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorLink](#)  
(/reference/android/R.attr.html#textColorLink).

Constant Value: 8 (0x00000008)

#### **public static final int TextView\_textCursorDrawable**

Reference to a drawable that will be drawn under the insertion cursor.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [textCursorDrawable](#)  
(/reference/android/R.attr.html#textCursorDrawable).

Constant Value: 70 (0x00000046)

#### **public static final int TextView\_textEditNoPasteWindowLayout**

Variation of editTextPasteWindowLayout displayed when the clipboard is empty.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [textEditNoPasteWindowLayout](#)  
(/reference/android/R.attr.html#textEditNoPasteWindowLayout).

Constant Value: 66 (0x00000042)

#### **public static final int TextView\_textEditPasteWindowLayout**

The layout of the view that is displayed on top of the cursor to paste inside a EditText field.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [textEditPasteWindowLayout](#)  
(/reference/android/R.attr.html#textEditPasteWindowLayout).

Constant Value: 65 (0x00000041)

#### **public static final int TextView\_textEditSideNoPasteWindowLayout**

Variation of editTextSidePasteWindowLayout displayed when the clipboard is empty.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [textEditSideNoPasteWindowLayout](#)  
(/reference/android/R.attr.html#textEditSideNoPasteWindowLayout).

Constant Value: 69 (0x00000045)

#### **public static final int TextView\_textEditSidePasteWindowLayout**

Used instead of editTextPasteWindowLayout when the window is moved on the side of the insertion cursor because it would be clipped if it were positioned on top.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [textEditSidePasteWindowLayout](#)  
(/reference/android/R.attr.html#textEditSidePasteWindowLayout).

Constant Value: 68 (0x00000044)

#### **public static final int TextView\_textEditSuggestionItemLayout**

Layout of the TextView item that will populate the suggestion popup window.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute

in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [textEditSuggestionItemLayout](#) ([/reference/android/R.attr.html#textEditSuggestionItemLayout](#)).

Constant Value: 71 (0x00000047)

#### public static final int **TextView\_textIsSelectable**

Indicates that the content of a non-editable text can be selected.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "?[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [textIsSelectable](#) ([/reference/android/R.attr.html#textIsSelectable](#)).

Constant Value: 67 (0x00000043)

#### public static final int **TextView\_textScaleX**

Sets the horizontal scaling factor for the text.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "?[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [textScaleX](#) ([/reference/android/R.attr.html#textScaleX](#)).

Constant Value: 20 (0x00000014)

#### public static final int **TextView\_textSelectHandle**

Reference to a drawable that will be used to display a text selection anchor for positioning the cursor within text.

Must be a reference to another resource, in the form "?[ + ][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [textSelectHandle](#) ([/reference/android/R.attr.html#textSelectHandle](#)).

Constant Value: 64 (0x00000040)

#### public static final int **TextView\_textSelectHandleLeft**

Reference to a drawable that will be used to display a text selection anchor on the left side of a selection region.

Must be a reference to another resource, in the form "?[ + ][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [textSelectHandleLeft](#) ([/reference/android/R.attr.html#textSelectHandleLeft](#)).

Constant Value: 62 (0x0000003e)

#### public static final int **TextView\_textSelectHandleRight**

Reference to a drawable that will be used to display a text selection anchor on the right side of a selection region.

Must be a reference to another resource, in the form "?[ + ][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [textSelectHandleRight](#) ([/reference/android/R.attr.html#textSelectHandleRight](#)).

Constant Value: 63 (0x0000003f)

#### public static final int **TextView\_textSize**

Size of the text. Recommended dimension type for text is "sp" for scaled-pixels (example: 15sp).

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [textSize](#)  
(/reference/android/R.attr.html#textSize).

Constant Value: 2 (0x00000002)

#### public static final int **TextView\_textStyle**

Style (bold, italic, bolditalic) for the text.

Must be one or more (separated by '|') of the following constant values.

##### Constant Value Description

normal	0
bold	1
italic	2

This corresponds to the global attribute resource symbol [textStyle](#)  
(/reference/android/R.attr.html#textStyle).

Constant Value: 4 (0x00000004)

#### public static final int **TextView\_typeface**

Typeface (normal, sans, serif, monospace) for the text.

Must be one of the following constant values.

##### Constant Value Description

normal	0
sans	1
serif	2
monospace	3

This corresponds to the global attribute resource symbol [typeface](#)  
(/reference/android/R.attr.html#typeface).

Constant Value: 3 (0x00000003)

#### public static final int **TextView\_width**

Makes the TextView be exactly this many pixels wide. You could get the same effect by specifying this number in the layout parameters.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [width](#)  
(/reference/android/R.attr.html#width).

Constant Value: 28 (0x0000001c)

#### public static final int **Theme\_absListViewStyle**

Default AbsListView style.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [absListViewStyle](#)  
(/reference/android/R.attr.html#absListViewStyle).

Constant Value: 51 (0x00000033)

#### public static final int **Theme\_actionBarDivider**

Custom divider drawable to use for elements in the action bar.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [\\_ actionBarDivider](#)  
(/reference/android/R.attr.html#actionBarDivider).

Constant Value: 204 (0x000000cc)

#### public static final int **Theme\_actionBarItemBackground**

Custom item state list drawable background for action bar items.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [\\_ actionBarItemBackground](#)  
(/reference/android/R.attr.html#actionBarItemBackground).

Constant Value: 205 (0x000000cd)

#### public static final int **Theme\_actionBarSize**

Size of the Action Bar, including the contextual bar used to present Action Modes.

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*][*type:*]*name*") containing a value of this type.

May be one of the following constant values.

Constant	Value	Description
----------	-------	-------------

wrap\_content 0

This corresponds to the global attribute resource symbol [\\_ actionBarSize](#)  
(/reference/android/R.attr.html#actionBarSize).

Constant Value: 140 (0x0000008c)

#### public static final int **Theme\_actionBarSplitStyle**

Reference to a style for the split Action Bar. This style controls the split component that holds the menu/action buttons. actionBarStyle is still used for the primary bar.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [\\_ actionBarSplitStyle](#)  
(/reference/android/R.attr.html#actionBarSplitStyle).

Constant Value: 196 (0x000000c4)

#### public static final int **Theme\_actionBarStyle**

Reference to a style for the Action Bar

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [\\_ actionBarStyle](#)  
(/reference/android/R.attr.html#actionBarStyle).

Constant Value: 132 (0x00000084)

#### public static final int **Theme\_actionBarTabBarStyle**

This symbol is the offset where the [\\_ actionBarTabBarStyle](#)  
(/reference/android/R.attr.html#actionBarTabBarStyle) attribute's value can be found in the [\\_ Theme](#)  
(/reference/android/R.styleable.html#Theme) array.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

Constant Value: 143 (0x0000008f)

#### public static final int **Theme\_actionBarTabStyle**

Default style for tabs within an action bar

Must be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:]type:name`".

This corresponds to the global attribute resource symbol  [actionBarTabStyle](#)  
[\(/reference/android/R.attr.html#actionBarTabStyle\)](#).

Constant Value: 142 (0x0000008e)

#### **public static final int Theme\_actionBarTabTextStyle**

This symbol is the offset where the  [actionBarTabTextStyle](#)  
[\(/reference/android/R.attr.html#actionBarTabTextStyle\)](#) attribute's value can be found in the [Theme](#)  
[\(/reference/android/R.styleable.html#Theme\)](#) array.

Must be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:]type:name`".

Constant Value: 144 (0x00000090)

#### **public static final int Theme\_actionBarWidgetTheme**

Reference to a theme that should be used to inflate widgets and layouts destined for the action bar. Most of the time this will be a reference to the current theme, but when the action bar has a significantly different contrast profile than the rest of the activity the difference can become important. If this is set to @null the current theme will be used.

Must be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:]type:name`".

This corresponds to the global attribute resource symbol  [actionBarWidgetTheme](#)  
[\(/reference/android/R.attr.html#actionBarWidgetTheme\)](#).

Constant Value: 203 (0x000000cb)

#### **public static final int ThemeActionButtonStyle**

Default action button style.

Must be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:]type:name`".

This corresponds to the global attribute resource symbol  [actionBarWidgetTheme](#)  
[\(/reference/android/R.attr.html#actionButtonStyle\)](#).

Constant Value: 135 (0x00000087)

#### **public static final int Theme\_actionDropDownStyle**

Default ActionBar dropdown style.

Must be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:]type:name`".

This corresponds to the global attribute resource symbol  [actionBarWidgetTheme](#)  
[\(/reference/android/R.attr.html#actionDropDownStyle\)](#).

Constant Value: 134 (0x00000086)

#### **public static final int Theme\_actionMenuTextAppearance**

TextAppearance style that will be applied to text that appears within action menu items.

Must be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:]type:name`".

This corresponds to the global attribute resource symbol  [actionBarWidgetTheme](#)  
[\(/reference/android/R.attr.html#actionMenuTextAppearance\)](#).

Constant Value: 188 (0x000000bc)

#### **public static final int Theme\_actionMenuTextColor**

Color for text that appears within action menu items.

May be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:]type:name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [actionMenuTextColor](#) ([/reference/android/R.attr.html#actionMenuTextColor](#)).

Constant Value: 189 (0x000000bd)

#### public static final int **Theme\_actionModeBackground**

Background drawable to use for action mode UI

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [actionModeBackground](#) ([/reference/android/R.attr.html#actionModeBackground](#)).

Constant Value: 136 (0x00000088)

#### public static final int **Theme\_actionModeCloseButtonStyle**

This symbol is the offset where the [actionModeCloseButtonStyle](#) ([/reference/android/R.attr.html#actionModeCloseButtonStyle](#)) attribute's value can be found in the [Theme](#) ([/reference/android/R.styleable.html#Theme](#)) array.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

Constant Value: 146 (0x00000092)

#### public static final int **Theme\_actionModeCloseDrawable**

Drawable to use for the close action mode button

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [actionModeCloseDrawable](#) ([/reference/android/R.attr.html#actionModeCloseDrawable](#)).

Constant Value: 137 (0x00000089)

#### public static final int **Theme\_actionModeCopyDrawable**

Drawable to use for the Copy action button in Contextual Action Bar

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [actionModeCopyDrawable](#) ([/reference/android/R.attr.html#actionModeCopyDrawable](#)).

Constant Value: 160 (0x000000a0)

#### public static final int **Theme\_actionModeCutDrawable**

Drawable to use for the Cut action button in Contextual Action Bar

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [actionModeCutDrawable](#) ([/reference/android/R.attr.html#actionModeCutDrawable](#)).

Constant Value: 159 (0x0000009f)

#### public static final int **Theme\_actionModePasteDrawable**

Drawable to use for the Paste action button in Contextual Action Bar

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [actionModePasteDrawable](#) ([/reference/android/R.attr.html#actionModePasteDrawable](#)).

Constant Value: 161 (0x000000a1)

**public static final int Theme\_actionModeSelectAllDrawable**

Drawable to use for the Select all action button in Contextual Action Bar

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [actionModeSelectAllDrawable](#)  
[\(/reference/android/R.attr.html#actionModeSelectAllDrawable\)](#).

Constant Value: 193 (0x000000c1)

**public static final int Theme\_actionModeSplitBackground**

Background drawable to use for action mode UI in the lower split bar

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [actionModeSplitBackground](#)  
[\(/reference/android/R.attr.html#actionModeSplitBackground\)](#).

Constant Value: 206 (0x000000ce)

**public static final int Theme\_actionModeStyle**

This symbol is the offset where the [actionModeStyle](#)  
[\(/reference/android/R.attr.html#actionModeStyle\)](#) attribute's value can be found in the [Theme](#)  
[\(/reference/android/R.styleable.html#Theme\)](#) array.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

Constant Value: 202 (0x000000ca)

**public static final int Theme\_overflowButtonStyle**

This symbol is the offset where the [actionOverflowButtonStyle](#)  
[\(/reference/android/R.attr.html#actionOverflowButtonStyle\)](#) attribute's value can be found in the  
[Theme](#) [\(/reference/android/R.styleable.html#Theme\)](#) array.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

Constant Value: 145 (0x00000091)

**public static final int Theme\_activatedBackgroundIndicator**

Drawable used as a background for activated items.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [activatedBackgroundIndicator](#)  
[\(/reference/android/R.attr.html#activatedBackgroundIndicator\)](#).

Constant Value: 147 (0x00000093)

**public static final int Theme\_alertDialogIcon**

Icon drawable to use for alerts

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [alertDialogIcon](#)  
[\(/reference/android/R.attr.html#alertDialogIcon\)](#).

Constant Value: 181 (0x000000b5)

**public static final int Theme\_alertDialogStyle**

This symbol is the offset where the [alertDialogStyle](#)  
[\(/reference/android/R.attr.html#alertDialogStyle\)](#) attribute's value can be found in the [Theme](#)  
[\(/reference/android/R.styleable.html#Theme\)](#) array.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute

in the form "?[ package: ][ type: ]name".

Constant Value: 45 (0x0000002d)

#### public static final int **Theme\_alertDialogTheme**

Theme to use for alert dialogs spawned from this theme.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol  [alertDialogTheme  
\(/reference/android/R.attr.html#alertDialogTheme\)](#).

Constant Value: 155 (0x0000009b)

#### public static final int **Theme\_autoCompleteTextViewStyle**

Default AutoCompleteTextView style.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol  [autoCompleteTextViewStyle  
\(/reference/android/R.attr.html#autoCompleteTextViewStyle\)](#).

Constant Value: 52 (0x00000034)

#### public static final int **Theme\_backgroundDimAmount**

Default background dim amount when a menu, dialog, or something similar pops up.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol  [backgroundDimAmount  
\(/reference/android/R.attr.html#backgroundDimAmount\)](#).

Constant Value: 2 (0x00000002)

#### public static final int **Theme\_backgroundDimEnabled**

Control whether dimming behind the window is enabled. The default theme does not set this value, meaning it is based on whether the window is floating.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol  [backgroundDimEnabled  
\(/reference/android/R.attr.html#backgroundDimEnabled\)](#).

Constant Value: 106 (0x0000006a)

#### public static final int **Theme\_borderlessButtonStyle**

Style for buttons without an explicit border, often used in groups.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol  [borderlessButtonStyle  
\(/reference/android/R.attr.html#borderlessButtonStyle\)](#).

Constant Value: 165 (0x000000a5)

#### public static final int **Theme\_buttonBarButtonStyle**

Style for buttons within button bars

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol  [buttonBarButtonStyle  
\(/reference/android/R.attr.html#buttonBarButtonStyle\)](#).

Constant Value: 168 (0x000000a8)

#### **public static final int Theme\_buttonBarStyle**

Style for button bars

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [buttonBarStyle](#)  
(/reference/android/R.attr.html#buttonBarStyle).

Constant Value: 167 (0x000000a7)

#### **public static final int Theme\_buttonStyle**

Normal Button style.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [buttonStyle](#)  
(/reference/android/R.attr.html#buttonStyle).

Constant Value: 24 (0x00000018)

#### **public static final int Theme\_buttonStyleInset**

Button style to inset into an EditText.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [buttonStyleInset](#)  
(/reference/android/R.attr.html#buttonStyleInset).

Constant Value: 26 (0x0000001a)

#### **public static final int Theme\_buttonStyleSmall**

Small Button style.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [buttonStyleSmall](#)  
(/reference/android/R.attr.html#buttonStyleSmall).

Constant Value: 25 (0x00000019)

#### **public static final int Theme\_buttonStyleToggle**

ToggleButton style.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [buttonStyleToggle](#)  
(/reference/android/R.attr.html#buttonStyleToggle).

Constant Value: 27 (0x0000001b)

#### **public static final int Theme\_calendarViewStyle**

The CalendarView style.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [calendarViewStyle](#)  
(/reference/android/R.attr.html#calendarViewStyle).

Constant Value: 185 (0x000000b9)

#### **public static final int Theme\_candidatesTextStyleSpans**

A styled string, specifying the style to be used for showing inline candidate text when composing with an

input method. The text itself will be ignored, but the style spans will be applied to the candidate text as it is edited.

May be a reference to another resource, in the form "`@[+][ package: ]type: name`" or to a theme attribute in the form "`?[ package: ][ type: ]name`".

May be a string value, using '\\' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This corresponds to the global attribute resource symbol [candidatesTextStyleSpans](#)  
[\(/reference/android/R.attr.html#candidatesTextStyleSpans\)](#).

Constant Value: 109 (0x0000006d)

#### **public static final int Theme\_checkBoxPreferenceStyle**

Default style for CheckBoxPreference.

Must be a reference to another resource, in the form "`@[+][ package: ]type: name`" or to a theme attribute in the form "`?[ package: ][ type: ]name`".

This corresponds to the global attribute resource symbol [checkBoxPreferenceStyle](#)  
[\(/reference/android/R.attr.html#checkBoxPreferenceStyle\)](#).

Constant Value: 87 (0x00000057)

#### **public static final int Theme\_checkboxStyle**

Default Checkbox style.

Must be a reference to another resource, in the form "`@[+][ package: ]type: name`" or to a theme attribute in the form "`?[ package: ][ type: ]name`".

This corresponds to the global attribute resource symbol [checkboxStyle](#)  
[\(/reference/android/R.attr.html#checkboxStyle\)](#).

Constant Value: 53 (0x00000035)

#### **public static final int Theme\_checkedTextViewStyle**

Default CheckedTextView style.

Must be a reference to another resource, in the form "`@[+][ package: ]type: name`" or to a theme attribute in the form "`?[ package: ][ type: ]name`".

This corresponds to the global attribute resource symbol [checkedTextViewStyle](#)  
[\(/reference/android/R.attr.html#checkedTextViewStyle\)](#).

Constant Value: 217 (0x000000d9)

#### **public static final int Theme\_colorActivatedHighlight**

Default highlight color for items that are activated. (Activated meaning persistent selection.)

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "`@[ package: ]type: name`") or theme attribute (in the form "`?[ package: ][ type: ]name`") containing a value of this type.

This corresponds to the global attribute resource symbol [colorActivatedHighlight](#)  
[\(/reference/android/R.attr.html#colorActivatedHighlight\)](#).

Constant Value: 200 (0x000000c8)

#### **public static final int Theme\_colorBackground**

Color that matches (as closely as possible) the window background.

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "`@[ package: ]type: name`") or theme attribute (in the form "`?[ package: ][ type: ]name`") containing a value of this type.

This corresponds to the global attribute resource symbol [colorBackground](#)  
[\(/reference/android/R.attr.html#colorBackground\)](#).

Constant Value: 1 (0x00000001)

#### **public static final int Theme\_colorBackgroundCacheHint**

This is a hint for a solid color that can be used for caching rendered views. This should be the color of the background when there is a solid background color; it should be null when the background is a texture or translucent. When a device is able to use accelerated drawing (thus setting state\_accelerated), the cache hint is ignored and always assumed to be transparent.

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [colorBackgroundCacheHint](#) ([/reference/android/R.attr.html#colorBackgroundCacheHint](#)).

Constant Value: 118 (0x000000076)

#### public static final int **Theme\_colorFocusedHighlight**

Default highlight color for items that are focused. (Focused meaning cursor-based selection.)

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [colorFocusedHighlight](#) ([/reference/android/R.attr.html#colorFocusedHighlight](#)).

Constant Value: 199 (0x0000000c7)

#### public static final int **Theme\_colorForeground**

Default color of foreground imagery.

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [colorForeground](#) ([/reference/android/R.attr.html#colorForeground](#)).

Constant Value: 0 (0x00000000)

#### public static final int **Theme\_colorForegroundInverse**

Default color of foreground imagery on an inverted background.

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [colorForegroundInverse](#) ([/reference/android/R.attr.html#colorForegroundInverse](#)).

Constant Value: 94 (0x00000005e)

#### public static final int **Theme\_colorLongPressedHighlight**

Default highlight color for items that are long-pressed.

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [colorLongPressedHighlight](#) ([/reference/android/R.attr.html#colorLongPressedHighlight](#)).

Constant Value: 198 (0x0000000c6)

#### public static final int **Theme\_colorMultiSelectHighlight**

Default highlight color for items in multiple selection mode.

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [colorMultiSelectHighlight](#)  
[\(/reference/android/R.attr.html#colorMultiSelectHighlight\)](#).

Constant Value: 201 (0x000000c9)

#### public static final int **Theme\_colorPressedHighlight**

Default highlight color for items that are pressed.

Must be a color value, in the form "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [colorPressedHighlight](#)  
[\(/reference/android/R.attr.html#colorPressedHighlight\)](#).

Constant Value: 197 (0x000000c5)

#### public static final int **Theme\_datePickerStyle**

The DatePicker style.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [datePickerStyle](#)  
[\(/reference/android/R.attr.html#datePickerStyle\)](#).

Constant Value: 184 (0x000000b8)

#### public static final int **Theme\_detailsElementBackground**

Background that can be used behind parts of a UI that provide details on data the user is selecting. For example, this is the background element of PreferenceActivity's embedded preference fragment.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [detailsElementBackground](#)  
[\(/reference/android/R.attr.html#detailsElementBackground\)](#).

Constant Value: 175 (0x000000af)

#### public static final int **Theme\_dialogPreferenceStyle**

Default style for DialogPreference.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [dialogPreferenceStyle](#)  
[\(/reference/android/R.attr.html#dialogPreferenceStyle\)](#).

Constant Value: 89 (0x00000059)

#### public static final int **Theme\_dialogTheme**

Theme to use for dialogs spawned from this theme.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [dialogTheme](#)  
[\(/reference/android/R.attr.html#dialogTheme\)](#).

Constant Value: 154 (0x0000009a)

#### public static final int **Theme\_disabledAlpha**

Default disabled alpha for widgets that set enabled/disabled alpha programmatically.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [disabledAlpha](#)

[\(/reference/android/R.attr.html#disabledAlpha\)](#).

Constant Value: 3 (0x00000003)

#### public static final int **Theme\_dividerHorizontal**

Drawable to use for generic horizontal dividers.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [dividerHorizontal](#)  
[\(/reference/android/R.attr.html#dividerHorizontal\)](#).

Constant Value: 166 (0x000000a6)

#### public static final int **Theme\_dividerVertical**

Drawable to use for generic vertical dividers.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [dividerVertical](#)  
[\(/reference/android/R.attr.html#dividerVertical\)](#).

Constant Value: 156 (0x0000009c)

#### public static final int **Theme\_dropDownHintAppearance**

Default style for drop down hints.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [dropDownHintAppearance](#)  
[\(/reference/android/R.attr.html#dropDownHintAppearance\)](#).

Constant Value: 80 (0x00000050)

#### public static final int **Theme\_dropDownItemStyle**

Default style for drop down items.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [dropDownItemStyle](#)  
[\(/reference/android/R.attr.html#dropDownItemStyle\)](#).

Constant Value: 78 (0x0000004e)

#### public static final int **Theme\_dropDownListviewStyle**

Default ListView style for drop downs.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [dropDownListviewStyle](#)  
[\(/reference/android/R.attr.html#dropDownListviewStyle\)](#).

Constant Value: 54 (0x00000036)

#### public static final int **Theme\_dropDownSpinnerStyle**

Default dropdown Spinner style.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [dropDownSpinnerStyle](#)  
[\(/reference/android/R.attr.html#dropDownSpinnerStyle\)](#).

Constant Value: 133 (0x00000085)

#### public static final int **Theme\_editTextBackground**

EditText background drawable.

Must be a reference to another resource, in the form "`@[+][package: ]type: name`" or to a theme attribute in the form "`?[package: ][type: ]name`".

This corresponds to the global attribute resource symbol [editTextBackground](#)  
[\(/reference/android/R.attr.html#editTextBackground\)](#).

Constant Value: 179 (0x000000b3)

#### **public static final int Theme\_editTextColor**

EditText text foreground color.

May be a reference to another resource, in the form "`@[+][package: ]type: name`" or to a theme attribute in the form "`?[package: ][type: ]name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [editTextColor](#)  
[\(/reference/android/R.attr.html#editTextColor\)](#).

Constant Value: 178 (0x000000b2)

#### **public static final int Theme\_editTextPreferenceStyle**

Default style for EditTextPreference.

Must be a reference to another resource, in the form "`@[+][package: ]type: name`" or to a theme attribute in the form "`?[package: ][type: ]name`".

This corresponds to the global attribute resource symbol [editTextPreferenceStyle](#)  
[\(/reference/android/R.attr.html#editTextPreferenceStyle\)](#).

Constant Value: 90 (0x0000005a)

#### **public static final int Theme\_editTextStyle**

Default EditText style.

Must be a reference to another resource, in the form "`@[+][package: ]type: name`" or to a theme attribute in the form "`?[package: ][type: ]name`".

This corresponds to the global attribute resource symbol [editTextStyle](#)  
[\(/reference/android/R.attr.html#editTextStyle\)](#).

Constant Value: 55 (0x00000037)

#### **public static final int Theme\_expandableListPreferredChildIndicatorLeft**

The preferred left bound for an expandable list child's indicator.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "`@[ package: ]type: name`") or theme attribute (in the form "`?[ package: ][type: ]name`") containing a value of this type.

This corresponds to the global attribute resource symbol [expandableListPreferredChildIndicatorLeft](#)  
[\(/reference/android/R.attr.html#expandableListPreferredChildIndicatorLeft\)](#).

Constant Value: 34 (0x00000022)

#### **public static final int Theme\_expandableListPreferredChildIndicatorRight**

The preferred right bound for an expandable list child's indicator.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "`@[ package: ]type: name`") or theme attribute (in the form "`?[ package: ][type: ]name`") containing a value of this type.

This corresponds to the global attribute resource symbol [expandableListPreferredChildIndicatorRight](#)

[\(/reference/android/R.attr.html#expandableListPreferredChildIndicatorRight\).](#)

Constant Value: 35 (0x00000023)

#### **public static final int Theme\_expandableListPreferredChildPaddingLeft**

The preferred left padding for an expandable list item that is a child. If this is not provided, it defaults to the expandableListPreferredItemPaddingLeft.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol

[expandableListPreferredChildPaddingLeft](#)

[\(/reference/android/R.attr.html#expandableListPreferredChildPaddingLeft\).](#)

Constant Value: 31 (0x0000001f)

#### **public static final int Theme\_expandableListPreferredItemIndicatorLeft**

The preferred left bound for an expandable list item's indicator. For a child-specific indicator, use expandableListPreferredChildIndicatorLeft.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol

[expandableListPreferredItemIndicatorLeft](#)

[\(/reference/android/R.attr.html#expandableListPreferredItemIndicatorLeft\).](#)

Constant Value: 32 (0x00000020)

#### **public static final int Theme\_expandableListPreferredItemIndicatorRight**

The preferred right bound for an expandable list item's indicator. For a child-specific indicator, use expandableListPreferredChildIndicatorRight.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol

[expandableListPreferredItemIndicatorRight](#)

[\(/reference/android/R.attr.html#expandableListPreferredItemIndicatorRight\).](#)

Constant Value: 33 (0x00000021)

#### **public static final int Theme\_expandableListPreferredItemPaddingLeft**

The preferred left padding for an expandable list item (for child-specific layouts, use expandableListPreferredChildPaddingLeft). This takes into account the indicator that will be shown to next to the item.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [expandableListPreferredItemPaddingLeft](#)

[\(/reference/android/R.attr.html#expandableListPreferredItemPaddingLeft\).](#)

Constant Value: 30 (0x0000001e)

#### **public static final int Theme\_expandableListViewStyle**

Default ExpandableListView style.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [expandableListViewStyle](#)  
(/reference/android/R.attr.html#expandableListViewStyle).

Constant Value: 56 (0x00000038)

#### public static final int **Theme\_expandableListViewWhiteStyle**

ExpandableListView with white background.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [expandableListViewWhiteStyle](#)  
(/reference/android/R.attr.html#expandableListViewWhiteStyle).

Constant Value: 125 (0x0000007d)

#### public static final int **Theme\_fastScrollOverlayPosition**

Position of the fast scroll index overlay window.

Must be one of the following constant values.

##### **Constant Value Description**

floating 0

atThumb 1

This corresponds to the global attribute resource symbol [fastScrollOverlayPosition](#)  
(/reference/android/R.attr.html#fastScrollOverlayPosition).

Constant Value: 174 (0x000000ae)

#### public static final int **Theme\_fastScrollPreviewBackgroundLeft**

Drawable to use as the fast scroll index preview window background when shown on the left.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [fastScrollPreviewBackgroundLeft](#)  
(/reference/android/R.attr.html#fastScrollPreviewBackgroundLeft).

Constant Value: 171 (0x000000ab)

#### public static final int **Theme\_fastScrollPreviewBackgroundRight**

Drawable to use as the fast scroll index preview window background when shown on the right.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [fastScrollPreviewBackgroundRight](#)  
(/reference/android/R.attr.html#fastScrollPreviewBackgroundRight).

Constant Value: 172 (0x000000ac)

#### public static final int **Theme\_fastScrollTextColor**

Text color for the fast scroll index overlay. Make sure it plays nicely with fastScrollPreviewBackground[Left|Right].

Must be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This may also be a reference to a resource (in the form "@[ package: ]type:name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [fastScrollTextColor](#)  
(/reference/android/R.attr.html#fastScrollTextColor).

Constant Value: 182 (0x000000b6)

#### public static final int **Theme\_fastScrollThumbDrawable**

Drawable to use as the fast scroll thumb.

Must be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

This corresponds to the global attribute resource symbol [fastScrollThumbDrawable](#)  
[\(/reference/android/R.attr.html#fastScrollThumbDrawable\)](#).

Constant Value: 170 (0x000000aa)

#### **public static final int Theme\_fastScrollTrackDrawable**

Drawable to use as the track for the fast scroll thumb. This may be null.

Must be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

This corresponds to the global attribute resource symbol [fastScrollTrackDrawable](#)  
[\(/reference/android/R.attr.html#fastScrollTrackDrawable\)](#).

Constant Value: 173 (0x000000ad)

#### **public static final int Theme\_galleryItemBackground**

The preferred background for gallery items. This should be set as the background of any Views you provide from the Adapter.

Must be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

This corresponds to the global attribute resource symbol [galleryItemBackground](#)  
[\(/reference/android/R.attr.html#galleryItemBackground\)](#).

Constant Value: 28 (0x0000001c)

#### **public static final int Theme\_galleryStyle**

Default Gallery style.

Must be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

This corresponds to the global attribute resource symbol [galleryStyle](#)  
[\(/reference/android/R.attr.html#galleryStyle\)](#).

Constant Value: 57 (0x00000039)

#### **public static final int Theme\_gridViewStyle**

Default GridView style.

Must be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

This corresponds to the global attribute resource symbol [gridViewStyle](#)  
[\(/reference/android/R.attr.html#gridViewStyle\)](#).

Constant Value: 58 (0x0000003a)

#### **public static final int Theme\_homeAsUpIndicator**

Specifies a drawable to use for the 'home as up' indicator.

Must be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

This corresponds to the global attribute resource symbol [homeAsUpIndicator](#)  
[\(/reference/android/R.attr.html#homeAsUpIndicator\)](#).

Constant Value: 157 (0x0000009d)

#### **public static final int Theme\_horizontalScrollViewStyle**

Default HorizontalScrollView style.

Must be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

This corresponds to the global attribute resource symbol [horizontalScrollViewStyle](#)  
(/reference/android/R.attr.html#horizontalScrollViewStyle).

Constant Value: 180 (0x000000b4)

#### public static final int **Theme\_imageButtonStyle**

The style resource to use for an ImageButton.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [imageButtonStyle](#)  
(/reference/android/R.attr.html#imageButtonStyle).

Constant Value: 59 (0x0000003b)

#### public static final int **Theme\_imageWellStyle**

The style resource to use for an ImageButton that is an image well.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [imageWellStyle](#)  
(/reference/android/R.attr.html#imageWellStyle).

Constant Value: 60 (0x0000003c)

#### public static final int **Theme\_listChoiceBackgroundIndicator**

Drawable used as a background for selected list items.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [listChoiceBackgroundIndicator](#)  
(/reference/android/R.attr.html#listChoiceBackgroundIndicator).

Constant Value: 141 (0x0000008d)

#### public static final int **Theme\_listChoiceIndicatorMultiple**

Drawable to use for multiple choice indicators.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [listChoiceIndicatorMultiple](#)  
(/reference/android/R.attr.html#listChoiceIndicatorMultiple).

Constant Value: 104 (0x00000068)

#### public static final int **Theme\_listChoiceIndicatorSingle**

Drawable to use for single choice indicators.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [listChoiceIndicatorSingle](#)  
(/reference/android/R.attr.html#listChoiceIndicatorSingle).

Constant Value: 103 (0x00000067)

#### public static final int **Theme\_listDivider**

The drawable for the list divider.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [listDivider](#)  
(/reference/android/R.attr.html#listDivider).

Constant Value: 102 (0x00000066)

## **public static final int Theme\_listDividerAlertDialog**

The list divider used in alert dialogs.

Must be a reference to another resource, in the form "`@[+][ package: ]type: name`" or to a theme attribute in the form "`?[ package: ][ type: ]name`".

This corresponds to the global attribute resource symbol [listDividerAlertDialog](#)  
[\(/reference/android/R.attr.html#listDividerAlertDialog\)](#).

Constant Value: 152 (0x00000098)

## **public static final int Theme\_listPopupWindowStyle**

Default ListPopupWindow style.

Must be a reference to another resource, in the form "`@[+][ package: ]type: name`" or to a theme attribute in the form "`?[ package: ][ type: ]name`".

This corresponds to the global attribute resource symbol [listPopupWindowStyle](#)  
[\(/reference/android/R.attr.html#listPopupWindowStyle\)](#).

Constant Value: 148 (0x00000094)

## **public static final int Theme\_listPreferredItemHeight**

The preferred list item height.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "`@[ package: ]type: name`") or theme attribute (in the form "`?[ package: ][ type: ]name`") containing a value of this type.

This corresponds to the global attribute resource symbol [listPreferredItemHeight](#)  
[\(/reference/android/R.attr.html#listPreferredItemHeight\)](#).

Constant Value: 29 (0x0000001d)

## **public static final int Theme\_listPreferredItemHeightLarge**

A larger, more robust list item height.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "`@[ package: ]type: name`") or theme attribute (in the form "`?[ package: ][ type: ]name`") containing a value of this type.

This corresponds to the global attribute resource symbol [listPreferredItemHeightLarge](#)  
[\(/reference/android/R.attr.html#listPreferredItemHeightLarge\)](#).

Constant Value: 194 (0x000000c2)

## **public static final int Theme\_listPreferredItemHeightSmall**

A smaller, sleeker list item height.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "`@[ package: ]type: name`") or theme attribute (in the form "`?[ package: ][ type: ]name`") containing a value of this type.

This corresponds to the global attribute resource symbol [listPreferredItemHeightSmall](#)  
[\(/reference/android/R.attr.html#listPreferredItemHeightSmall\)](#).

Constant Value: 195 (0x000000c3)

## **public static final int Theme\_listPreferredItemPaddingEnd**

The preferred padding along the end edge of list items.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font

size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*:]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [listPreferredItemPaddingEnd](#)  
(/reference/android/R.attr.html#listPreferredItemPaddingEnd).

Constant Value: 214 (0x000000d6)

#### public static final int **Theme\_listPreferredItemPaddingLeft**

The preferred padding along the left edge of list items.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*:]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [listPreferredItemPaddingLeft](#)  
(/reference/android/R.attr.html#listPreferredItemPaddingLeft).

Constant Value: 209 (0x000000d1)

#### public static final int **Theme\_listPreferredItemPaddingRight**

The preferred padding along the right edge of list items.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*:]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [listPreferredItemPaddingRight](#)  
(/reference/android/R.attr.html#listPreferredItemPaddingRight).

Constant Value: 210 (0x000000d2)

#### public static final int **Theme\_listPreferredItemPaddingStart**

The preferred padding along the start edge of list items.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[*package*: ]*type*:*name*") or theme attribute (in the form "?[*package*: ][*type*:]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [listPreferredItemPaddingStart](#)  
(/reference/android/R.attr.html#listPreferredItemPaddingStart).

Constant Value: 213 (0x000000d5)

#### public static final int **Theme\_listSeparatorTextViewStyle**

TextView style for list separators.

Must be a reference to another resource, in the form "@[+][*package*: ]*type*:*name*" or to a theme attribute in the form "?[*package*: ][*type*:]*name*".

This corresponds to the global attribute resource symbol [listSeparatorTextViewStyle](#)  
(/reference/android/R.attr.html#listSeparatorTextViewStyle).

Constant Value: 96 (0x00000060)

#### public static final int **Theme\_listViewStyle**

Default ListView style.

Must be a reference to another resource, in the form "@[+][*package*: ]*type*:*name*" or to a theme attribute in the form "?[*package*: ][*type*:]*name*".

This corresponds to the global attribute resource symbol [listViewStyle](#)

[\(/reference/android/R.attr.html#listViewStyle\).](#)

Constant Value: 61 (0x0000003d)

#### public static final int **Theme\_listViewWhiteStyle**

ListView with white background.

Must be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:]type:name`".

This corresponds to the global attribute resource symbol [listViewWhiteStyle](#)

[\(/reference/android/R.attr.html#listViewWhiteStyle\).](#)

Constant Value: 62 (0x0000003e)

#### public static final int **Theme\_mapViewStyle**

Default MapView style.

Must be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:]type:name`".

This corresponds to the global attribute resource symbol [mapViewStyle](#)

[\(/reference/android/R.attr.html#mapViewStyle\).](#)

Constant Value: 82 (0x00000052)

#### public static final int **Theme\_mediaRouteButtonStyle**

Default style for the MediaRouteButton widget.

Must be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:]type:name`".

This corresponds to the global attribute resource symbol [mediaRouteButtonStyle](#)

[\(/reference/android/R.attr.html#mediaRouteButtonStyle\).](#)

Constant Value: 212 (0x000000d4)

#### public static final int **Theme\_panelBackground**

The background of a panel when it is inset from the left and right edges of the screen.

May be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:]type:name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [panelBackground](#)

[\(/reference/android/R.attr.html#panelBackground\).](#)

Constant Value: 46 (0x0000002e)

#### public static final int **Theme\_panelColorBackground**

Color that matches (as closely as possible) the panel background.

May be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:]type:name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [panelColorBackground](#)

[\(/reference/android/R.attr.html#panelColorBackground\).](#)

Constant Value: 49 (0x00000031)

#### public static final int **Theme\_panelColorForeground**

Default color of foreground panel imagery.

May be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:]type:name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [panelColorForeground](#)

[\(/reference/android/R.attr.html#panelColorForeground\)](#).

Constant Value: 48 (0x00000030)

#### **public static final int Theme.panelFullBackground**

The background of a panel when it extends to the left and right edges of the screen.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [panelFullBackground](#)  
[\(/reference/android/R.attr.html#panelFullBackground\)](#).

Constant Value: 47 (0x0000002f)

#### **public static final int Theme.panelTextAppearance**

Default appearance of panel text.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [panelTextAppearance](#)  
[\(/reference/android/R.attr.html#panelTextAppearance\)](#).

Constant Value: 50 (0x00000032)

#### **public static final int Theme.popupMenuStyle**

Default PopupMenu style.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [popupMenuStyle](#)  
[\(/reference/android/R.attr.html#popupMenuStyle\)](#).

Constant Value: 149 (0x00000095)

#### **public static final int Theme.popupWindowStyle**

Default PopupWindow style.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [popupWindowStyle](#)  
[\(/reference/android/R.attr.html#popupWindowStyle\)](#).

Constant Value: 63 (0x0000003f)

#### **public static final int Theme.preferenceCategoryStyle**

Default style for PreferenceCategory.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [preferenceCategoryStyle](#)  
[\(/reference/android/R.attr.html#preferenceCategoryStyle\)](#).

Constant Value: 84 (0x00000054)

#### **public static final int Theme.preferenceInformationStyle**

Default style for informational Preference.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [preferenceInformationStyle](#)  
[\(/reference/android/R.attr.html#preferenceInformationStyle\)](#).

Constant Value: 85 (0x00000055)

## **public static final int Theme\_preferenceLayoutChild**

The preference layout that has the child/tabbed effect.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [\\_preferenceLayoutChild](#)  
[\(/reference/android/R.attr.html#preferenceLayoutChild\)](#).

Constant Value: 92 (0x0000005c)

## **public static final int Theme\_preferenceScreenStyle**

Default style for PreferenceScreen.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [\\_preferenceScreenStyle](#)  
[\(/reference/android/R.attr.html#preferenceScreenStyle\)](#).

Constant Value: 83 (0x00000053)

## **public static final int Theme\_preferenceStyle**

Default style for Preference.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [\\_preferenceStyle](#)  
[\(/reference/android/R.attr.html#preferenceStyle\)](#).

Constant Value: 86 (0x00000056)

## **public static final int Theme\_presentationTheme**

Theme to use for presentations spawned from this theme.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [\\_presentationTheme](#)  
[\(/reference/android/R.attr.html#presentationTheme\)](#).

Constant Value: 215 (0x000000d7)

## **public static final int Theme\_progressBarStyle**

Default ProgressBar style. This is a medium circular progress bar.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [\\_progressBarStyle](#)  
[\(/reference/android/R.attr.html#progressBarStyle\)](#).

Constant Value: 64 (0x00000040)

## **public static final int Theme\_progressBarStyleHorizontal**

Horizontal ProgressBar style. This is a horizontal progress bar.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [\\_progressBarStyleHorizontal](#)  
[\(/reference/android/R.attr.html#progressBarStyleHorizontal\)](#).

Constant Value: 65 (0x00000041)

## **public static final int Theme\_progressBarStyleInverse**

Inverse ProgressBar style. This is a medium circular progress bar.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [progressBarStyleInverse](#)  
(/reference/android/R.attr.html#progressBarStyleInverse).

Constant Value: 111 (0x0000006f)

#### public static final int **Theme\_progressBarStyleLarge**

Large ProgressBar style. This is a large circular progress bar.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [progressBarStyleLarge](#)  
(/reference/android/R.attr.html#progressBarStyleLarge).

Constant Value: 67 (0x00000043)

#### public static final int **Theme\_progressBarStyleLargeInverse**

Large inverse ProgressBar style. This is a large circular progress bar.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [progressBarStyleLargeInverse](#)  
(/reference/android/R.attr.html#progressBarStyleLargeInverse).

Constant Value: 113 (0x00000071)

#### public static final int **Theme\_progressBarStyleSmall**

Small ProgressBar style. This is a small circular progress bar.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [progressBarStyleSmall](#)  
(/reference/android/R.attr.html#progressBarStyleSmall).

Constant Value: 66 (0x00000042)

#### public static final int **Theme\_progressBarStyleSmallInverse**

Small inverse ProgressBar style. This is a small circular progress bar.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [progressBarStyleSmallInverse](#)  
(/reference/android/R.attr.html#progressBarStyleSmallInverse).

Constant Value: 112 (0x00000070)

#### public static final int **Theme\_progressBarStyleSmallTitle**

Small ProgressBar in title style. This is a small circular progress bar that will be placed in title bars.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [progressBarStyleSmallTitle](#)  
(/reference/android/R.attr.html#progressBarStyleSmallTitle).

Constant Value: 98 (0x00000062)

#### public static final int **Theme\_quickContactBadgeStyleSmallWindowLarge**

Default quickcontact badge style with large quickcontact window.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [quickContactBadgeStyleSmallWindowLarge](#)  
(/reference/android/R.attr.html#quickContactBadgeStyleSmallWindowLarge).

Constant Value: 124 (0x0000007c)

**public static final int Theme\_quickContactBadgeStyleSmallWindowMedium**

Default quickcontact badge style with medium quickcontact window.

Must be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:][type:]name`".

This corresponds to the global attribute resource symbol [quickContactBadgeStyleSmallWindowMedium](#)  
[\(/reference/android/R.attr.html#quickContactBadgeStyleSmallWindowMedium\)](#).

Constant Value: 123 (0x0000007b)

**public static final int Theme\_quickContactBadgeStyleSmallWindowSmall**

Default quickcontact badge style with small quickcontact window.

Must be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:][type:]name`".

This corresponds to the global attribute resource symbol [quickContactBadgeStyleSmallWindowSmall](#)  
[\(/reference/android/R.attr.html#quickContactBadgeStyleSmallWindowSmall\)](#).

Constant Value: 122 (0x0000007a)

**public static final int Theme\_quickContactBadgeStyleWindowLarge**

Default quickcontact badge style with large quickcontact window.

Must be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:][type:]name`".

This corresponds to the global attribute resource symbol [quickContactBadgeStyleWindowLarge](#)  
[\(/reference/android/R.attr.html#quickContactBadgeStyleWindowLarge\)](#).

Constant Value: 121 (0x00000079)

**public static final int Theme\_quickContactBadgeStyleWindowMedium**

Default quickcontact badge style with medium quickcontact window.

Must be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:][type:]name`".

This corresponds to the global attribute resource symbol [quickContactBadgeStyleWindowMedium](#)  
[\(/reference/android/R.attr.html#quickContactBadgeStyleWindowMedium\)](#).

Constant Value: 120 (0x00000078)

**public static final int Theme\_quickContactBadgeStyleWindowSmall**

Default quickcontact badge style with small quickcontact window.

Must be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:][type:]name`".

This corresponds to the global attribute resource symbol [quickContactBadgeStyleWindowSmall](#)  
[\(/reference/android/R.attr.html#quickContactBadgeStyleWindowSmall\)](#).

Constant Value: 119 (0x00000077)

**public static final int Theme\_radioButtonStyle**

Default RadioButton style.

Must be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:][type:]name`".

This corresponds to the global attribute resource symbol [radioButtonStyle](#)  
[\(/reference/android/R.attr.html#radioButtonStyle\)](#).

Constant Value: 71 (0x00000047)

**public static final int Theme\_ratingBarStyle**

Default RatingBar style.

Must be a reference to another resource, in the form "`@[+][package:]type:name`" or to a theme attribute in the form "`?[package:][type:]name`".

in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [ratingBarStyle](#) ([/reference/android/R.attr.html#ratingBarStyle](#)).

Constant Value: 69 (0x00000045)

#### public static final int **Theme\_ratingBarStyleIndicator**

Indicator RatingBar style.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [ratingBarStyleIndicator](#) ([/reference/android/R.attr.html#ratingBarStyleIndicator](#)).

Constant Value: 99 (0x00000063)

#### public static final int **Theme\_ratingBarStyleSmall**

Small indicator RatingBar style.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [ratingBarStyleSmall](#) ([/reference/android/R.attr.html#ratingBarStyleSmall](#)).

Constant Value: 70 (0x00000046)

#### public static final int **Theme\_ringtonePreferenceStyle**

Default style for RingtonePreference.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [ringtonePreferenceStyle](#) ([/reference/android/R.attr.html#ringtonePreferenceStyle](#)).

Constant Value: 91 (0x0000005b)

#### public static final int **Theme\_scrollViewStyle**

Default ScrollView style.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [scrollViewStyle](#) ([/reference/android/R.attr.html#scrollViewStyle](#)).

Constant Value: 72 (0x00000048)

#### public static final int **Theme\_seekBarStyle**

Default SeekBar style.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [seekBarStyle](#) ([/reference/android/R.attr.html#seekBarStyle](#)).

Constant Value: 68 (0x00000044)

#### public static final int **Theme\_segmentedButtonStyle**

Style for segmented buttons - a container that houses several buttons with the appearance of a single button broken into segments.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [segmentedButtonStyle](#) ([/reference/android/R.attr.html#segmentedButtonStyle](#)).

Constant Value: 169 (0x000000a9)

#### **public static final int Theme\_selectableItemBackground**

Background drawable for standalone items that need focus/pressed states.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [selectableItemBackground](#)  
[\(/reference/android/R.attr.html#selectableItemBackground\)](#).

Constant Value: 158 (0x0000009e)

#### **public static final int Theme\_spinnerDropDownItemStyle**

Default style for spinner drop down items.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [spinnerDropDownItemStyle](#)  
[\(/reference/android/R.attr.html#spinnerDropDownItemStyle\)](#).

Constant Value: 79 (0x0000004f)

#### **public static final int Theme\_spinnerItemStyle**

Default spinner item style.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [spinnerItemStyle](#)  
[\(/reference/android/R.attr.html#spinnerItemStyle\)](#).

Constant Value: 81 (0x00000051)

#### **public static final int Theme\_spinnerStyle**

Default Spinner style.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [spinnerStyle](#)  
[\(/reference/android/R.attr.html#spinnerStyle\)](#).

Constant Value: 73 (0x00000049)

#### **public static final int Theme\_starStyle**

Default Star style.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [starStyle](#)  
[\(/reference/android/R.attr.html#starStyle\)](#).

Constant Value: 74 (0x0000004a)

#### **public static final int Theme\_switchPreferenceStyle**

Default style for switch preferences.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [switchPreferenceStyle](#)  
[\(/reference/android/R.attr.html#switchPreferenceStyle\)](#).

Constant Value: 190 (0x000000be)

#### **public static final int Theme\_tabWidgetStyle**

Default TabWidget style.

Must be a reference to another resource, in the form "@[+][*package*: ]*type*:*name*" or to a theme attribute in the form "?[*package*: ][*type*: ]*name*".

This corresponds to the global attribute resource symbol [tabWidgetStyle](#)  
(/reference/android/R.attr.html#tabWidgetStyle).

Constant Value: 75 (0x0000004b)

#### public static final int **Theme\_textAppearance**

Default appearance of text: color, typeface, size, and style.

Must be a reference to another resource, in the form "@[+][*package*: ]*type*:*name*" or to a theme attribute in the form "?[*package*: ][*type*: ]*name*".

This corresponds to the global attribute resource symbol [textAppearance](#)  
(/reference/android/R.attr.html#textAppearance).

Constant Value: 4 (0x00000004)

#### public static final int **Theme\_textAppearanceButton**

Text color, typeface, size, and style for the text inside of a button.

Must be a reference to another resource, in the form "@[+][*package*: ]*type*:*name*" or to a theme attribute in the form "?[*package*: ][*type*: ]*name*".

This corresponds to the global attribute resource symbol [textAppearanceButton](#)  
(/reference/android/R.attr.html#textAppearanceButton).

Constant Value: 95 (0x00000005f)

#### public static final int **Theme\_textAppearanceInverse**

Default appearance of text against an inverted background: color, typeface, size, and style.

Must be a reference to another resource, in the form "@[+][*package*: ]*type*:*name*" or to a theme attribute in the form "?[*package*: ][*type*: ]*name*".

This corresponds to the global attribute resource symbol [textAppearanceInverse](#)  
(/reference/android/R.attr.html#textAppearanceInverse).

Constant Value: 5 (0x00000005)

#### public static final int **Theme\_textAppearanceLarge**

Text color, typeface, size, and style for "large" text. Defaults to primary text color.

Must be a reference to another resource, in the form "@[+][*package*: ]*type*:*name*" or to a theme attribute in the form "?[*package*: ][*type*: ]*name*".

This corresponds to the global attribute resource symbol [textAppearanceLarge](#)  
(/reference/android/R.attr.html#textAppearanceLarge).

Constant Value: 16 (0x00000010)

#### public static final int **Theme\_textAppearanceLargeInverse**

Text color, typeface, size, and style for "large" inverse text. Defaults to primary inverse text color.

Must be a reference to another resource, in the form "@[+][*package*: ]*type*:*name*" or to a theme attribute in the form "?[*package*: ][*type*: ]*name*".

This corresponds to the global attribute resource symbol [textAppearanceLargeInverse](#)  
(/reference/android/R.attr.html#textAppearanceLargeInverse).

Constant Value: 19 (0x00000013)

#### public static final int **Theme\_textAppearanceLargePopupMenu**

Text color, typeface, size, and style for the text inside of a popup menu.

Must be a reference to another resource, in the form "@[+][*package*: ]*type*:*name*" or to a theme attribute in the form "?[*package*: ][*type*: ]*name*".

This corresponds to the global attribute resource symbol [textAppearanceLargePopupMenu](#)  
(/reference/android/R.attr.html#textAppearanceLargePopupMenu).

Constant Value: 150 (0x00000096)

#### **public static final int Theme\_textAppearanceListItem**

The preferred TextAppearance for the primary text of list items.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [textAppearanceListItem](#)  
(/reference/android/R.attr.html#textAppearanceListItem).

Constant Value: 207 (0x000000cf)

#### **public static final int Theme\_textAppearanceListItemSmall**

The preferred TextAppearance for the primary text of small list items.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [textAppearanceListItemSmall](#)  
(/reference/android/R.attr.html#textAppearanceListItemSmall).

Constant Value: 208 (0x000000d0)

#### **public static final int Theme\_textAppearanceMedium**

Text color, typeface, size, and style for "medium" text. Defaults to primary text color.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [textAppearanceMedium](#)  
(/reference/android/R.attr.html#textAppearanceMedium).

Constant Value: 17 (0x00000011)

#### **public static final int Theme\_textAppearanceMediumInverse**

Text color, typeface, size, and style for "medium" inverse text. Defaults to primary inverse text color.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [textAppearanceMediumInverse](#)  
(/reference/android/R.attr.html#textAppearanceMediumInverse).

Constant Value: 20 (0x00000014)

#### **public static final int Theme\_textAppearanceSearchResultSubtitle**

Text color, typeface, size, and style for system search result subtitle. Defaults to primary inverse text color.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [textAppearanceSearchResultSubtitle](#)  
(/reference/android/R.attr.html#textAppearanceSearchResultSubtitle).

Constant Value: 116 (0x00000074)

#### **public static final int Theme\_textAppearanceSearchResultTitle**

Text color, typeface, size, and style for system search result title. Defaults to primary inverse text color.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [textAppearanceSearchResultTitle](#)  
(/reference/android/R.attr.html#textAppearanceSearchResultTitle).

Constant Value: 117 (0x00000075)

#### **public static final int Theme\_textAppearanceSmall**

Text color, typeface, size, and style for "small" text. Defaults to secondary text color.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [textAppearanceSmall](#)  
(/reference/android/R.attr.html#textAppearanceSmall).

Constant Value: 18 (0x00000012)

#### public static final int **Theme\_textAppearanceSmallInverse**

Text color, typeface, size, and style for "small" inverse text. Defaults to secondary inverse text color.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [textAppearanceSmallInverse](#)  
(/reference/android/R.attr.html#textAppearanceSmallInverse).

Constant Value: 21 (0x00000015)

#### public static final int **Theme\_textAppearanceSmallPopupMenu**

Text color, typeface, size, and style for small text inside of a popup menu.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [textAppearanceSmallPopupMenu](#)  
(/reference/android/R.attr.html#textAppearanceSmallPopupMenu).

Constant Value: 151 (0x00000097)

#### public static final int **Theme\_textCheckMark**

Drawable to use for check marks.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [textCheckMark](#)  
(/reference/android/R.attr.html#textCheckMark).

Constant Value: 22 (0x00000016)

#### public static final int **Theme\_textCheckMarkInverse**

This symbol is the offset where the [textCheckMarkInverse](#)  
(/reference/android/R.attr.html#textCheckMarkInverse) attribute's value can be found in the [Theme](#)  
(/reference/android/R.styleable.html#Theme) array.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

Constant Value: 23 (0x00000017)

#### public static final int **Theme\_textColorAlertDialogListItem**

Color of list item text in alert dialogs.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorAlertDialogListItem](#)  
(/reference/android/R.attr.html#textColorAlertDialogListItem).

Constant Value: 153 (0x00000099)

#### public static final int **Theme\_textColorHighlightInverse**

Color of highlighted text, when used in a light theme.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorHighlightInverse](#)  
(/reference/android/R.attr.html#textColorHighlightInverse).

Constant Value: 176 (0x000000b0)

#### public static final int **Theme\_textColorHintInverse**

Inverse hint text color.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorHintInverse](#)  
(/reference/android/R.attr.html#textColorHintInverse).

Constant Value: 15 (0x0000000f)

#### public static final int **Theme\_textColorLinkInverse**

Color of link text (URLs), when used in a light theme.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorLinkInverse](#)  
(/reference/android/R.attr.html#textColorLinkInverse).

Constant Value: 177 (0x000000b1)

#### public static final int **Theme\_textColorPrimary**

The most prominent text color.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorPrimary](#)  
(/reference/android/R.attr.html#textColorPrimary).

Constant Value: 6 (0x00000006)

#### public static final int **Theme\_textColorPrimaryDisableOnly**

Bright text color. Only differentiates based on the disabled state.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorPrimaryDisableOnly](#)  
(/reference/android/R.attr.html#textColorPrimaryDisableOnly).

Constant Value: 7 (0x00000007)

#### public static final int **Theme\_textColorPrimaryInverse**

Primary inverse text color, useful for inverted backgrounds.

May be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorPrimaryInverse](#)  
(/reference/android/R.attr.html#textColorPrimaryInverse).

Constant Value: 9 (0x00000009)

#### public static final int **Theme\_textColorPrimaryInverseDisableOnly**

Bright inverse text color. Only differentiates based on the disabled state.

May be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ][ type: ] name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorPrimaryInverseDisableOnly](#)  
[\(/reference/android/R.attr.html#textColorPrimaryInverseDisableOnly\)](#).

Constant Value: 114 (0x00000072)

#### public static final int **Theme\_textColorPrimaryInverseNoDisable**

Bright inverse text color. This does not differentiate the disabled state.

May be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ][ type: ] name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorPrimaryInverseNoDisable](#)  
[\(/reference/android/R.attr.html#textColorPrimaryInverseNoDisable\)](#).

Constant Value: 13 (0x0000000d)

#### public static final int **Theme\_textColorPrimaryNoDisable**

Bright text color. This does not differentiate the disabled state. As an example, buttons use this since they display the disabled state via the background and not the foreground text color.

May be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ][ type: ] name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorPrimaryNoDisable](#)  
[\(/reference/android/R.attr.html#textColorPrimaryNoDisable\)](#).

Constant Value: 11 (0x0000000b)

#### public static final int **Theme\_textColorSecondary**

Secondary text color.

May be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ][ type: ] name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorSecondary](#)  
[\(/reference/android/R.attr.html#textColorSecondary\)](#).

Constant Value: 8 (0x00000008)

#### public static final int **Theme\_textColorSecondaryInverse**

Secondary inverse text color, useful for inverted backgrounds.

May be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ][ type: ] name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorSecondaryInverse](#)  
[\(/reference/android/R.attr.html#textColorSecondaryInverse\)](#).

Constant Value: 10 (0x0000000a)

#### public static final int **Theme\_textColorSecondaryInverseNoDisable**

Dim inverse text color. This does not differentiate the disabled state.

May be a reference to another resource, in the form "`@[+][ package: ] type: name`" or to a theme attribute in the form "`?[ package: ][ type: ] name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorSecondaryInverseNoDisable](#)  
[\(/reference/android/R.attr.html#textColorSecondaryInverseNoDisable\)](#).

Constant Value: 14 (0x0000000e)

#### public static final int **Theme\_textColorSecondaryNoDisable**

Dim text color. This does not differentiate the disabled state.

May be a reference to another resource, in the form "`@[+][package: ]type: name`" or to a theme attribute in the form "`?[package: ][type: ]name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorSecondaryNoDisable](#)  
[\(/reference/android/R.attr.html#textColorSecondaryNoDisable\)](#).

Constant Value: 12 (0x0000000c)

#### public static final int **Theme\_textColorTertiary**

Tertiary text color.

May be a reference to another resource, in the form "`@[+][package: ]type: name`" or to a theme attribute in the form "`?[package: ][type: ]name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorTertiary](#)  
[\(/reference/android/R.attr.html#textColorTertiary\)](#).

Constant Value: 100 (0x00000064)

#### public static final int **Theme\_textColorTertiaryInverse**

Tertiary inverse text color, useful for inverted backgrounds.

May be a reference to another resource, in the form "`@[+][package: ]type: name`" or to a theme attribute in the form "`?[package: ][type: ]name`".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColorTertiaryInverse](#)  
[\(/reference/android/R.attr.html#textColorTertiaryInverse\)](#).

Constant Value: 101 (0x00000065)

#### public static final int **Theme\_textEditNoPasteWindowLayout**

Variation of `textEditPasteWindowLayout` displayed when the clipboard is empty.

Must be a reference to another resource, in the form "`@[+][package: ]type: name`" or to a theme attribute in the form "`?[package: ][type: ]name`".

This corresponds to the global attribute resource symbol [textEditNoPasteWindowLayout](#)  
[\(/reference/android/R.attr.html#textEditNoPasteWindowLayout\)](#).

Constant Value: 163 (0x000000a3)

#### public static final int **Theme\_textEditPasteWindowLayout**

The layout of the view that is displayed on top of the cursor to paste inside a `TextEdit` field.

Must be a reference to another resource, in the form "`@[+][package: ]type: name`" or to a theme attribute in the form "`?[package: ][type: ]name`".

This corresponds to the global attribute resource symbol [textEditPasteWindowLayout](#)  
[\(/reference/android/R.attr.html#textEditPasteWindowLayout\)](#).

Constant Value: 162 (0x000000a2)

#### public static final int **Theme\_textEditSideNoPasteWindowLayout**

Variation of `textEditSidePasteWindowLayout` displayed when the clipboard is empty.

Must be a reference to another resource, in the form "`@[+][package: ]type: name`" or to a theme attribute in the form "`?[package: ][type: ]name`".

This corresponds to the global attribute resource symbol [textEditSideNoPasteWindowLayout](#)  
[\(/reference/android/R.attr.html#textEditSideNoPasteWindowLayout\)](#).

Constant Value: 187 (0x000000bb)

#### **public static final int Theme\_textEditSidePasteWindowLayout**

Used instead of `textEditPasteWindowLayout` when the window is moved on the side of the insertion cursor because it would be clipped if it were positioned on top.

Must be a reference to another resource, in the form "`@[+][package: ]type: name`" or to a theme attribute in the form "`?[package: ][type: ]name`".

This corresponds to the global attribute resource symbol [textEditSidePasteWindowLayout](#)  
[\(/reference/android/R.attr.html#textEditSidePasteWindowLayout\)](#).

Constant Value: 186 (0x000000ba)

#### **public static final int Theme\_textEditSuggestionItemLayout**

Layout of the `TextView` item that will populate the suggestion popup window.

Must be a reference to another resource, in the form "`@[+][package: ]type: name`" or to a theme attribute in the form "`?[package: ][type: ]name`".

This corresponds to the global attribute resource symbol [textEditSuggestionItemLayout](#)  
[\(/reference/android/R.attr.html#textEditSuggestionItemLayout\)](#).

Constant Value: 192 (0x000000c0)

#### **public static final int Theme\_textSelectHandle**

Reference to a drawable that will be used to display a text selection anchor for positioning the cursor within text.

Must be a reference to another resource, in the form "`@[+][package: ]type: name`" or to a theme attribute in the form "`?[package: ][type: ]name`".

This corresponds to the global attribute resource symbol [textSelectHandle](#)  
[\(/reference/android/R.attr.html#textSelectHandle\)](#).

Constant Value: 129 (0x00000081)

#### **public static final int Theme\_textSelectHandleLeft**

Reference to a drawable that will be used to display a text selection anchor on the left side of a selection region.

Must be a reference to another resource, in the form "`@[+][package: ]type: name`" or to a theme attribute in the form "`?[package: ][type: ]name`".

This corresponds to the global attribute resource symbol [textSelectHandleLeft](#)  
[\(/reference/android/R.attr.html#textSelectHandleLeft\)](#).

Constant Value: 127 (0x0000007f)

#### **public static final int Theme\_textSelectHandleRight**

Reference to a drawable that will be used to display a text selection anchor on the right side of a selection region.

Must be a reference to another resource, in the form "`@[+][package: ]type: name`" or to a theme attribute in the form "`?[package: ][type: ]name`".

This corresponds to the global attribute resource symbol [textSelectHandleRight](#)  
[\(/reference/android/R.attr.html#textSelectHandleRight\)](#).

Constant Value: 128 (0x00000080)

#### **public static final int Theme\_textSelectHandleWindowStyle**

Reference to a style that will be used for the window containing a text selection anchor.

Must be a reference to another resource, in the form "`@[+][package: ]type: name`" or to a theme attribute in the form "`?[package: ][type: ]name`".

This corresponds to the global attribute resource symbol [textSelectHandleWindowStyle](#)  
[\(/reference/android/R.attr.html#textSelectHandleWindowStyle\)](#).

Constant Value: 130 (0x00000082)

**public static final int Theme\_textSuggestionsWindowStyle**

Reference to a style that will be used for the window containing a list of possible text suggestions in an EditText.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [textSuggestionsWindowStyle](#)  
[\(/reference/android/R.attr.html#textSuggestionsWindowStyle\)](#).

Constant Value: 191 (0x000000bf)

**public static final int Theme\_textViewStyle**

Default TextView style.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [textViewStyle](#)  
[\(/reference/android/R.attr.html#textViewStyle\)](#).

Constant Value: 76 (0x0000004c)

**public static final int Theme\_webTextViewStyle**

Default WebTextView style.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [webTextViewStyle](#)  
[\(/reference/android/R.attr.html#webTextViewStyle\)](#).

Constant Value: 126 (0x0000007e)

**public static final int Theme\_webViewStyle**

Default WebView style.

Must be a reference to another resource, in the form "@[+][*package:*]*type:name*" or to a theme attribute in the form "?[*package:*][*type:*]*name*".

This corresponds to the global attribute resource symbol [webViewStyle](#)  
[\(/reference/android/R.attr.html#webViewStyle\)](#).

Constant Value: 77 (0x0000004d)

**public static final int Theme\_windowActionBar**

Flag indicating whether this window should have an Action Bar in place of the usual title bar.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*][*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [windowActionBar](#)  
[\(/reference/android/R.attr.html#windowActionBar\)](#).

Constant Value: 131 (0x00000083)

**public static final int Theme\_windowActionBarOverlay**

Flag indicating whether this window's Action Bar should overlay application content. Does nothing if the window would not have an Action Bar.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*]*type:name*") or theme attribute (in the form "?[*package:*][*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [windowActionBarOverlay](#)  
[\(/reference/android/R.attr.html#windowActionBarOverlay\)](#).

Constant Value: 139 (0x0000008b)

## **public static final int Theme\_windowActionBarOverlay**

Flag indicating whether action modes should overlay window content when there is not reserved space for their UI (such as an Action Bar).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:] type: name") or theme attribute (in the form "?[package:] [type:] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowActionBarOverlay](#) ([/reference/android/R.attr.html#windowActionBarOverlay](#)).

Constant Value: 138 (0x0000008a)

## **public static final int Theme\_windowAnimationStyle**

Reference to a style resource holding the set of window animations to use, which can be any of the attributes defined by [windowAnimation](#) ([/reference/android/R.styleable.html#WindowAnimation](#)).

Must be a reference to another resource, in the form "@[+] [package:] type: name" or to a theme attribute in the form "?[package:] [type:] name".

This corresponds to the global attribute resource symbol [windowAnimationStyle](#) ([/reference/android/R.attr.html#windowAnimationStyle](#)).

Constant Value: 93 (0x0000005d)

## **public static final int Theme\_windowBackground**

Drawable to use as the overall window background. As of [HONEYCOMB](#) ([/reference/android/os/Build.VERSION\\_CODES.html#HONEYCOMB](#)), this may be a selector that uses state\_accelerated to pick a non-solid color when running on devices that can draw such a bitmap with complex compositing on top at 60fps.

There are a few special considerations to use when setting this drawable:

- This information will be used to infer the pixel format for your window's surface. If the drawable has any non-opaque pixels, your window will be translucent (32 bpp).
- If you want to draw the entire background yourself, you should set this drawable to some solid color that closely matches that background (so the system's preview of your window will match), and then in code manually set your window's background to null so it will not be drawn.

Must be a reference to another resource, in the form "@[+] [package:] type: name" or to a theme attribute in the form "?[package:] [type:] name".

This corresponds to the global attribute resource symbol [windowBackground](#) ([/reference/android/R.attr.html#windowBackground](#)).

Constant Value: 36 (0x00000024)

## **public static final int Theme\_windowCloseOnTouchOutside**

Control whether a container should automatically close itself if the user touches outside of it. This only applies to activities and dialogs.

Note: this attribute will only be respected for applications that are targeting [HONEYCOMB](#) ([/reference/android/os/Build.VERSION\\_CODES.html#HONEYCOMB](#)) or later.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:] type: name") or theme attribute (in the form "?[package:] [type:] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowCloseOnTouchOutside](#) ([/reference/android/R.attr.html#windowCloseOnTouchOutside](#)).

Constant Value: 183 (0x000000b7)

## **public static final int Theme\_windowContentOverlay**

This Drawable is overlaid over the foreground of the Window's content area, usually to place a shadow below the title.

Must be a reference to another resource, in the form "@[+] [package:] type: name" or to a theme attribute in the form "?[package:] [type:] name".

This corresponds to the global attribute resource symbol [windowContentOverlay](#) ([/reference/android/R.attr.html#windowContentOverlay](#)).

Constant Value: 41 (0x00000029)

#### **public static final int Theme\_windowDisablePreview**

Flag allowing you to disable the preview animation for a window. The default value is false; if set to true, the system can never use the window's theme to show a preview of it before your actual instance is shown to the user.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowDisablePreview](#)  
[\(/reference/android/R.attr.html#windowDisablePreview\)](#).

Constant Value: 107 (0x0000006b)

#### **public static final int Theme\_windowEnableSplitTouch**

Flag indicating that this window should allow touches to be split across other windows that also support split touch. The default value is true for applications with a targetSdkVersion of Honeycomb or newer; false otherwise. When this flag is false, the first pointer that goes down determines the window to which all subsequent touches go until all pointers go up. When this flag is true, each pointer (not necessarily the first) that goes down determines the window to which all subsequent touches of that pointer will go until that pointers go up thereby enabling touches with multiple pointers to be split across multiple windows.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowEnableSplitTouch](#)  
[\(/reference/android/R.attr.html#windowEnableSplitTouch\)](#).

Constant Value: 164 (0x000000a4)

#### **public static final int Theme\_windowFrame**

Drawable to use as a frame around the window.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [windowFrame](#)  
[\(/reference/android/R.attr.html#windowFrame\)](#).

Constant Value: 37 (0x00000025)

#### **public static final int Theme\_windowFullscreen**

Flag indicating whether this window should fill the entire screen.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowFullscreen](#)  
[\(/reference/android/R.attr.html#windowFullscreen\)](#).

Constant Value: 97 (0x00000061)

#### **public static final int Theme\_windowIsFloating**

Flag indicating whether this is a floating window.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowIsFloating](#)  
[\(/reference/android/R.attr.html#windowIsFloating\)](#).

Constant Value: 39 (0x00000027)

#### **public static final int Theme\_windowIsTranslucent**

Flag indicating whether this is a translucent window.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowIsTranslucent](#)  
[\(/reference/android/R.attr.html#windowIsTranslucent\)](#).

Constant Value: 40 (0x00000028)

#### **public static final int Theme\_windowNoDisplay**

Flag indicating that this window should not be displayed at all. The default value is false; if set to true, and this window is the main window of an Activity, then it will never actually be added to the window manager. This means that your activity must immediately quit without waiting for user interaction, because there will be no such interaction coming.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowNoDisplay](#)  
[\(/reference/android/R.attr.html#windowNoDisplay\)](#).

Constant Value: 105 (0x00000069)

#### **public static final int Theme\_windowNoTitle**

Flag indicating whether there should be no title on this window.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowNoTitle](#)  
[\(/reference/android/R.attr.html#windowNoTitle\)](#).

Constant Value: 38 (0x00000026)

#### **public static final int Theme\_windowShowWallpaper**

Flag indicating that this window's background should be the user's current wallpaper.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowShowWallpaper](#)  
[\(/reference/android/R.attr.html#windowShowWallpaper\)](#).

Constant Value: 115 (0x00000073)

#### **public static final int Theme\_windowSoftInputMode**

Defines the default soft input state that this window would like when it is displayed.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
stateUnspecified	0	Not specified, use what the system thinks is best. This is the default.
stateUnchanged	1	Leave the soft input window as-is, in whatever state it last was.
stateHidden	2	Make the soft input area hidden when normally appropriate (when the user is navigating forward to your window).
stateAlwaysHidden	3	Always make the soft input area hidden when this window has input focus.
stateVisible	4	Make the soft input area visible when normally appropriate (when the user is navigating forward to your window).
stateAlwaysVisible	5	Always make the soft input area visible when this window has input focus.  The window resize/pan adjustment has not been specified, the system will automatically select between resize and pan modes, depending on whether the content of the window has any layout views that can scroll
	0x00	

adjustUnspecified		their contents. If there is such a view, then the window will be resized, with the assumption being that the resizable area can be reduced to make room for the input UI.
adjustResize	0x10	Always resize the window: the content area of the window is reduced to make room for the soft input area.
adjustPan	0x20	Don't resize the window to make room for the soft input area; instead pan the contents of the window as focus moves inside of it so that the user can see what they are typing. This is generally less desireable than panning because the user may need to close the input area to get at and interact with parts of the window.
adjustNothing	0x30	Don't resize or pan the window to make room for the soft input area; the window is never adjusted for it.

This corresponds to the global attribute resource symbol [windowSoftInputMode](#)  
[\(/reference/android/R.attr.html#windowSoftInputMode\)](#).

Constant Value: 108 (0x0000006c)

#### **public static final int Theme\_windowTitleBackgroundColor**

The style resource to use for a window's title area.

Must be a reference to another resource, in the form "`@[+][ package: ]type: name`" or to a theme attribute in the form "`?[ package: ][ type: ]name`".

This corresponds to the global attribute resource symbol [windowTitleBackgroundColor](#)  
[\(/reference/android/R.attr.html#windowTitleBackgroundColor\)](#).

Constant Value: 44 (0x0000002c)

#### **public static final int Theme\_windowTitleSize**

The style resource to use for a window's title bar height.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "`@[ package: ]type: name`") or theme attribute (in the form "`?[ package: ][ type: ]name`") containing a value of this type.

This corresponds to the global attribute resource symbol [windowTitleSize](#)  
[\(/reference/android/R.attr.html#windowTitleSize\)](#).

Constant Value: 42 (0x0000002a)

#### **public static final int Theme\_windowTitleStyle**

The style resource to use for a window's title text.

Must be a reference to another resource, in the form "`@[+][ package: ]type: name`" or to a theme attribute in the form "`?[ package: ][ type: ]name`".

This corresponds to the global attribute resource symbol [windowTitleStyle](#)  
[\(/reference/android/R.attr.html#windowTitleStyle\)](#).

Constant Value: 43 (0x0000002b)

#### **public static final int Theme\_yesNoPreferenceStyle**

Default style for YesNoPreference.

Must be a reference to another resource, in the form "`@[+][ package: ]type: name`" or to a theme attribute in the form "`?[ package: ][ type: ]name`".

This corresponds to the global attribute resource symbol [yesNoPreferenceStyle](#)  
[\(/reference/android/R.attr.html#yesNoPreferenceStyle\)](#).

Constant Value: 88 (0x00000058)

#### **public static final int ToggleButton\_disabledAlpha**

The alpha to apply to the indicator when disabled.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "`@[ package: ]type: name`") or theme attribute (in

the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [disabledAlpha](#) ([/reference/android/R.attr.html#disabledAlpha](#)).

Constant Value: 0 (0x00000000)

#### public static final int ToggleButton\_textOff

The text for the button when it is not checked.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [textOff](#) ([/reference/android/R.attr.html#textOff](#)).

Constant Value: 2 (0x00000002)

#### public static final int ToggleButton\_textOn

The text for the button when it is checked.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [textOn](#) ([/reference/android/R.attr.html#textOn](#)).

Constant Value: 1 (0x00000001)

#### public static final int TranslateAnimation\_fromXDelta

This symbol is the offset where the [fromXDelta](#) ([/reference/android/R.attr.html#fromXDelta](#)) attribute's value can be found in the [TranslateAnimation](#) ([/reference/android/R.styleable.html#TranslateAnimation](#)) array.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 0 (0x00000000)

#### public static final int TranslateAnimation\_fromYDelta

This symbol is the offset where the [fromYDelta](#) ([/reference/android/R.attr.html#fromYDelta](#)) attribute's value can be found in the [TranslateAnimation](#) ([/reference/android/R.styleable.html#TranslateAnimation](#)) array.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 2 (0x00000002)

#### public static final int TranslateAnimation\_toXDelta

This symbol is the offset where the [toXDelta](#) ([/reference/android/R.attr.html#toXDelta](#)) attribute's value can be found in the [TranslateAnimation](#) ([/reference/android/R.styleable.html#TranslateAnimation](#)) array.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%".

The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 1 (0x00000001)

#### public static final int TranslateAnimation\_toYDelta

This symbol is the offset where the [toYDelta](#) ([/reference/android/R.attr.html#toYDelta](#)) attribute's value can be found in the [TranslateAnimation](#) ([/reference/android/R.styleable.html#TranslateAnimation](#)) array.

May be a floating point value, such as "1.2".

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 3 (0x00000003)

#### public static final int TwoLineListItem\_mode

This symbol is the offset where the [mode](#) ([/reference/android/R.attr.html#mode](#)) attribute's value can be found in the [TwoLineListItem](#) ([/reference/android/R.styleable.html#TwoLineListItem](#)) array.

Must be one of the following constant values.

Constant	Value	Description
oneLine	1	Always show only the first line.
collapsing	2	When selected show both lines, otherwise show only the first line. This is the default mode.
twoLine	3	Always show both lines.

Constant Value: 0 (0x00000000)

#### public static final int VerticalSlider\_Layout\_layout\_scale

This symbol is the offset where the [layout\\_scale](#) ([/reference/android/R.attr.html#layout\\_scale](#)) attribute's value can be found in the [VerticalSlider\\_Layout](#) ([/reference/android/R.styleable.html#VerticalSlider\\_Layout](#)) array.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 0 (0x00000000)

#### public static final int ViewAnimator\_animateFirstView

Defines whether to animate the current View when the ViewAnimation is first displayed.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [animateFirstView](#) ([/reference/android/R.attr.html#animateFirstView](#)).

Constant Value: 2 (0x00000002)

#### public static final int ViewAnimator\_inAnimation

Identifier for the animation to use when a view is shown.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [inAnimation](#) ([/reference/android/R.attr.html#inAnimation](#)).

Constant Value: 0 (0x00000000)

#### **public static final int ViewAnimator\_outAnimation**

Identifier for the animation to use when a view is hidden.

Must be a reference to another resource, in the form "@[+][*package:*]*type:**name*" or to a theme attribute in the form "?[*package:*] [*type:*]*name*".

This corresponds to the global attribute resource symbol [outAnimation](#)  
[\(/reference/android/R.attr.html#outAnimation\)](#).

Constant Value: 1 (0x00000001)

#### **public static final int ViewDrawableStates\_state\_accelerated**

State value for [StateListDrawable](#)

[\(/reference/android/graphics/drawable/StateListDrawable.html\)](#), indicating that the Drawable is in a view that is hardware accelerated. This means that the device can at least render a full-screen scaled bitmap with one layer of text and bitmaps composited on top of it at 60fps. When this is set, the colorBackgroundCacheHint will be ignored even if it specifies a solid color, since that optimization is not needed.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*]*type:**name*") or theme attribute (in the form "?[*package:*] [*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_accelerated](#)  
[\(/reference/android/R.attr.html#state\\_accelerated\)](#).

Constant Value: 6 (0x00000006)

#### **public static final int ViewDrawableStates\_state\_activated**

State value for [StateListDrawable](#)

[\(/reference/android/graphics/drawable/StateListDrawable.html\)](#), set when a view or its parent has been "activated" meaning the user has currently marked it as being of interest. This is an alternative representation of state\_checked for when the state should be propagated down the view hierarchy.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*]*type:**name*") or theme attribute (in the form "?[*package:*] [*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_activated](#)  
[\(/reference/android/R.attr.html#state\\_activated\)](#).

Constant Value: 5 (0x00000005)

#### **public static final int ViewDrawableStates\_state\_drag\_can\_accept**

State for [StateListDrawable](#) [\(/reference/android/graphics/drawable/StateListDrawable.html\)](#) indicating that the Drawable is in a view that is capable of accepting a drop of the content currently being manipulated in a drag-and-drop operation.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*]*type:**name*") or theme attribute (in the form "?[*package:*] [*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_drag\\_can\\_accept](#)  
[\(/reference/android/R.attr.html#state\\_drag\\_can\\_accept\)](#).

Constant Value: 8 (0x00000008)

#### **public static final int ViewDrawableStates\_state\_drag\_hovered**

State for [StateListDrawable](#) [\(/reference/android/graphics/drawable/StateListDrawable.html\)](#) indicating that a drag operation (for which the Drawable's view is a valid recipient) is currently positioned over the Drawable.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[*package:*]*type:**name*") or theme attribute (in the form "?[*package:*] [*type:*]*name*") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_drag\\_hovered](#)

[\(/reference/android/R.attr.html#state\\_drag\\_hovered\)](#).

Constant Value: 9 (0x00000009)

#### public static final int **ViewDrawableStates\_state\_enabled**

State value for [StateListDrawable](#)

[\(/reference/android/graphics/drawable/StateListDrawable.html\)](#), set when a view is enabled.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_enabled](#)

[\(/reference/android/R.attr.html#state\\_enabled\)](#).

Constant Value: 2 (0x00000002)

#### public static final int **ViewDrawableStates\_state\_focused**

State value for [StateListDrawable](#)

[\(/reference/android/graphics/drawable/StateListDrawable.html\)](#), set when a view has input focus.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_focused](#)

[\(/reference/android/R.attr.html#state\\_focused\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **ViewDrawableStates\_state\_hovered**

State value for [StateListDrawable](#)

[\(/reference/android/graphics/drawable/StateListDrawable.html\)](#), set when a pointer is hovering over the view.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_hovered](#)

[\(/reference/android/R.attr.html#state\\_hovered\)](#).

Constant Value: 7 (0x00000007)

#### public static final int **ViewDrawableStates\_state\_pressed**

State value for [StateListDrawable](#)

[\(/reference/android/graphics/drawable/StateListDrawable.html\)](#), set when the user is pressing down in a view.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_pressed](#)

[\(/reference/android/R.attr.html#state\\_pressed\)](#).

Constant Value: 4 (0x00000004)

#### public static final int **ViewDrawableStates\_state\_selected**

State value for [StateListDrawable](#)

[\(/reference/android/graphics/drawable/StateListDrawable.html\)](#), set when a view (or one of its parents) is currently selected.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:]type:name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_selected](#)

[\(/reference/android/R.attr.html#state\\_selected\)](#).  
Constant Value: 3 (0x00000003)

**public static final int ViewDrawableStates\_state\_window\_focused**

State value for [StateListDrawable](#)  
[\(/reference/android/graphics/drawable/StateListDrawable.html\)](#), set when a view's window has input focus.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [state\\_window\\_focused](#)  
[\(/reference/android/R.attr.html#state\\_window\\_focused\)](#).

Constant Value: 1 (0x00000001)

**public static final int ViewFlipper\_autoStart**

When true, automatically start animating

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [autoStart](#)  
[\(/reference/android/R.attr.html#autoStart\)](#).

Constant Value: 1 (0x00000001)

**public static final int ViewFlipper\_flipInterval**

This symbol is the offset where the [flipInterval](#) [\(/reference/android/R.attr.html#flipInterval\)](#) attribute's value can be found in the [ViewFlipper](#) [\(/reference/android/R.styleable.html#ViewFlipper\)](#) array.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

Constant Value: 0 (0x00000000)

**public static final int ViewGroup\_Layout\_layout\_height**

Specifies the basic height of the view. This is a required attribute for any view inside of a containing layout manager. Its value may be a dimension (such as "12dip") for a constant height or one of the special constants.

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

May be one of the following constant values.

Constant	Value	Description
fill_parent	-1	The view should be as big as its parent (minus padding). This constant is deprecated starting from API Level 8 and is replaced by <code>match_parent</code> .
match_parent	-1	The view should be as big as its parent (minus padding). Introduced in API Level 8.
wrap_content	-2	The view should be only big enough to enclose its content (plus padding).

This corresponds to the global attribute resource symbol [layout\\_height](#)  
[\(/reference/android/R.attr.html#layout\\_height\)](#).

Constant Value: 1 (0x00000001)

**public static final int ViewGroup\_Layout\_layout\_width**

Specifies the basic width of the view. This is a required attribute for any view inside of a containing layout manager. Its value may be a dimension (such as "12dip") for a constant width or one of the special

constants.

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

May be one of the following constant values.

Constant	Value	Description
fill_parent	-1	The view should be as big as its parent (minus padding). This constant is deprecated starting from API Level 8 and is replaced by <code>match_parent</code> .
match_parent	-1	The view should be as big as its parent (minus padding). Introduced in API Level 8.
wrap_content	-2	The view should be only big enough to enclose its content (plus padding).

This corresponds to the global attribute resource symbol `layout_width`  
[\(/reference/android/R.attr.html#layout\\_width\)](#).

Constant Value: 0 (0x00000000)

#### **public static final int ViewGroup.MarginLayout.layout\_height**

Specifies the basic height of the view. This is a required attribute for any view inside of a containing layout manager. Its value may be a dimension (such as "12dip") for a constant height or one of the special constants.

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

May be one of the following constant values.

Constant	Value	Description
fill_parent	-1	The view should be as big as its parent (minus padding). This constant is deprecated starting from API Level 8 and is replaced by <code>match_parent</code> .
match_parent	-1	The view should be as big as its parent (minus padding). Introduced in API Level 8.
wrap_content	-2	The view should be only big enough to enclose its content (plus padding).

This corresponds to the global attribute resource symbol `layout_height`  
[\(/reference/android/R.attr.html#layout\\_height\)](#).

Constant Value: 1 (0x00000001)

#### **public static final int ViewGroup.MarginLayout.layout\_margin**

Specifies extra space on the left, top, right and bottom sides of this view. This space is outside this view's bounds.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol `layout_margin`  
[\(/reference/android/R.attr.html#layout\\_margin\)](#).

Constant Value: 2 (0x00000002)

#### **public static final int ViewGroup.MarginLayout.layout\_marginBottom**

Specifies extra space on the bottom side of this view. This space is outside this view's bounds.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_marginBottom](#)  
(/reference/android/R.attr.html#layout\_marginBottom).

Constant Value: 6 (0x00000006)

#### **public static final int ViewGroup.MarginLayout.layout\_marginEnd**

Specifies extra space on the end side of this view. This space is outside this view's bounds.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_marginEnd](#)  
(/reference/android/R.attr.html#layout\_marginEnd).

Constant Value: 8 (0x00000008)

#### **public static final int ViewGroup.MarginLayout.layout\_marginLeft**

Specifies extra space on the left side of this view. This space is outside this view's bounds.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_marginLeft](#)  
(/reference/android/R.attr.html#layout\_marginLeft).

Constant Value: 3 (0x00000003)

#### **public static final int ViewGroup.MarginLayout.layout\_marginRight**

Specifies extra space on the right side of this view. This space is outside this view's bounds.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_marginRight](#)  
(/reference/android/R.attr.html#layout\_marginRight).

Constant Value: 5 (0x00000005)

#### **public static final int ViewGroup.MarginLayout.layout\_marginStart**

Specifies extra space on the start side of this view. This space is outside this view's bounds.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_marginStart](#)  
(/reference/android/R.attr.html#layout\_marginStart).

Constant Value: 7 (0x00000007)

#### **public static final int ViewGroup.MarginLayout.layout\_marginTop**

Specifies extra space on the top side of this view. This space is outside this view's bounds.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in

the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [layout\\_marginTop](#) ([/reference/android/R.attr.html#layout\\_marginTop](#)).

Constant Value: 4 (0x00000004)

#### **public static final int ViewGroup.MarginLayout.layout\_width**

Specifies the basic width of the view. This is a required attribute for any view inside of a containing layout manager. Its value may be a dimension (such as "12dip") for a constant width or one of the special constants.

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

May be one of the following constant values.

Constant	Value	Description
fill_parent	-1	The view should be as big as its parent (minus padding). This constant is deprecated starting from API Level 8 and is replaced by match_parent.
match_parent	-1	The view should be as big as its parent (minus padding). Introduced in API Level 8.
wrap_content	-2	The view should be only big enough to enclose its content (plus padding).

This corresponds to the global attribute resource symbol [layout\\_width](#) ([/reference/android/R.attr.html#layout\\_width](#)).

Constant Value: 0 (0x00000000)

#### **public static final int ViewGroup.addStatesFromChildren**

Sets whether this ViewGroup's drawable states also include its children's drawable states. This is used, for example, to make a group appear to be focused when its child EditText or button is focused.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [addStatesFromChildren](#) ([/reference/android/R.attr.html#addStatesFromChildren](#)).

Constant Value: 6 (0x00000006)

#### **public static final int ViewGroup.alwaysDrawnWithCache**

Defines whether the ViewGroup should always draw its children using their drawing cache or not. The default value is true.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [alwaysDrawnWithCache](#) ([/reference/android/R.attr.html#alwaysDrawnWithCache](#)).

Constant Value: 5 (0x00000005)

#### **public static final int ViewGroup.animateLayoutChanges**

Defines whether changes in layout (caused by adding and removing items) should cause a LayoutTransition to run. When this flag is set to true, a default LayoutTransition object will be set on the ViewGroup container and default animations will run when these layout changes occur.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [animateLayoutChanges](#) ([/reference/android/R.attr.html#animateLayoutChanges](#)).

Constant Value: 9 (0x00000009)

## **public static final int ViewGroup\_animationCache**

Defines whether layout animations should create a drawing cache for their children. Enabling the animation cache consumes more memory and requires a longer initialization but provides better performance. The animation cache is enabled by default.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [animationCache](#)  
(/reference/android/R.attr.html#animationCache).

Constant Value: 3 (0x00000003)

## **public static final int ViewGroup\_clipChildren**

Defines whether a child is limited to draw inside of its bounds or not. This is useful with animations that scale the size of the children to more than 100% for instance. In such a case, this property should be set to false to allow the children to draw outside of their bounds. The default value of this property is true.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [clipChildren](#)  
(/reference/android/R.attr.html#clipChildren).

Constant Value: 0 (0x00000000)

## **public static final int ViewGroup\_clipToPadding**

Defines whether the ViewGroup will clip its drawing surface so as to exclude the padding area. This property is set to true by default.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [clipToPadding](#)  
(/reference/android/R.attr.html#clipToPadding).

Constant Value: 1 (0x00000001)

## **public static final int ViewGroup\_descendantFocusability**

Defines the relationship between the ViewGroup and its descendants when looking for a View to take focus.

Must be one of the following constant values.

Constant	Value	Description
beforeDescendants	0	The ViewGroup will get focus before any of its descendants.
afterDescendants	1	The ViewGroup will get focus only if none of its descendants want it.
blocksDescendants	2	The ViewGroup will block its descendants from receiving focus.

This corresponds to the global attribute resource symbol [descendantFocusability](#)  
(/reference/android/R.attr.html#descendantFocusability).

Constant Value: 7 (0x00000007)

## **public static final int ViewGroup\_layoutAnimation**

Defines the layout animation to use the first time the ViewGroup is laid out. Layout animations can also be started manually after the first layout.

Must be a reference to another resource, in the form "[+][ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [layoutAnimation](#)  
(/reference/android/R.attr.html#layoutAnimation).

Constant Value: 2 (0x00000002)

## **public static final int ViewGroup\_persistentDrawingCache**

Defines the persistence of the drawing cache. The drawing cache might be enabled by a ViewGroup for all its children in specific situations (for instance during a scrolling.) This property lets you persist the cache in memory after its initial usage. Persisting the cache consumes more memory but may prevent frequent garbage collection if the cache is created over and over again. By default the persistence is set to scrolling.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
none	0x0	The drawing cache is not persisted after use.
animation	0x1	The drawing cache is persisted after a layout animation.
scrolling	0x2	The drawing cache is persisted after a scroll.
all	0x3	The drawing cache is always persisted.

This corresponds to the global attribute resource symbol [persistentDrawingCache](#)  
[\(/reference/android/R.attr.html#persistentDrawingCache\)](#).

Constant Value: 4 (0x00000004)

#### public static final int **ViewGroup\_splitMotionEvents**

Sets whether this ViewGroup should split MotionEvents to separate child views during touch event dispatch. If false (default), touch events will be dispatched to the child view where the first pointer went down until the last pointer goes up. If true, touch events may be dispatched to multiple children. MotionEvents for each pointer will be dispatched to the child view where the initial ACTION\_DOWN event happened. See [setMotionEventSplittingEnabled\(boolean\)](#)  
[\(/reference/android/view/ViewGroup.html#setMotionEventSplittingEnabled\(boolean\)\)](#) for more information.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [splitMotionEvents](#)  
[\(/reference/android/R.attr.html#splitMotionEvents\)](#).

Constant Value: 8 (0x00000008)

#### public static final int **ViewStub\_inflatedId**

Overrides the id of the inflated View with this value.

Must be a reference to another resource, in the form "@[+] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [inflatedId](#)  
[\(/reference/android/R.attr.html#inflatedId\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **ViewStub\_layout**

Supply an identifier for the layout resource to inflate when the ViewStub becomes visible or when forced to do so. The layout resource must be a valid reference to a layout.

Must be a reference to another resource, in the form "@[+] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [layout](#)  
[\(/reference/android/R.attr.html#layout\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **View\_alpha**

alpha property of the view, as a value between 0 (completely transparent) and 1 (completely opaque).

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [alpha](#)  
[\(/reference/android/R.attr.html#alpha\)](#).

Constant Value: 47 (0x0000002f)

## **public static final int View\_background**

A drawable to use as the background. This can be either a reference to a full drawable resource (such as a PNG image, 9-patch, XML state list description, etc), or a solid color such as "#ff000000" (black).

May be a reference to another resource, in the form "@[+][ package: ] type: name" or to a theme attribute in the form "?[ package: ][ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [background](#)  
[\(/reference/android/R.attr.html#background\)](#).

Constant Value: 12 (0x0000000c)

## **public static final int View\_clickable**

Defines whether this view reacts to click events.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ][ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [clickable](#)  
[\(/reference/android/R.attr.html#clickable\)](#).

Constant Value: 29 (0x0000001d)

## **public static final int View\_contentDescription**

Defines text that briefly describes content of the view. This property is used primarily for accessibility. Since some views do not have textual representation this attribute can be used for providing such.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ][ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [contentDescription](#)  
[\(/reference/android/R.attr.html#contentDescription\)](#).

Constant Value: 41 (0x00000029)

## **public static final int View\_drawingCacheQuality**

Defines the quality of translucent drawing caches. This property is used only when the drawing cache is enabled and translucent. The default value is auto.

Must be one of the following constant values.

Constant Value	Description
auto 0	Lets the framework decide what quality level should be used for the drawing cache.
low 1	Low quality. When set to low quality, the drawing cache uses a lower color depth, thus losing precision in rendering gradients, but uses less memory.
high 2	High quality. When set to high quality, the drawing cache uses a higher color depth but uses more memory.

This corresponds to the global attribute resource symbol [drawingCacheQuality](#)  
[\(/reference/android/R.attr.html#drawingCacheQuality\)](#).

Constant Value: 32 (0x00000020)

## **public static final int View\_duplicateParentState**

When this attribute is set to true, the view gets its drawable state (focused, pressed, etc.) from its direct parent rather than from itself.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ][ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [duplicateParentState](#)  
[\(/reference/android/R.attr.html#duplicateParentState\)](#).

Constant Value: 33 (0x00000021)

## **public static final int View\_fadeScrollbars**

Defines whether to fade out scrollbars when they are not in use.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [fadeScrollbars](#)  
(/reference/android/R.attr.html#fadeScrollbars).

Constant Value: 44 (0x0000002c)

## **public static final int View\_fadingEdge**

This attribute is deprecated and will be ignored as of API level 14 ([ICE\\_CREAM\\_SANDWICH](#)  
(/reference/android/os/Build.VERSION\_CODES.html#ICE\_CREAM\_SANDWICH)). Using fading edges may introduce noticeable performance degradations and should be used only when required by the application's visual design. To request fading edges with API level 14 and above, use the android:requiresFadingEdge attribute instead.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
none	0x00000000	No edge is faded.
horizontal	0x00001000	Fades horizontal edges only.
vertical	0x00002000	Fades vertical edges only.

This corresponds to the global attribute resource symbol [fadingEdge](#)  
(/reference/android/R.attr.html#fadingEdge).

Constant Value: 23 (0x00000017)

## **public static final int View\_fadingEdgeLength**

Defines the length of the fading edges.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [fadingEdgeLength](#)  
(/reference/android/R.attr.html#fadingEdgeLength).

Constant Value: 24 (0x00000018)

## **public static final int View\_filterTouchesWhenObscured**

Specifies whether to filter touches when the view's window is obscured by another visible window. When set to true, the view will not receive touches whenever a toast, dialog or other window appears above the view's window. Refer to the [View](#) (/reference/android/view/View.html) security documentation for more details.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [filterTouchesWhenObscured](#)  
(/reference/android/R.attr.html#filterTouchesWhenObscured).

Constant Value: 46 (0x0000002e)

## **public static final int View.fitsSystemWindows**

Boolean internal attribute to adjust view layout based on system windows such as the status bar. If true, adjusts the padding of this view to leave space for the system windows. Will only take effect if this view is in a non-embedded activity.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [fitsSystemWindows](#)  
[\(/reference/android/R.attr.html#fitsSystemWindows\)](#).

Constant Value: 21 (0x00000015)

#### **public static final int View\_focusable**

Boolean that controls whether a view can take focus. By default the user can not move focus to a view; by setting this attribute to true the view is allowed to take focus. This value does not impact the behavior of directly calling [requestFocus\(\)](#) ([/reference/android/view/View.html#requestFocus\(\)](#)), which will always request focus regardless of this view. It only impacts where focus navigation will try to move focus.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [focusable](#)  
[\(/reference/android/R.attr.html#focusable\)](#).

Constant Value: 18 (0x00000012)

#### **public static final int View\_focusableInTouchMode**

Boolean that controls whether a view can take focus while in touch mode. If this is true for a view, that view can gain focus when clicked on, and can keep focus if another view is clicked on that doesn't have this attribute set to true.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [focusableInTouchMode](#)  
[\(/reference/android/R.attr.html#focusableInTouchMode\)](#).

Constant Value: 19 (0x00000013)

#### **public static final int View\_hapticFeedbackEnabled**

Boolean that controls whether a view should have haptic feedback enabled for events such as long presses.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:][type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [hapticFeedbackEnabled](#)  
[\(/reference/android/R.attr.html#hapticFeedbackEnabled\)](#).

Constant Value: 39 (0x00000027)

#### **public static final int View\_id**

Supply an identifier name for this view, to later retrieve it with [view.findViewById\(\)](#)  
[\(/reference/android/view/View.html#findViewById\(int\)\)](#) or [Activity.findViewById\(\)](#)  
[\(/reference/android/app/Activity.html#findViewById\(int\)\)](#). This must be a resource reference; typically you set this using the @+ syntax to create a new ID resources. For example: android:id="@+id/my\_id" which allows you to later retrieve the view with findViewById(R.id.my\_id).

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [id](#) ([/reference/android/R.attr.html#id](#)).

Constant Value: 8 (0x00000008)

#### **public static final int View\_importantForAccessibility**

Controls how this View is important for accessibility which is if it fires accessibility events and if it is reported to accessibility services that query the screen. Note: While not recommended, an accessibility service may decide to ignore this attribute and operate on all views in the view tree.

May be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

May be one of the following constant values.

Constant Value	Description
auto 0	The system determines whether the view is important for accessibility - default (recommended).
yes 1	The view is important for accessibility.
no 2	The view is not important for accessibility.

This corresponds to the global attribute resource symbol [importantForAccessibility](#) [\(/reference/android/R.attr.html#importantForAccessibility\)](#).

Constant Value: 61 (0x0000003d)

#### **public static final int View\_isScrollContainer**

Set this if the view will serve as a scrolling container, meaning that it can be resized to shrink its overall window so that there will be space for an input method. If not set, the default value will be true if "scrollbars" has the vertical scrollbar set, else it will be false.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [isScrollContainer](#) [\(/reference/android/R.attr.html#isScrollContainer\)](#).

Constant Value: 38 (0x00000026)

#### **public static final int View\_keepScreenOn**

Controls whether the view's window should keep the screen on while visible.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [keepScreenOn](#) [\(/reference/android/R.attr.html#keepScreenOn\)](#).

Constant Value: 37 (0x00000025)

#### **public static final int View\_labelFor**

Specifies the id of a view for which this view serves as a label for accessibility purposes. For example, a TextView before an EditText in the UI usually specifies what information is contained in the EditText. Hence, the TextView is a label for the EditText.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [labelFor](#) [\(/reference/android/R.attr.html#labelFor\)](#).

Constant Value: 67 (0x00000043)

#### **public static final int View\_layerType**

Specifies the type of layer backing this view. The default value is none. Refer to [setLayerType\(int, android.graphics.Paint\)](#) [\(/reference/android/view/View.html#setLayerType\(int, android.graphics.Paint\)\)](#) for more information.

Must be one of the following constant values.

Constant Value	Description
none 0	Don't use a layer.
software 1	Use a software layer. Refer to <a href="#">setLayerType(int, android.graphics.Paint)</a> for more information.
hardware 2	Use a hardware layer. Refer to <a href="#">setLayerType(int, android.graphics.Paint)</a> for more information.

This corresponds to the global attribute resource symbol [layerType](#)  
(/reference/android/R.attr.html#layerType).

Constant Value: 59 (0x0000003b)

#### public static final int **View\_layoutDirection**

Defines the direction of layout drawing. This typically is associated with writing direction of the language script used. The possible values are "ltr" for Left-to-Right, "rtl" for Right-to-Left, "locale" and "inherit" from parent view. If there is nothing to inherit, "locale" is used. "locale" falls back to "en-US". "ltr" is the direction used in "en-US". The default for this attribute is "inherit".

Must be one of the following constant values.

Constant Value	Description
ltr	0 Left-to-Right
rtl	1 Right-to-Left
inherit	2 Inherit from parent
locale	3 Locale

This corresponds to the global attribute resource symbol [layoutDirection](#)  
(/reference/android/R.attr.html#layoutDirection).

Constant Value: 64 (0x00000040)

#### public static final int **View\_longClickable**

Defines whether this view reacts to long click events.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [longClickable](#)  
(/reference/android/R.attr.html#longClickable).

Constant Value: 30 (0x0000001e)

#### public static final int **View\_minHeight**

Defines the minimum height of the view. It is not guaranteed the view will be able to achieve this minimum height (for example, if its parent layout constrains it with less available height).

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [minHeight](#)  
(/reference/android/R.attr.html#minHeight).

Constant Value: 35 (0x00000023)

#### public static final int **View\_minWidth**

Defines the minimum width of the view. It is not guaranteed the view will be able to achieve this minimum width (for example, if its parent layout constrains it with less available width).

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [minWidth](#)  
(/reference/android/R.attr.html#minWidth).

Constant Value: 34 (0x00000022)

#### public static final int **View\_nextFocusDown**

Defines the next view to give focus to when the next focus is [FOCUS\\_DOWN](#)  
(/reference/android/view/View.html#FOCUS\_DOWN). If the reference refers to a view that does not exist or is

part of a hierarchy that is invisible, a [RuntimeException](#)  
(/reference/java/lang/RuntimeException.html) will result when the reference is accessed.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute  
in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [nextFocusDown](#)  
(/reference/android/R.attr.html#nextFocusDown).

Constant Value: 28 (0x00000001c)

#### public static final int **View.nextFocusForward**

Defines the next view to give focus to when the next focus is [FOCUS\\_FORWARD](#)  
(/reference/android/view/View.html#FOCUS\_FORWARD). If the reference refers to a view that does not exist  
or is part of a hierarchy that is invisible, a [RuntimeException](#)  
(/reference/java/lang/RuntimeException.html) will result when the reference is accessed.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute  
in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [nextFocusForward](#)  
(/reference/android/R.attr.html#nextFocusForward).

Constant Value: 58 (0x00000003a)

#### public static final int **View.nextFocusLeft**

Defines the next view to give focus to when the next focus is [FOCUS\\_LEFT](#)  
(/reference/android/view/View.html#FOCUS\_LEFT). If the reference refers to a view that does not exist or  
is part of a hierarchy that is invisible, a [RuntimeException](#)  
(/reference/java/lang/RuntimeException.html) will result when the reference is accessed.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute  
in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [nextFocusLeft](#)  
(/reference/android/R.attr.html#nextFocusLeft).

Constant Value: 25 (0x000000019)

#### public static final int **View.nextFocusRight**

Defines the next view to give focus to when the next focus is [FOCUS\\_RIGHT](#)  
(/reference/android/view/View.html#FOCUS\_RIGHT). If the reference refers to a view that does not exist or  
is part of a hierarchy that is invisible, a [RuntimeException](#)  
(/reference/java/lang/RuntimeException.html) will result when the reference is accessed.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute  
in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [nextFocusRight](#)  
(/reference/android/R.attr.html#nextFocusRight).

Constant Value: 26 (0x00000001a)

#### public static final int **View.nextFocusUp**

Defines the next view to give focus to when the next focus is [FOCUS\\_UP](#)  
(/reference/android/view/View.html#FOCUS\_UP). If the reference refers to a view that does not exist or is  
part of a hierarchy that is invisible, a [RuntimeException](#)  
(/reference/java/lang/RuntimeException.html) will result when the reference is accessed.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute  
in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [nextFocusUp](#)  
(/reference/android/R.attr.html#nextFocusUp).

Constant Value: 27 (0x00000001b)

#### public static final int **View.onClick**

Name of the method in this View's context to invoke when the view is clicked. This name must correspond  
to a public method that takes exactly one parameter of type View. For instance, if you specify  
android:onClick="sayHello", you must declare a public void sayHello(View v) method of your

context (typically, your Activity).

Must be a string value, using '\\.' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [onClick](#)  
(/reference/android/R.attr.html#onClick).

Constant Value: 40 (0x00000028)

#### public static final int **View\_overScrollMode**

Defines over-scrolling behavior. This property is used only if the View is scrollable. Over-scrolling is the ability for the user to receive feedback when attempting to scroll beyond meaningful content.

Must be one of the following constant values.

Constant	Value	Description
always	0	Always show over-scroll effects, even if the content fits entirely within the available space.
ifContentScrolls	1	Only show over-scroll effects if the content is large enough to meaningfully scroll.
never	2	Never show over-scroll effects.

This corresponds to the global attribute resource symbol [overScrollMode](#)  
(/reference/android/R.attr.html#overScrollMode).

Constant Value: 45 (0x0000002d)

#### public static final int **View\_padding**

Sets the padding, in pixels, of all four edges. Padding is defined as space between the edges of the view and the view's content. A views size will include it's padding. If a [background](#)  
(/reference/android/R.attr.html#background) is provided, the padding will initially be set to that (0 if the drawable does not have padding). Explicitly setting a padding value will override the corresponding padding found in the background.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [padding](#)  
(/reference/android/R.attr.html#padding).

Constant Value: 13 (0x0000000d)

#### public static final int **View\_paddingBottom**

Sets the padding, in pixels, of the bottom edge; see [padding](#)  
(/reference/android/R.attr.html#padding).

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [paddingBottom](#)  
(/reference/android/R.attr.html#paddingBottom).

Constant Value: 17 (0x00000011)

#### public static final int **View\_paddingEnd**

Sets the padding, in pixels, of the end edge; see [padding](#) (/reference/android/R.attr.html#padding).

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [paddingEnd](#)  
[\(/reference/android/R.attr.html#paddingEnd\)](#).

Constant Value: 66 (0x00000042)

#### public static final int **View\_paddingLeft**

Sets the padding, in pixels, of the left edge; see [padding](#) [\(/reference/android/R.attr.html#padding\)](#).

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [paddingLeft](#)  
[\(/reference/android/R.attr.html#paddingLeft\)](#).

Constant Value: 14 (0x0000000e)

#### public static final int **View\_paddingRight**

Sets the padding, in pixels, of the right edge; see [padding](#) [\(/reference/android/R.attr.html#padding\)](#).

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [paddingRight](#)  
[\(/reference/android/R.attr.html#paddingRight\)](#).

Constant Value: 16 (0x00000010)

#### public static final int **View\_paddingStart**

Sets the padding, in pixels, of the start edge; see [padding](#) [\(/reference/android/R.attr.html#padding\)](#).

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [paddingStart](#)  
[\(/reference/android/R.attr.html#paddingStart\)](#).

Constant Value: 65 (0x00000041)

#### public static final int **View\_paddingTop**

Sets the padding, in pixels, of the top edge; see [padding](#) [\(/reference/android/R.attr.html#padding\)](#).

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [paddingTop](#)  
[\(/reference/android/R.attr.html#paddingTop\)](#).

Constant Value: 15 (0x0000000f)

#### public static final int **View\_requiresFadingEdge**

Defines which edges should be faded on scrolling.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
none	0x00000000	No edge is faded.

horizontal 0x00001000 Fades horizontal edges only.  
vertical 0x00002000 Fades vertical edges only.

This corresponds to the global attribute resource symbol [requiresFadingEdge](#)  
[\(/reference/android/R.attr.html#requiresFadingEdge\)](#).

Constant Value: 60 (0x0000003c)

#### public static final int **View\_rotation**

rotation of the view, in degrees.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [rotation](#)  
[\(/reference/android/R.attr.html#rotation\)](#).

Constant Value: 54 (0x00000036)

#### public static final int **View\_rotationX**

rotation of the view around the x axis, in degrees.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [rotationX](#)  
[\(/reference/android/R.attr.html#rotationX\)](#).

Constant Value: 55 (0x00000037)

#### public static final int **View\_rotationY**

rotation of the view around the y axis, in degrees.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [rotationY](#)  
[\(/reference/android/R.attr.html#rotationY\)](#).

Constant Value: 56 (0x00000038)

#### public static final int **View\_saveEnabled**

If unset, no state will be saved for this view when it is being frozen. The default is true, allowing the view to be saved (however it also must have an ID assigned to it for its state to be saved). Setting this to false only disables the state for this view, not for its children which may still be saved.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [saveEnabled](#)  
[\(/reference/android/R.attr.html#saveEnabled\)](#).

Constant Value: 31 (0x0000001f)

#### public static final int **View\_scaleX**

scale of the view in the x direction.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[package:]type:name") or theme attribute (in the form "?[package:] [type:]name") containing a value of this type.

This corresponds to the global attribute resource symbol [scaleX](#)  
[\(/reference/android/R.attr.html#scaleX\)](#).

Constant Value: 52 (0x00000034)

## **public static final int View\_scaleY**

scale of the view in the y direction.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [scaleY](#)  
[\(/reference/android/R.attr.html#scaleY\)](#).

Constant Value: 53 (0x00000035)

## **public static final int View\_scrollX**

The initial horizontal scroll offset, in pixels.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [scrollX](#)  
[\(/reference/android/R.attr.html#scrollX\)](#).

Constant Value: 10 (0x0000000a)

## **public static final int View\_scrollY**

The initial vertical scroll offset, in pixels.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [scrollY](#)  
[\(/reference/android/R.attr.html#scrollY\)](#).

Constant Value: 11 (0x0000000b)

## **public static final int View\_scrollbarAlwaysDrawHorizontalTrack**

Defines whether the horizontal scrollbar track should always be drawn.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [scrollbarAlwaysDrawHorizontalTrack](#)  
[\(/reference/android/R.attr.html#scrollbarAlwaysDrawHorizontalTrack\)](#).

Constant Value: 5 (0x00000005)

## **public static final int View\_scrollbarAlwaysDrawVerticalTrack**

Defines whether the vertical scrollbar track should always be drawn.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [scrollbarAlwaysDrawVerticalTrack](#)  
[\(/reference/android/R.attr.html#scrollbarAlwaysDrawVerticalTrack\)](#).

Constant Value: 6 (0x00000006)

## **public static final int View\_scrollbarDefaultDelayBeforeFade**

Defines the delay in milliseconds that a scrollbar waits before fade out.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [scrollbarDefaultDelayBeforeFade](#)  
(/reference/android/R.attr.html#scrollbarDefaultDelayBeforeFade).

Constant Value: 43 (0x0000002b)

#### public static final int **View\_scrollbarFadeDuration**

Defines the delay in milliseconds that a scrollbar takes to fade out.

Must be an integer value, such as "100".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [scrollbarFadeDuration](#)  
(/reference/android/R.attr.html#scrollbarFadeDuration).

Constant Value: 42 (0x0000002a)

#### public static final int **View\_scrollbarSize**

Sets the width of vertical scrollbars and height of horizontal scrollbars.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [scrollbarSize](#)  
(/reference/android/R.attr.html#scrollbarSize).

Constant Value: 0 (0x00000000)

#### public static final int **View\_scrollbarStyle**

Controls the scrollbar style and position. The scrollbars can be overlaid or inset. When inset, they add to the padding of the view. And the scrollbars can be drawn inside the padding area or on the edge of the view. For example, if a view has a background drawable and you want to draw the scrollbars inside the padding specified by the drawable, you can use insideOverlay or insideInset. If you want them to appear at the edge of the view, ignoring the padding, then you can use outsideOverlay or outsideInset.

Must be one of the following constant values.

Constant	Value	Description
insideOverlay	0x0	Inside the padding and overlaid
insideInset	0x01000000	Inside the padding and inset
outsideOverlay	0x02000000	Edge of the view and overlaid
outsideInset	0x03000000	Edge of the view and inset

This corresponds to the global attribute resource symbol [scrollbarStyle](#)  
(/reference/android/R.attr.html#scrollbarStyle).

Constant Value: 7 (0x00000007)

#### public static final int **View\_scrollbarThumbHorizontal**

Defines the horizontal scrollbar thumb drawable.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [scrollbarThumbHorizontal](#)  
(/reference/android/R.attr.html#scrollbarThumbHorizontal).

Constant Value: 1 (0x00000001)

#### public static final int **View\_scrollbarThumbVertical**

Defines the vertical scrollbar thumb drawable.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [scrollbarThumbVertical](#)  
(/reference/android/R.attr.html#scrollbarThumbVertical).

Constant Value: 2 (0x00000002)

#### public static final int **View\_scrollbarTrackHorizontal**

Defines the horizontal scrollbar track drawable.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [scrollbarTrackHorizontal](#)  
(/reference/android/R.attr.html#scrollbarTrackHorizontal).

Constant Value: 3 (0x00000003)

#### public static final int **View\_scrollbarTrackVertical**

Defines the vertical scrollbar track drawable.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [scrollbarTrackVertical](#)  
(/reference/android/R.attr.html#scrollbarTrackVertical).

Constant Value: 4 (0x00000004)

#### public static final int **View\_scrollbars**

Defines which scrollbars should be displayed on scrolling or not.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
none	0x00000000	No scrollbar is displayed.
horizontal	0x00000100	Displays horizontal scrollbar only.
vertical	0x00000200	Displays vertical scrollbar only.

This corresponds to the global attribute resource symbol [scrollbars](#)  
(/reference/android/R.attr.html#scrollbars).

Constant Value: 22 (0x00000016)

#### public static final int **View\_soundEffectsEnabled**

Boolean that controls whether a view should have sound effects enabled for events such as clicking and touching.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [soundEffectsEnabled](#)  
(/reference/android/R.attr.html#soundEffectsEnabled).

Constant Value: 36 (0x00000024)

#### public static final int **View\_tag**

Supply a tag for this view containing a String, to be retrieved later with [View.getTag\(\)](#)  
(/reference/android/view/View.html#getTag()) or searched for with [View.findViewWithTag\(\)](#)  
(/reference/android/view/View.html#findViewWithTag(java.lang.Object)). It is generally preferable to use IDs (through the android:id attribute) instead of tags because they are faster and allow for compile-time type checking.

Must be a string value, using '\\\' to escape characters such as '\\n' or '\\xxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [tag](#) (/reference/android/R.attr.html#tag).

Constant Value: 9 (0x00000009)

## **public static final int View.textAlignment**

Defines the alignment of the text. A heuristic is used to determine the resolved text alignment.

May be an integer value, such as "100".

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ] [ type: ] name`") containing a value of this type.

May be one of the following constant values.

Constant	Value	Description
inherit	0	Default
gravity	1	Default for the root view. The gravity determines the alignment, ALIGN_NORMAL, ALIGN_CENTER, or ALIGN_OPPOSITE, which are relative to each paragraph's text direction
textStart	2	Align to the start of the paragraph, e.g. ALIGN_NORMAL.
textEnd	3	Align to the end of the paragraph, e.g. ALIGN_OPPOSITE.
center	4	Center the paragraph, e.g. ALIGN_CENTER.
viewStart	5	Align to the start of the view, which is ALIGN_LEFT if the view's resolved layoutDirection is LTR, and ALIGN_RIGHT otherwise.
viewEnd	6	Align to the end of the view, which is ALIGN_RIGHT if the view's resolved layoutDirection is LTR, and ALIGN_LEFT otherwise

This corresponds to the global attribute resource symbol [textAlignment](#)  
[\(/reference/android/R.attr.html#textAlignment\)](#).

Constant Value: 63 (0x0000003f)

## **public static final int View.textDirection**

Defines the direction of the text. A heuristic is used to determine the resolved text direction of paragraphs.

May be an integer value, such as "100".

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ] [ type: ] name`") containing a value of this type.

May be one of the following constant values.

Constant	Value	Description
inherit	0	Default
firstStrong	1	Default for the root view. The first strong directional character determines the paragraph direction. If there is no strong directional character, the paragraph direction is the view's resolved layout direction.
anyRtl	2	The paragraph direction is RTL if it contains any strong RTL character, otherwise it is LTR if it contains any strong LTR characters. If there are neither, the paragraph direction is the view's resolved layout direction.
ltr	3	The paragraph direction is left to right.
rtl	4	The paragraph direction is right to left.
locale	5	The paragraph direction is coming from the system Locale.

This corresponds to the global attribute resource symbol [textDirection](#)  
[\(/reference/android/R.attr.html#textDirection\)](#).

Constant Value: 62 (0x0000003e)

## **public static final int View.transformPivotX**

x location of the pivot point around which the view will rotate and scale. This xml attribute sets the pivotX property of the View.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp".

Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "`@[ package: ] type: name`") or theme attribute (in the form "`?[ package: ] [ type: ] name`") containing a value of this type.

This corresponds to the global attribute resource symbol [transformPivotX](#)  
[\(/reference/android/R.attr.html#transformPivotX\)](#).

Constant Value: 48 (0x00000030)

## **public static final int View.transformPivotY**

y location of the pivot point around which the view will rotate and scale. This xml attribute sets the pivotY property of the View.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [transformPivotY](#)  
(/reference/android/R.attr.html#transformPivotY).

Constant Value: 49 (0x00000031)

#### public static final int **View\_translationX**

translation in x of the view. This value is added post-layout to the left property of the view, which is set by its layout.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [translationX](#)  
(/reference/android/R.attr.html#translationX).

Constant Value: 50 (0x00000032)

#### public static final int **View\_translationY**

translation in y of the view. This value is added post-layout to the left property of the view, which is set by its layout.

Must be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?@[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [translationY](#)  
(/reference/android/R.attr.html#translationY).

Constant Value: 51 (0x00000033)

#### public static final int **View\_verticalScrollbarPosition**

Determines which side the vertical scroll bar should be placed on.

Must be one of the following constant values.

Constant	Value	Description
defaultPosition	0	Place the scroll bar wherever the system default determines.
left	1	Place the scroll bar on the left.
right	2	Place the scroll bar on the right.

This corresponds to the global attribute resource symbol [verticalScrollbarPosition](#)  
(/reference/android/R.attr.html#verticalScrollbarPosition).

Constant Value: 57 (0x00000039)

#### public static final int **View\_visibility**

Controls the initial visibility of the view.

Must be one of the following constant values.

Constant	Value	Description
visible	0	Visible on screen; the default value.
invisible	1	Not displayed, but taken into account during layout (space is left for it).
gone	2	Completely hidden, as if the view had not been added.

This corresponds to the global attribute resource symbol [visibility](#)  
(/reference/android/R.attr.html#visibility).

Constant Value: 20 (0x00000014)

#### public static final int **VolumePreference\_streamType**

Different audio stream types.

Must be one of the following constant values.

##### **Constant Value Description**

voice	0
system	1
ring	2
music	3
alarm	4

This corresponds to the global attribute resource symbol [streamType](#)  
[\(/reference/android/R.attr.html#streamType\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **WallpaperPreviewInfo\_staticWallpaperPreview**

A resource id of a static drawable.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [staticWallpaperPreview](#)  
[\(/reference/android/R.attr.html#staticWallpaperPreview\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **Wallpaper\_author**

Name of the author of this component, e.g. Google.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [author](#)  
[\(/reference/android/R.attr.html#author\)](#).

Constant Value: 3 (0x00000003)

#### public static final int **Wallpaper\_description**

Short description of the component's purpose or behavior.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [description](#)  
[\(/reference/android/R.attr.html#description\)](#).

Constant Value: 0 (0x00000000)

#### public static final int **Wallpaper\_settingsActivity**

Component name of an activity that allows the user to modify the settings for this service.

Must be a string value, using '\\;' to escape characters such as '\\n' or '\\uxxx' for a unicode character.

This may also be a reference to a resource (in the form "@[ package: ]type: name") or theme attribute (in the form "?[ package: ][ type: ]name") containing a value of this type.

This corresponds to the global attribute resource symbol [settingsActivity](#)  
[\(/reference/android/R.attr.html#settingsActivity\)](#).

Constant Value: 1 (0x00000001)

#### public static final int **Wallpaper\_thumbnail**

Reference to a the wallpaper's thumbnail bitmap.

Must be a reference to another resource, in the form "@[+][ package: ]type: name" or to a theme attribute in the form "?[ package: ][ type: ]name".

This corresponds to the global attribute resource symbol [thumbnail](#)  
[\(/reference/android/R.attr.html#thumbnail\)](#).

Constant Value: 2 (0x00000002)

#### **public static final int WindowAnimation\_activityCloseEnterAnimation**

When closing the current activity, this is the animation that is run on the next activity (which is entering the screen).

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [activityCloseEnterAnimation](#)  
[\(/reference/android/R.attr.html#activityCloseEnterAnimation\)](#).

Constant Value: 6 (0x00000006)

#### **public static final int WindowAnimation\_activityCloseExitAnimation**

When closing the current activity, this is the animation that is run on the current activity (which is exiting the screen).

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [activityCloseExitAnimation](#)  
[\(/reference/android/R.attr.html#activityCloseExitAnimation\)](#).

Constant Value: 7 (0x00000007)

#### **public static final int WindowAnimation\_activityOpenEnterAnimation**

When opening a new activity, this is the animation that is run on the next activity (which is entering the screen).

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [activityOpenEnterAnimation](#)  
[\(/reference/android/R.attr.html#activityOpenEnterAnimation\)](#).

Constant Value: 4 (0x00000004)

#### **public static final int WindowAnimation\_activityOpenExitAnimation**

When opening a new activity, this is the animation that is run on the previous activity (which is exiting the screen).

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [activityOpenExitAnimation](#)  
[\(/reference/android/R.attr.html#activityOpenExitAnimation\)](#).

Constant Value: 5 (0x00000005)

#### **public static final int WindowAnimation\_taskCloseEnterAnimation**

When closing the last activity of a task, this is the animation that is run on the activity of the next task (which is entering the screen).

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [taskCloseEnterAnimation](#)  
[\(/reference/android/R.attr.html#taskCloseEnterAnimation\)](#).

Constant Value: 10 (0x0000000a)

#### **public static final int WindowAnimation\_taskCloseExitAnimation**

When opening an activity in a new task, this is the animation that is run on the activity of the old task (which is exiting the screen).

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [taskCloseExitAnimation](#)  
(/reference/android/R.attr.html#taskCloseExitAnimation).

Constant Value: 11 (0x0000000b)

#### **public static final int WindowAnimation\_taskOpenEnterAnimation**

When opening an activity in a new task, this is the animation that is run on the activity of the new task  
(which is entering the screen).

Must be a reference to another resource, in the form "@[+][package: ]type: name" or to a theme attribute  
in the form "?[package: ][type: ]name".

This corresponds to the global attribute resource symbol [taskOpenEnterAnimation](#)  
(/reference/android/R.attr.html#taskOpenEnterAnimation).

Constant Value: 8 (0x00000008)

#### **public static final int WindowAnimation\_taskOpenExitAnimation**

When opening an activity in a new task, this is the animation that is run on the activity of the old task  
(which is exiting the screen).

Must be a reference to another resource, in the form "@[+][package: ]type: name" or to a theme attribute  
in the form "?[package: ][type: ]name".

This corresponds to the global attribute resource symbol [taskOpenExitAnimation](#)  
(/reference/android/R.attr.html#taskOpenExitAnimation).

Constant Value: 9 (0x00000009)

#### **public static final int WindowAnimation\_taskToBackEnterAnimation**

When sending the current task to the background, this is the animation that is run on the top activity of the  
task behind it (which is entering the screen).

Must be a reference to another resource, in the form "@[+][package: ]type: name" or to a theme attribute  
in the form "?[package: ][type: ]name".

This corresponds to the global attribute resource symbol [taskToBackEnterAnimation](#)  
(/reference/android/R.attr.html#taskToBackEnterAnimation).

Constant Value: 14 (0x0000000e)

#### **public static final int WindowAnimation\_taskToBackExitAnimation**

When sending the current task to the background, this is the animation that is run on the top activity of the  
current task (which is exiting the screen).

Must be a reference to another resource, in the form "@[+][package: ]type: name" or to a theme attribute  
in the form "?[package: ][type: ]name".

This corresponds to the global attribute resource symbol [taskToBackExitAnimation](#)  
(/reference/android/R.attr.html#taskToBackExitAnimation).

Constant Value: 15 (0x0000000f)

#### **public static final int WindowAnimation\_taskToFrontEnterAnimation**

When bringing an existing task to the foreground, this is the animation that is run on the top activity of the  
task being brought to the foreground (which is entering the screen).

Must be a reference to another resource, in the form "@[+][package: ]type: name" or to a theme attribute  
in the form "?[package: ][type: ]name".

This corresponds to the global attribute resource symbol [taskToFrontEnterAnimation](#)  
(/reference/android/R.attr.html#taskToFrontEnterAnimation).

Constant Value: 12 (0x0000000c)

#### **public static final int WindowAnimation\_taskToFrontExitAnimation**

When bringing an existing task to the foreground, this is the animation that is run on the current  
foreground activity (which is exiting the screen).

Must be a reference to another resource, in the form "@[+][package: ]type: name" or to a theme attribute  
in the form "?[package: ][type: ]name".

This corresponds to the global attribute resource symbol [taskToFrontExitAnimation](#)  
(/reference/android/R.attr.html#taskToFrontExitAnimation).

Constant Value: 13 (0x0000000d)

#### **public static final int WindowAnimation\_wallpaperCloseEnterAnimation**

When opening a new activity that hides the wallpaper, while currently showing the wallpaper, this is the animation that is run on the new activity (which is entering the screen).

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [wallpaperCloseEnterAnimation](#)  
(/reference/android/R.attr.html#wallpaperCloseEnterAnimation).

Constant Value: 18 (0x00000012)

#### **public static final int WindowAnimation\_wallpaperCloseExitAnimation**

When opening a new activity that hides the wallpaper, while currently showing the wallpaper, this is the animation that is run on the old wallpaper activity (which is exiting the screen).

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [wallpaperCloseExitAnimation](#)  
(/reference/android/R.attr.html#wallpaperCloseExitAnimation).

Constant Value: 19 (0x00000013)

#### **public static final int WindowAnimation\_wallpaperIntraCloseEnterAnimation**

When closing a foreground activity that is on top of the wallpaper when the previous activity is also on top of the wallpaper, this is the animation that is run on the previous activity (which is entering the screen). The wallpaper remains static behind the animation.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [wallpaperIntraCloseEnterAnimation](#)  
(/reference/android/R.attr.html#wallpaperIntraCloseEnterAnimation).

Constant Value: 22 (0x00000016)

#### **public static final int WindowAnimation\_wallpaperIntraCloseExitAnimation**

When closing a foreground activity that is on top of the wallpaper when the previous activity is also on top of the wallpaper, this is the animation that is run on the current activity (which is exiting the screen). The wallpaper remains static behind the animation.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [wallpaperIntraCloseExitAnimation](#)  
(/reference/android/R.attr.html#wallpaperIntraCloseExitAnimation).

Constant Value: 23 (0x00000017)

#### **public static final int WindowAnimation\_wallpaperIntraOpenEnterAnimation**

When opening a new activity that is on top of the wallpaper when the current activity is also on top of the wallpaper, this is the animation that is run on the new activity (which is entering the screen). The wallpaper remains static behind the animation.

Must be a reference to another resource, in the form "@[+][package:]type:name" or to a theme attribute in the form "?[package:][type:]name".

This corresponds to the global attribute resource symbol [wallpaperIntraOpenEnterAnimation](#)  
(/reference/android/R.attr.html#wallpaperIntraOpenEnterAnimation).

Constant Value: 20 (0x00000014)

#### **public static final int WindowAnimation\_wallpaperIntraOpenExitAnimation**

When opening a new activity that is on top of the wallpaper when the current activity is also on top of the wallpaper, this is the animation that is run on the current activity (which is exiting the screen). The

wallpaper remains static behind the animation.

Must be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

This corresponds to the global attribute resource symbol [wallpaperIntraOpenExitAnimation](#) ([/reference/android/R.attr.html#wallpaperIntraOpenExitAnimation](#)).

Constant Value: 21 (0x00000015)

#### public static final int **WindowAnimation\_wallpaperOpenEnterAnimation**

When opening a new activity that shows the wallpaper, while currently not showing the wallpaper, this is the animation that is run on the new wallpaper activity (which is entering the screen).

Must be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

This corresponds to the global attribute resource symbol [wallpaperOpenEnterAnimation](#) ([/reference/android/R.attr.html#wallpaperOpenEnterAnimation](#)).

Constant Value: 16 (0x00000010)

#### public static final int **WindowAnimation\_wallpaperOpenExitAnimation**

When opening a new activity that shows the wallpaper, while currently not showing the wallpaper, this is the animation that is run on the current activity (which is exiting the screen).

Must be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

This corresponds to the global attribute resource symbol [wallpaperOpenExitAnimation](#) ([/reference/android/R.attr.html#wallpaperOpenExitAnimation](#)).

Constant Value: 17 (0x00000011)

#### public static final int **WindowAnimation\_windowEnterAnimation**

The animation used when a window is being added.

Must be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

This corresponds to the global attribute resource symbol [windowEnterAnimation](#) ([/reference/android/R.attr.html#windowEnterAnimation](#)).

Constant Value: 0 (0x00000000)

#### public static final int **WindowAnimation\_windowExitAnimation**

The animation used when a window is being removed.

Must be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

This corresponds to the global attribute resource symbol [windowExitAnimation](#) ([/reference/android/R.attr.html#windowExitAnimation](#)).

Constant Value: 1 (0x00000001)

#### public static final int **WindowAnimation\_windowHideAnimation**

The animation used when a window is going from VISIBLE to INVISIBLE.

Must be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

This corresponds to the global attribute resource symbol [windowHideAnimation](#) ([/reference/android/R.attr.html#windowHideAnimation](#)).

Constant Value: 3 (0x00000003)

#### public static final int **WindowAnimation\_windowShowAnimation**

The animation used when a window is going from INVISIBLE to VISIBLE.

Must be a reference to another resource, in the form "@[+][*package*:]*type*:*name*" or to a theme attribute in the form "?[*package*:]*type*:*name*".

This corresponds to the global attribute resource symbol [windowShowAnimation](#)  
(/reference/android/R.attr.html#windowShowAnimation).

Constant Value: 2 (0x00000002)

#### **public static final int Window\_backgroundDimAmount**

Default background dim amount when a menu, dialog, or something similar pops up.

Must be a floating point value, such as "1.2".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [backgroundDimAmount](#)  
(/reference/android/R.attr.html#backgroundDimAmount).

Constant Value: 0 (0x00000000)

#### **public static final int Window\_backgroundDimEnabled**

Control whether dimming behind the window is enabled. The default theme does not set this value, meaning it is based on whether the window is floating.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [backgroundDimEnabled](#)  
(/reference/android/R.attr.html#backgroundDimEnabled).

Constant Value: 11 (0x0000000b)

#### **public static final int Window\_textColor**

Color of text (usually same as colorForeground).

May be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

May be a color value, in the form of "#rgb", "#argb", "#rrggbb", or "#aarrggbb".

This corresponds to the global attribute resource symbol [textColor](#)  
(/reference/android/R.attr.html#textColor).

Constant Value: 7 (0x00000007)

#### **public static final int Window\_windowActionBar**

Flag indicating whether this window should have an Action Bar in place of the usual title bar.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowActionBar](#)  
(/reference/android/R.attr.html#windowActionBar).

Constant Value: 15 (0x0000000f)

#### **public static final int Window\_windowActionBarOverlay**

Flag indicating whether this window's Action Bar should overlay application content. Does nothing if the window would not have an Action Bar.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowActionBarOverlay](#)  
(/reference/android/R.attr.html#windowActionBarOverlay).

Constant Value: 17 (0x00000011)

#### **public static final int Window\_windowActionModeOverlay**

Flag indicating whether action modes should overlay window content when there is not reserved space for their UI (such as an Action Bar).

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowActionBarOverlay](#)  
(/reference/android/R.attr.html#windowActionBarOverlay).

Constant Value: 16 (0x00000010)

#### **public static final int Window\_windowAnimationStyle**

Reference to a style resource holding the set of window animations to use, which can be any of the attributes defined by [WindowAnimation](#) (/reference/android/R.styleable.html#WindowAnimation).

Must be a reference to another resource, in the form "@[ +][ package: ] type: name" or to a theme attribute in the form "?[ package: ][ type: ] name".

This corresponds to the global attribute resource symbol [windowAnimationStyle](#)  
(/reference/android/R.attr.html#windowAnimationStyle).

Constant Value: 8 (0x00000008)

#### **public static final int Window\_windowBackground**

Drawable to use as the overall window background. As of [HONEYCOMB](#)  
(/reference/android/os/Build.VERSION\_CODES.html#HONEYCOMB), this may be a selector that uses state\_accelerated to pick a non-solid color when running on devices that can draw such a bitmap with complex compositing on top at 60fps.

There are a few special considerations to use when setting this drawable:

- This information will be used to infer the pixel format for your window's surface. If the drawable has any non-opaque pixels, your window will be translucent (32 bpp).
- If you want to draw the entire background yourself, you should set this drawable to some solid color that closely matches that background (so the system's preview of your window will match), and then in code manually set your window's background to null so it will not be drawn.

Must be a reference to another resource, in the form "@[ +][ package: ] type: name" or to a theme attribute in the form "?[ package: ][ type: ] name".

This corresponds to the global attribute resource symbol [windowBackground](#)  
(/reference/android/R.attr.html#windowBackground).

Constant Value: 1 (0x00000001)

#### **public static final int Window\_windowCloseOnTouchOutside**

Control whether a container should automatically close itself if the user touches outside of it. This only applies to activities and dialogs.

Note: this attribute will only be respected for applications that are targeting [HONEYCOMB](#)  
(/reference/android/os/Build.VERSION\_CODES.html#HONEYCOMB) or later.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ][ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowCloseOnTouchOutside](#)  
(/reference/android/R.attr.html#windowCloseOnTouchOutside).

Constant Value: 21 (0x00000015)

#### **public static final int Window\_windowContentOverlay**

This Drawable is overlaid over the foreground of the Window's content area, usually to place a shadow below the title.

Must be a reference to another resource, in the form "@[ +][ package: ] type: name" or to a theme attribute in the form "?[ package: ][ type: ] name".

This corresponds to the global attribute resource symbol [windowContentOverlay](#)  
(/reference/android/R.attr.html#windowContentOverlay).

Constant Value: 6 (0x00000006)

## **public static final int Window\_windowDisablePreview**

Flag allowing you to disable the preview animation for a window. The default value is false; if set to true, the system can never use the window's theme to show a preview of it before your actual instance is shown to the user.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowDisablePreview](#)  
(/reference/android/R.attr.html#windowDisablePreview).

Constant Value: 12 (0x0000000c)

## **public static final int Window\_windowEnableSplitTouch**

Flag indicating that this window should allow touches to be split across other windows that also support split touch. The default value is true for applications with a targetSdkVersion of Honeycomb or newer; false otherwise. When this flag is false, the first pointer that goes down determines the window to which all subsequent touches go until all pointers go up. When this flag is true, each pointer (not necessarily the first) that goes down determines the window to which all subsequent touches of that pointer will go until that pointers go up thereby enabling touches with multiple pointers to be split across multiple windows.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowEnableSplitTouch](#)  
(/reference/android/R.attr.html#windowEnableSplitTouch).

Constant Value: 18 (0x00000012)

## **public static final int Window\_windowFrame**

Drawable to use as a frame around the window.

Must be a reference to another resource, in the form "@[ + ] [ package: ] type: name" or to a theme attribute in the form "?[ package: ] [ type: ] name".

This corresponds to the global attribute resource symbol [windowFrame](#)  
(/reference/android/R.attr.html#windowFrame).

Constant Value: 2 (0x00000002)

## **public static final int Window\_windowFullscreen**

Flag indicating whether this window should fill the entire screen.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowFullscreen](#)  
(/reference/android/R.attr.html#windowFullscreen).

Constant Value: 9 (0x00000009)

## **public static final int Window\_windowIsFloating**

Flag indicating whether this is a floating window.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowIsFloating](#)  
(/reference/android/R.attr.html#windowIsFloating).

Constant Value: 4 (0x00000004)

## **public static final int Window\_windowIsTranslucent**

Flag indicating whether this is a translucent window.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowIsTranslucent](#)  
(/reference/android/R.attr.html#windowIsTranslucent).

Constant Value: 5 (0x00000005)

#### public static final int **Window\_windowMinWidthMajor**

The minimum width the window is allowed to be, along the major axis of the screen. That is, when in landscape. Can be either an absolute dimension or a fraction of the screen size in that dimension.

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowMinWidthMajor](#)  
(/reference/android/R.attr.html#windowMinWidthMajor).

Constant Value: 19 (0x00000013)

#### public static final int **Window\_windowMinWidthMinor**

The minimum width the window is allowed to be, along the minor axis of the screen. That is, when in portrait. Can be either an absolute dimension or a fraction of the screen size in that dimension.

May be a dimension value, which is a floating point number appended with a unit such as "14.5sp". Available units are: px (pixels), dp (density-independent pixels), sp (scaled pixels based on preferred font size), in (inches), mm (millimeters).

May be a fractional value, which is a floating point number appended with either % or %p, such as "14.5%". The % suffix always means a percentage of the base size; the optional %p suffix provides a size relative to some parent container.

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowMinWidthMinor](#)  
(/reference/android/R.attr.html#windowMinWidthMinor).

Constant Value: 20 (0x00000014)

#### public static final int **Window\_windowNoDisplay**

Flag indicating that this window should not be displayed at all. The default value is false; if set to true, and this window is the main window of an Activity, then it will never actually be added to the window manager. This means that your activity must immediately quit without waiting for user interaction, because there will be no such interaction coming.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowNoDisplay](#)  
(/reference/android/R.attr.html#windowNoDisplay).

Constant Value: 10 (0x0000000a)

#### public static final int **Window\_windowNoTitle**

Flag indicating whether there should be no title on this window.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowNoTitle](#)

[\(/reference/android/R.attr.html#windowNoTitle\).](#)

Constant Value: 3 (0x00000003)

### public static final int **Window\_windowShowWallpaper**

Flag indicating that this window's background should be the user's current wallpaper.

Must be a boolean value, either "true" or "false".

This may also be a reference to a resource (in the form "@[ package: ] type: name") or theme attribute (in the form "?[ package: ] [ type: ] name") containing a value of this type.

This corresponds to the global attribute resource symbol [windowShowWallpaper](#)  
[\(/reference/android/R.attr.html#windowShowWallpaper\).](#)

Constant Value: 14 (0x0000000e)

### public static final int **Window\_windowSoftInputMode**

Defines the default soft input state that this window would like when it is displayed.

Must be one or more (separated by '|') of the following constant values.

Constant	Value	Description
stateUnspecified	0	Not specified, use what the system thinks is best. This is the default.
stateUnchanged	1	Leave the soft input window as-is, in whatever state it last was.
stateHidden	2	Make the soft input area hidden when normally appropriate (when the user is navigating forward to your window).
stateAlwaysHidden	3	Always make the soft input area hidden when this window has input focus.
stateVisible	4	Make the soft input area visible when normally appropriate (when the user is navigating forward to your window).
stateAlwaysVisible	5	Always make the soft input area visible when this window has input focus.
adjustUnspecified	0x00	The window resize/pan adjustment has not been specified, the system will automatically select between resize and pan modes, depending on whether the content of the window has any layout views that can scroll their contents. If there is such a view, then the window will be resized, with the assumption being that the resizable area can be reduced to make room for the input UI.
adjustResize	0x10	Always resize the window: the content area of the window is reduced to make room for the soft input area.
adjustPan	0x20	Don't resize the window to make room for the soft input area; instead pan the contents of the window as focus moves inside of it so that the user can see what they are typing. This is generally less desirable than panning because the user may need to close the input area to get at and interact with parts of the window.
adjustNothing	0x30	Don't resize or pan the window to make room for the soft input area; the window is never adjusted for it.

This corresponds to the global attribute resource symbol [windowSoftInputMode](#)  
[\(/reference/android/R.attr.html#windowSoftInputMode\).](#)

Constant Value: 13 (0x0000000d)

## Fields

---

### public static final int[] **AbsListView**

Attributes that can be used with a AbsListView.

Includes the following attributes:

Attribute	Description
<a href="#">android:cacheColorHint</a>	Indicates that this list will always be drawn on top of solid, single-color opaque background.
<a href="#">android:choiceMode</a>	Defines the choice behavior for the view.
<a href="#">android:drawSelectorOnTop</a>	When set to true, the selector will be drawn over the selected item.
<a href="#">android:fastScrollAlwaysVisible</a>	When set to true, the list will always show the fast scroll interface.
<a href="#">android:fastScrollEnabled</a>	Enables the fast scroll thumb that can be dragged to quickly

<a href="#">android:listSelector</a>	scroll through the list.
<a href="#">android:scrollingCache</a>	Drawable used to indicate the currently selected item in the list.
<a href="#">android:smoothScrollbar</a>	When set to true, the list uses a drawing cache during scrolling.
<a href="#">android:stackFromBottom</a>	When set to true, the list will use a more refined calculation method based on the pixels height of the items visible on screen.
<a href="#">android:textFilterEnabled</a>	Used by ListView and GridView to stack their content from the bottom.
<a href="#">android:transcriptMode</a>	When set to true, the list will filter results as the user types.

**See Also**

[AbsListView\\_cacheColorHint](#)  
[AbsListView\\_choiceMode](#)  
[AbsListView\\_drawSelectorOnTop](#)  
[AbsListView\\_fastScrollAlwaysVisible](#)  
[AbsListView\\_fastScrollEnabled](#)  
[AbsListView\\_listSelector](#)  
[AbsListView\\_scrollingCache](#)  
[AbsListView\\_smoothScrollbar](#)  
[AbsListView\\_stackFromBottom](#)  
[AbsListView\\_textFilterEnabled](#)  
[AbsListView\\_transcriptMode](#)

## public static final int[] AbsSpinner

Attributes that can be used with a AbsSpinner.

Includes the following attributes:

Attribute	Description
<a href="#">android:entries</a>	Reference to an array resource that will populate the Spinner.

**See Also**

[AbsSpinner\\_entries](#)

## public static final int[] AbsoluteLayout\_Layout

Attributes that can be used with a AbsoluteLayout\_Layout.

Includes the following attributes:

Attribute	Description
<a href="#">android:layout_x</a>	
<a href="#">android:layout_y</a>	

**See Also**

[AbsoluteLayout\\_layout\\_layout\\_x](#)  
[AbsoluteLayout\\_layout\\_layout\\_y](#)

## public static final int[] AccelerateInterpolator

Attributes that can be used with a AccelerateInterpolator.

Includes the following attributes:

Attribute	Description
<a href="#">android:factor</a>	This is the amount of deceleration to add when easing in.

**See Also**

[AccelerateInterpolator\\_factor](#)

## public static final int[] AccessibilityService

Use accessibility-service as the root tag of the XML resource that describes an [AccessibilityService](#) ([/reference/android/accessibilityservice/AccessibilityService.html](#)) service, which is referenced from its [SERVICE\\_META\\_DATA](#) ([/reference/android/accessibilityservice/AccessibilityService.html#SERVICE\\_META\\_DATA](#)) meta-data entry.

Includes the following attributes:

Attribute	Description
<a href="#">android:accessibilityEventTypes</a>	The event types this service would like to receive as specified in <a href="#">AccessibilityEvent</a> .
<a href="#">android:accessibilityFeedbackType</a>	The feedback types this service provides as specified in <a href="#">AccessibilityServiceInfo</a> .
<a href="#">android:accessibilityFlags</a>	Additional flags as specified in <a href="#">AccessibilityServiceInfo</a> .
<a href="#">android:canRetrieveWindowContent</a>	Flag whether the accessibility service wants to be able to retrieve the active window content.
<a href="#">android:description</a>	Short description of the accessibility service purpose or behavior.
<a href="#">android:notificationTimeout</a>	The minimal period in milliseconds between two accessibility events of the same type are sent to this service.
<a href="#">android:packageNames</a>	Comma separated package names from which this service would like to receive events (leave out for all packages).
<a href="#">android:settingsActivity</a>	Component name of an activity that allows the user to modify the settings for this service.

#### See Also

[AccessibilityService.accessibilityEventTypes](#)  
[AccessibilityService.accessibilityFeedbackType](#)  
[AccessibilityService.accessibilityFlags](#)  
[AccessibilityService.canRetrieveWindowContent](#)  
[AccessibilityService.description](#)  
[AccessibilityService.notificationTimeout](#)  
[AccessibilityService.packageNames](#)  
[AccessibilityService.settingsActivity](#)

## public static final int[] AccountAuthenticator

Use account-authenticator as the root tag of the XML resource that describes an account authenticator.

Includes the following attributes:

Attribute	Description
<a href="#">android:accountPreferences</a>	A preferences.
<a href="#">android:accountType</a>	The account type this authenticator handles.
<a href="#">android:customTokens</a>	Account handles its own token storage and permissions.
<a href="#">android:icon</a>	The icon of the authenticator.
<a href="#">android:label</a>	The user-visible name of the authenticator.
<a href="#">android:smallIcon</a>	Smaller icon of the authenticator.

#### See Also

[AccountAuthenticator.accountPreferences](#)  
[AccountAuthenticator.accountType](#)  
[AccountAuthenticator.customTokens](#)  
[AccountAuthenticator.icon](#)  
[AccountAuthenticator.label](#)  
[AccountAuthenticator.smallIcon](#)

## public static final int[] ActionBar

Attributes used to style the Action Bar.

Includes the following attributes:

Attribute	Description
<a href="#">android:background</a>	Specifies a background drawable for the action bar.
<a href="#">android:backgroundSplit</a>	Specifies a background drawable for the bottom component of a split action bar.
<a href="#">android:backgroundStacked</a>	Specifies a background drawable for a second stacked row of the action bar.
<a href="#">android:customNavigationLayout</a>	Specifies a layout for custom navigation.
<a href="#">android:displayOptions</a>	Options affecting how the action bar is displayed.
<a href="#">android:divider</a>	Specifies the drawable used for item dividers.
<a href="#">android:height</a>	Specifies a fixed height.
<a href="#">android:homeLayout</a>	Specifies a layout to use for the "home" section of the action bar.
<a href="#">android:icon</a>	Specifies the drawable used for the application icon.

<a href="#"><u>android:indeterminateProgressStyle</u></a>	Specifies a style resource to use for an indeterminate progress spinner.
<a href="#"><u>android:itemPadding</u></a>	Specifies padding that should be applied to the left and right sides of system-provided items in the bar.
<a href="#"><u>android:logo</u></a>	Specifies the drawable used for the application logo.
<a href="#"><u>android:navigationMode</u></a>	The type of navigation to use.
<a href="#"><u>android:progressBarPadding</u></a>	Specifies the horizontal padding on either end for an embedded progress bar.
<a href="#"><u>android:progressBarStyle</u></a>	Specifies a style resource to use for an embedded progress bar.
<a href="#"><u>android:subtitle</u></a>	Specifies subtitle text used for navigationMode="normal"
<a href="#"><u>android:subtitleTextStyle</u></a>	Specifies a style to use for subtitle text.
<a href="#"><u>android:title</u></a>	Specifies title text used for navigationMode="normal"
<a href="#"><u>android:titleTextStyle</u></a>	Specifies a style to use for title text.

#### See Also

[ActionBar\\_background](#)  
[ActionBar\\_backgroundSplit](#)  
[ActionBar\\_backgroundStacked](#)  
[ActionBar\\_customNavigationLayout](#)  
[ActionBar\\_displayOptions](#)  
[ActionBar\\_divider](#)  
[ActionBar\\_height](#)  
[ActionBar\\_homeLayout](#)  
[ActionBar\\_icon](#)  
[ActionBar\\_ineterminateProgressStyle](#)  
[ActionBar\\_itemPadding](#)  
[ActionBar\\_logo](#)  
[ActionBar\\_navigationMode](#)  
[ActionBar\\_progressBarPadding](#)  
[ActionBar\\_progressBarStyle](#)  
[ActionBar\\_subtitle](#)  
[ActionBar\\_subtitleTextStyle](#)  
[ActionBar\\_title](#)  
[ActionBar\\_titleTextStyle](#)

## public static final int[] **ActionBar\_LayoutParams**

Attributes that can be used with a ActionBar.LayoutParams.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:layout_gravity</u></a>	Standard gravity constant that a child supplies to its parent.

#### See Also

[ActionBar\\_LayoutParams\\_layout\\_gravity](#)

## public static final int[] **ActionMenuItemView**

Attributes that can be used with a ActionMenuItemView.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:minWidth</u></a>	

#### See Also

[ActionMenuItemView\\_minWidth](#)

## public static final int[] **ActionMode**

Attributes that can be used with a ActionMode.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:background</u></a>	Specifies a background for the action mode bar.
<a href="#"><u>android:backgroundSplit</u></a>	Specifies a background for the split action mode bar.
<a href="#"><u>android:height</u></a>	Specifies a fixed height for the action mode bar.
<a href="#"><u>android:subtitleTextStyle</u></a>	Specifies a style to use for subtitle text.

[android:titleTextStyle](#) Specifies a style to use for title text.

**See Also**

[ActionMode\\_background](#)  
[ActionMode\\_backgroundSplit](#)  
[ActionMode\\_height](#)  
[ActionMode\\_subtitleTextStyle](#)  
[ActionMode\\_titleTextStyle](#)

**public static final int[] ActivityChooserView**

Attributes for a ActivityChooserView.

Includes the following attributes:

**Attribute Description**

**public static final int[] AdapterViewAnimator**

Attributes that can be used with a AdapterViewAnimator.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:animateFirstView</u></a>	Defines whether to animate the current View when the ViewAnimation is first displayed.
<a href="#"><u>android:inAnimation</u></a>	Identifier for the animation to use when a view is shown.
<a href="#"><u>android:loopViews</u></a>	Defines whether the animator loops to the first view once it has reached the end of the list.
<a href="#"><u>android:outAnimation</u></a>	Identifier for the animation to use when a view is hidden.

**See Also**

[AdapterViewAnimator\\_animateFirstView](#)  
[AdapterViewAnimator\\_inAnimation](#)  
[AdapterViewAnimator\\_loopViews](#)  
[AdapterViewAnimator\\_outAnimation](#)

**public static final int[] AdapterViewFlipper**

Attributes that can be used with a AdapterViewFlipper.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:autoStart</u></a>	When true, automatically start animating
<a href="#"><u>android:flipInterval</u></a>	

**See Also**

[AdapterViewFlipper\\_autoStart](#)  
[AdapterViewFlipper\\_flipInterval](#)

**public static final int[] AlertDialog**

The set of attributes that describe a AlertDialog's theme.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:bottomBright</u></a>	
<a href="#"><u>android:bottomDark</u></a>	
<a href="#"><u>android:bottomMedium</u></a>	
<a href="#"><u>android:centerBright</u></a>	
<a href="#"><u>android:centerDark</u></a>	
<a href="#"><u>android:centerMedium</u></a>	
<a href="#"><u>android:fullBright</u></a>	
<a href="#"><u>android:fullDark</u></a>	
<a href="#"><u>android:layout</u></a>	Supply an identifier for the layout resource to inflate when the ViewStub becomes visible or when forced to do so.
<a href="#"><u>android:topBright</u></a>	
<a href="#"><u>android:topDark</u></a>	

**See Also**

[AlertDialog\\_bottomBright](#)  
[AlertDialog\\_bottomDark](#)  
[AlertDialog\\_bottomMedium](#)  
[AlertDialog\\_centerBright](#)  
[AlertDialog\\_centerDark](#)  
[AlertDialog\\_centerMedium](#)  
[AlertDialog\\_fullBright](#)  
[AlertDialog\\_fullDark](#)  
[AlertDialog\\_layout](#)  
[AlertDialog\\_topBright](#)  
[AlertDialog\\_topDark](#)

## public static final int[] AlphaAnimation

Attributes that can be used with a AlphaAnimation.

Includes the following attributes:

Attribute	Description
<a href="#">android:fromAlpha</a>	
<a href="#">android:toAlpha</a>	

### See Also

[AlphaAnimation\\_fromAlpha](#)  
[AlphaAnimation\\_toAlpha](#)

## public static final int[] AnalogClock

Attributes that can be used with a AnalogClock.

Includes the following attributes:

Attribute	Description
<a href="#">android:dial</a>	
<a href="#">android:hand_hour</a>	
<a href="#">android:hand_minute</a>	

### See Also

[AnalogClock\\_dial](#)  
[AnalogClock\\_hand\\_hour](#)  
[AnalogClock\\_hand\\_minute](#)

## public static final int[] AndroidManifest

The manifest tag is the root of an `AndroidManifest.xml` file, describing the contents of an Android package (.apk) file. One attribute must always be supplied: package gives a unique name for the package, using a Java-style naming convention to avoid name collisions. For example, applications published by Google could have names of the form `com.google.app.appname`

Inside of the manifest tag, may appear the following tags in any order: [permission](#)  
[/reference/android/R.styleable.html#AndroidManifestPermission](#), [permission-group](#)  
[/reference/android/R.styleable.html#AndroidManifestPermissionGroup](#), [permission-tree](#)  
[/reference/android/R.styleable.html#AndroidManifestPermissionTree](#), [uses-sdk](#)  
[/reference/android/R.styleable.html#AndroidManifestUsesSdk](#), [uses-permission](#)  
[/reference/android/R.styleable.html#AndroidManifestUsesPermission](#), [uses-configuration](#)  
[/reference/android/R.styleable.html#AndroidManifestUsesConfiguration](#), [application](#)  
[/reference/android/R.styleable.html#AndroidManifestApplication](#), [instrumentation](#)  
[/reference/android/R.styleable.html#AndroidManifestInstrumentation](#), [uses-feature](#)  
[/reference/android/R.styleable.html#AndroidManifestUsesFeature](#).

Includes the following attributes:

Attribute	Description
<a href="#">android:installLocation</a>	The default install location defined by an application.
<a href="#">android:sharedUserId</a>	Specify the name of a user ID that will be shared between multiple packages.
<a href="#">android:sharedUserLabel</a>	Specify a label for the shared user UID of this package.
<a href="#">android:versionCode</a>	Internal version code.
<a href="#">android:versionName</a>	The text shown to the user to indicate the version they have.

### See Also

[AndroidManifest\\_installLocation](#)  
[AndroidManifest\\_sharedUserId](#)

[AndroidManifest\\_sharedUserLabel](#)  
[AndroidManifest\\_versionCode](#)  
[AndroidManifest\\_versionName](#)

## public static final int[] **AndroidManifestAction**

Attributes that can be supplied in an AndroidManifest.xml action tag, a child of the [intent-filter](#) ([/reference/android/R.styleable.html#AndroidManifestIntentFilter](#)) tag. See [addAction\(String\)](#) ([/reference/android/content/IntentFilter.html#addAction\(java.lang.String\)](#)) for more information.

Includes the following attributes:

Attribute	Description
<a href="#">android:name</a>	The name of an action that is handled, using the Java-style naming convention.

### See Also

[AndroidManifestAction\\_name](#)

## public static final int[] **AndroidManifestActivity**

The activity tag declares an [Activity](#) ([/reference/android/app/Activity.html](#)) class that is available as part of the package's application components, implementing a part of the application's user interface.

Zero or more [intent-filter](#) ([/reference/android/R.styleable.html#AndroidManifestIntentFilter](#)) tags can be included inside of an activity, to specify the intents that it can handle. If none are specified, the activity can only be started through direct specification of its class name. The activity tag appears as a child tag of the [application](#) ([/reference/android/R.styleable.html#AndroidManifestApplication](#)) tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:allowTaskReparenting</a>	Specify that an activity can be moved out of a task it is in to the task it has an affinity for when appropriate.
<a href="#">android:alwaysRetainTaskState</a>	Specify whether an activity's task state should always be maintained by the system, or if it is allowed to reset the task to its initial state in certain situations.
<a href="#">android:clearTaskOnLaunch</a>	Specify whether an activity's task should be cleared when it is re-launched from the home screen.
<a href="#">android:configChanges</a>	Specify one or more configuration changes that the activity will handle itself.
<a href="#">android:description</a>	Descriptive text for the associated data.
<a href="#">android:enabled</a>	Specify whether the activity is enabled or not (that is, can be instantiated by the system).
<a href="#">android:excludeFromRecents</a>	Indicates that an Activity should be excluded from the list of recently launched activities.
<a href="#">android:exported</a>	Flag indicating whether the given application component is available to other applications.
<a href="#">android:finishOnCloseSystemDialogs</a>	Specify whether an activity should be finished when a "close system windows" request has been made.
<a href="#">android:finishOnTaskLaunch</a>	Specify whether an activity should be finished when its task is brought to the foreground by relaunching from the home screen.
<a href="#">android:hardwareAccelerated</a>	Flag indicating whether the application's rendering should be hardware accelerated if possible.
<a href="#">android:icon</a>	A Drawable resource providing a graphical representation of its associated item.
<a href="#">android:immersive</a>	Flag declaring this activity to be 'immersive'; immersive activities should not be interrupted with other activities or notifications.
<a href="#">android:label</a>	A user-legible name for the given item.
<a href="#">android:launchMode</a>	Specify how an activity should be launched.
<a href="#">android:logo</a>	A Drawable resource providing an extended graphical logo for its associated item.
<a href="#">android:multiprocess</a>	Specify whether a component is allowed to have multiple instances of itself running in different processes.
<a href="#">android:name</a>	Required name of the class implementing the activity, deriving from <a href="#">Activity</a> .
<a href="#">android:noHistory</a>	Specify whether an activity should be kept in its history stack.

<a href="#">android:parentActivityName</a>	The name of the logical parent of the activity as it appears in the manifest.
<a href="#">android:permission</a>	Specify a permission that a client is required to have in order to use the associated object.
<a href="#">android:process</a>	Specify a specific process that the associated code is to run in.
<a href="#">android:screenOrientation</a>	Specify the orientation an activity should be run in.
<a href="#">android:showOnLockScreen</a>	Specify that an Activity should be shown over the lock screen and, in a multiuser environment, across all users' windows
<a href="#">android:singleUser</a>	If set to true, a single instance of this component will run for all users.
<a href="#">android:stateNotNeeded</a>	Indicates that an Activity does not need to have its freeze state (as returned by <a href="#">onSaveInstanceState(Bundle)</a> ) retained in order to be restarted.
<a href="#">android:taskAffinity</a>	Specify a task name that activities have an "affinity" to.
<a href="#">android:theme</a>	The overall theme to use for an activity.
<a href="#">android:uiOptions</a>	Extra options for an activity's UI.
<a href="#">android:windowSoftInputMode</a>	Specify the default soft-input mode for the main window of this activity.

#### See Also

[AndroidManifestActivity.allowTaskReparenting](#)  
[AndroidManifestActivity.alwaysRetainTaskState](#)  
[AndroidManifestActivity.clearTaskOnLaunch](#)  
[AndroidManifestActivity.configChanges](#)  
[AndroidManifestActivity.description](#)  
[AndroidManifestActivity.enabled](#)  
[AndroidManifestActivity.excludeFromRecents](#)  
[AndroidManifestActivity.exported](#)  
[AndroidManifestActivity.finishOnCloseSystemDialogs](#)  
[AndroidManifestActivity.finishOnTaskLaunch](#)  
[AndroidManifestActivity.hardwareAccelerated](#)  
[AndroidManifestActivity.icon](#)  
[AndroidManifestActivity.immersive](#)  
[AndroidManifestActivity.label](#)  
[AndroidManifestActivity.launchMode](#)  
[AndroidManifestActivity.logo](#)  
[AndroidManifestActivity.multiprocess](#)  
[AndroidManifestActivity.name](#)  
[AndroidManifestActivity.noHistory](#)  
[AndroidManifestActivity.parentActivityName](#)  
[AndroidManifestActivity.permission](#)  
[AndroidManifestActivity.process](#)  
[AndroidManifestActivity.screenOrientation](#)  
[AndroidManifestActivity.showOnLockScreen](#)  
[AndroidManifestActivity.singleUser](#)  
[AndroidManifestActivity.stateNotNeeded](#)  
[AndroidManifestActivity.taskAffinity](#)  
[AndroidManifestActivity.theme](#)  
[AndroidManifestActivity.uiOptions](#)  
[AndroidManifestActivity.windowSoftInputMode](#)

#### public static final int[] **AndroidManifestActivityAlias**

The `activity-alias` tag declares a new name for an existing [activity](#) ([/reference/android/R.styleable.html#AndroidManifestActivity](#)) tag.

Zero or more [intent-filter](#) ([/reference/android/R.styleable.html#AndroidManifestIntentFilter](#)) tags can be included inside of an activity-alias, to specify the Intents that it can handle. If none are specified, the activity can only be started through direct specification of its class name. The activity-alias tag appears as a child tag of the [application](#) ([/reference/android/R.styleable.html#AndroidManifestApplication](#)) tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:description</a>	Descriptive text for the associated data.
<a href="#">android:enabled</a>	Specify whether the activity-alias is enabled or not (that is, can be instantiated by the system).
<a href="#">android:exported</a>	Flag indicating whether the given application component is available to other applications.

<a href="#">android:icon</a>	A Drawable resource providing a graphical representation of its associated item.
<a href="#">android:label</a>	A user-legible name for the given item.
<a href="#">android:logo</a>	A Drawable resource providing an extended graphical logo for its associated item.
<a href="#">android:name</a>	Required name of the class implementing the activity, deriving from <a href="#">Activity</a> .
<a href="#">android:parentActivityName</a>	The name of the logical parent of the activity as it appears in the manifest.
<a href="#">android:permission</a>	Specify a permission that a client is required to have in order to use the associated object.
<a href="#">android:targetActivity</a>	The name of the activity this alias should launch.

#### See Also

[AndroidManifestActivityAlias\\_description](#)  
[AndroidManifestActivityAlias\\_enabled](#)  
[AndroidManifestActivityAlias\\_exported](#)  
[AndroidManifestActivityAlias\\_icon](#)  
[AndroidManifestActivityAlias\\_label](#)  
[AndroidManifestActivityAlias\\_logo](#)  
[AndroidManifestActivityAlias\\_name](#)  
[AndroidManifestActivityAlias\\_parentActivityName](#)  
[AndroidManifestActivityAlias\\_permission](#)  
[AndroidManifestActivityAlias\\_targetActivity](#)

### public static final int[] **AndroidManifestApplication**

The application tag describes application-level components contained in the package, as well as general application attributes. Many of the attributes you can supply here (such as theme, label, icon, permission, process, taskAffinity, and allowTaskReparenting) serve as default values for the corresponding attributes of components declared inside of the application.

Inside of this element you specify what the application contains, using the elements [provider](#)  
[\(/reference/android/R.styleable.html#AndroidManifestProvider\)](#), [service](#)  
[\(/reference/android/R.styleable.html#AndroidManifestService\)](#), [receiver](#)  
[\(/reference/android/R.styleable.html#AndroidManifestReceiver\)](#), [activity](#)  
[\(/reference/android/R.styleable.html#AndroidManifestActivity\)](#), [activity-alias](#)  
[\(/reference/android/R.styleable.html#AndroidManifestActivityAlias\)](#), and [uses-library](#)  
[\(/reference/android/R.styleable.html#AndroidManifestUsesLibrary\)](#). The application tag appears as a child of the root [manifest](#) ([\(/reference/android/R.styleable.html#AndroidManifest\)](#)) tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:allowBackup</a>	Whether to allow the application to participate in the backup and restore infrastructure.
<a href="#">android:allowClearUserData</a>	Option to let applications specify that user data can/cannot be cleared.
<a href="#">android:allowTaskReparenting</a>	Specify that an activity can be moved out of a task it is in to the task it has an affinity for when appropriate.
<a href="#">android:backupAgent</a>	The name of the class subclassing BackupAgent to manage backup and restore of the application's data on external storage.
<a href="#">android:debuggable</a>	Flag indicating whether the application can be debugged, even when running on a device that is running in user mode.
<a href="#">android:description</a>	Descriptive text for the associated data.
<a href="#">android:enabled</a>	Specify whether the components in this application are enabled or not (that is, can be instantiated by the system).
<a href="#">android:hardwareAccelerated</a>	Flag indicating whether the application's rendering should be hardware accelerated if possible.
<a href="#">android:hasCode</a>	Indicate whether this application contains code.
<a href="#">android:icon</a>	A Drawable resource providing a graphical representation of its associated item.
<a href="#">android:killAfterRestore</a>	Whether the application in question should be terminated after its settings have been restored during a full-system restore operation.
<a href="#">android:label</a>	A user-legible name for the given item.
<a href="#">android:largeHeap</a>	Request that your application's processes be created with a large Dalvik heap.
<a href="#">android:logo</a>	A Drawable resource providing an extended graphical logo for

	its associated item.
<a href="#">android:manageSpaceActivity</a>	Name of activity to be launched for managing the application's space on the device.
<a href="#">android:name</a>	An optional name of a class implementing the overall <a href="#">Application</a> for this package.
<a href="#">android:permission</a>	Specify a permission that a client is required to have in order to use the associated object.
<a href="#">android:persistent</a>	Flag to control special persistent mode of an application.
<a href="#">android:process</a>	Specify a specific process that the associated code is to run in.
<a href="#">android:restoreAnyVersion</a>	Indicate that the application is prepared to attempt a restore of any backed-up dataset, even if the backup is apparently from a newer version of the application than is currently installed on the device.
<a href="#">android:restoreNeedsApplication</a>	@deprecated This attribute is not used by the Android operating system.
<a href="#">android:supportsRtl</a>	Declare that your application will be able to deal with RTL (right to left) layouts.
<a href="#">android:taskAffinity</a>	Specify a task name that activities have an "affinity" to.
<a href="#">android:testOnly</a>	Option to indicate this application is only for testing purposes.
<a href="#">android:theme</a>	The overall theme to use for an activity.
<a href="#">android:uiOptions</a>	Extra options for an activity's UI.
<a href="#">android:vmSafeMode</a>	Flag indicating whether the application requests the VM to operate in the safe mode.

#### See Also

[AndroidManifestApplication.allowBackup](#)  
[AndroidManifestApplication.allowClearUserData](#)  
[AndroidManifestApplication.allowTaskReparenting](#)  
[AndroidManifestApplication.backupAgent](#)  
[AndroidManifestApplication.debuggable](#)  
[AndroidManifestApplication.description](#)  
[AndroidManifestApplication.enabled](#)  
[AndroidManifestApplication.hardwareAccelerated](#)  
[AndroidManifestApplication.hasCode](#)  
[AndroidManifestApplication.icon](#)  
[AndroidManifestApplication.killAfterRestore](#)  
[AndroidManifestApplication.label](#)  
[AndroidManifestApplication.largeHeap](#)  
[AndroidManifestApplication.logo](#)  
[AndroidManifestApplication.manageSpaceActivity](#)  
[AndroidManifestApplication.name](#)  
[AndroidManifestApplication.permission](#)  
[AndroidManifestApplication.persistent](#)  
[AndroidManifestApplication.process](#)  
[AndroidManifestApplication.restoreAnyVersion](#)  
[AndroidManifestApplication.restoreNeedsApplication](#)  
[AndroidManifestApplication.supportsRtl](#)  
[AndroidManifestApplication.taskAffinity](#)  
[AndroidManifestApplication.testOnly](#)  
[AndroidManifestApplication.theme](#)  
[AndroidManifestApplication.uiOptions](#)  
[AndroidManifestApplication.vmSafeMode](#)

## public static final int[] **AndroidManifestCategory**

Attributes that can be supplied in an `AndroidManifest.xml` category tag, a child of the [intent-filter](#) (`/reference/android/R.styleable.html#AndroidManifestIntentFilter`) tag. See [addCategory\(String\)](#) (`/reference/android/content/IntentFilter.html#addCategory(java.lang.String)`) for more information.

Includes the following attributes:

Attribute	Description
<a href="#">android:name</a>	The name of category that is handled, using the Java-style naming convention.

#### See Also

[AndroidManifestCategory.name](#)

## public static final int[] **AndroidManifestCompatibleScreensScreen**

Attributes that can be supplied in an `AndroidManifest.xml` screen tag, a child of `compatible-screens`,

which is itself a child of the root [manifest](#) tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:screenDensity</a>	Specifies a compatible screen density, as per the device configuration screen density bins.
<a href="#">android:screenSize</a>	Specifies a compatible screen size, as per the device configuration screen size bins.

#### See Also

[AndroidManifestCompatibleScreensScreen\\_screenDensity](#)  
[AndroidManifestCompatibleScreensScreen\\_screenSize](#)

### public static final int[] **AndroidManifestData**

Attributes that can be supplied in an AndroidManifest.xml data tag, a child of the [intent-filter](#) tag, describing the types of data that match. This tag can be specified multiple times to supply multiple data options, as described in the [IntentFilter](#) class. Note that all such tags are adding options to the same IntentFilter so that, for example, <data android:scheme="myscheme" android:host="me.com" /> is equivalent to <data android:scheme="myscheme" /> <data android:host="me.com" />.

Includes the following attributes:

Attribute	Description
<a href="#">android:host</a>	Specify a URI authority host that is handled, as per <a href="#">IntentFilter.addDataAuthority()</a> .
<a href="#">android:mimeType</a>	Specify a MIME type that is handled, as per <a href="#">IntentFilter.addDataType()</a> .
<a href="#">android:path</a>	Specify a URI path that must exactly match, as per <a href="#">IntentFilter.addDataAuthority()</a> with <a href="#">PATTERN_LITERAL</a> .
<a href="#">android:pathPattern</a>	Specify a URI path that matches a simple pattern, as per <a href="#">IntentFilter.addDataAuthority()</a> with <a href="#">PATTERN_SIMPLE_GLOB</a> .
<a href="#">android:pathPrefix</a>	Specify a URI path that must be a prefix to match, as per <a href="#">IntentFilter.addDataAuthority()</a> with <a href="#">PATTERN_PREFIX</a> .
<a href="#">android:port</a>	Specify a URI authority port that is handled, as per <a href="#">IntentFilter.addDataAuthority()</a> .
<a href="#">android:scheme</a>	Specify a URI scheme that is handled, as per <a href="#">IntentFilter.addDataScheme()</a> .

#### See Also

[AndroidManifestData\\_host](#)  
[AndroidManifestData\\_mimeType](#)  
[AndroidManifestData\\_path](#)  
[AndroidManifestData\\_pathPattern](#)  
[AndroidManifestData\\_pathPrefix](#)  
[AndroidManifestData\\_port](#)  
[AndroidManifestData\\_scheme](#)

### public static final int[] **AndroidManifestGrantUriPermission**

Attributes that can be supplied in an AndroidManifest.xml grant-uri-permission tag, a child of the [provider](#) tag, describing a specific URI path that can be granted as a permission. This tag can be specified multiple time to supply multiple paths.

Includes the following attributes:

Attribute	Description
<a href="#">android:path</a>	Specify a URI path that must exactly match, as per <a href="#">PatternMatcher</a> with <a href="#">PATTERN_LITERAL</a> .
<a href="#">android:pathPattern</a>	Specify a URI path that matches a simple pattern, as per <a href="#">PatternMatcher</a> with <a href="#">PATTERN_SIMPLE_GLOB</a> .
<a href="#">android:pathPrefix</a>	Specify a URI path that must be a prefix to match, as per <a href="#">PatternMatcher</a> with <a href="#">PATTERN_PREFIX</a> .

#### See Also

[AndroidManifestGrantUriPermission\\_path](#)  
[AndroidManifestGrantUriPermission\\_pathPattern](#)  
[AndroidManifestGrantUriPermission\\_pathPrefix](#)

## public static final int[] **AndroidManifestInstrumentation**

Attributes that can be supplied in an `AndroidManifest.xml` instrumentation tag, a child of the root [manifest](#) (`/reference/android/R.styleable.html#AndroidManifest`) tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:functionalTest</a>	Flag indicating that an Instrumentation class should be run as a functional test.
<a href="#">android:handleProfiling</a>	Flag indicating that an Instrumentation class wants to take care of starting/stopping profiling itself, rather than relying on the default behavior of profiling the complete time it is running.
<a href="#">android:icon</a>	A Drawable resource providing a graphical representation of its associated item.
<a href="#">android:label</a>	A user-legible name for the given item.
<a href="#">android:logo</a>	A Drawable resource providing an extended graphical logo for its associated item.
<a href="#">android:name</a>	Required name of the class implementing the instrumentation, deriving from <a href="#">Instrumentation</a> .
<a href="#">android:targetPackage</a>	The name of the application package that an Instrumentation object will run against.

### See Also

[AndroidManifestInstrumentation\\_functionalTest](#)  
[AndroidManifestInstrumentation\\_handleProfiling](#)  
[AndroidManifestInstrumentation\\_icon](#)  
[AndroidManifestInstrumentation\\_label](#)  
[AndroidManifestInstrumentation\\_logo](#)  
[AndroidManifestInstrumentation\\_name](#)  
[AndroidManifestInstrumentation\\_targetPackage](#)

## public static final int[] **AndroidManifestIntentFilter**

The intent-filter tag is used to construct an [IntentFilter](#) (`/reference/android/content/IntentFilter.html`) object that will be used to determine which component can handle a particular [Intent](#) (`/reference/android/content/Intent.html`) that has been given to the system. It can be used as a child of the [activity](#) (`/reference/android/R.styleable.html#AndroidManifestActivity`), [receiver](#) (`/reference/android/R.styleable.html#AndroidManifestReceiver`) and [service](#) (`/reference/android/R.styleable.html#AndroidManifestService`) tags.

Zero or more [action](#) (`/reference/android/R.styleable.html#AndroidManifestAction`), [category](#) (`/reference/android/R.styleable.html#AndroidManifestCategory`), and/or [data](#) (`/reference/android/R.styleable.html#AndroidManifestData`) tags should be included inside to describe the contents of the filter.

The optional label and icon attributes here are used with an activity to supply an alternative description of that activity when it is being started through an Intent matching this filter.

Includes the following attributes:

Attribute	Description
<a href="#">android:icon</a>	A Drawable resource providing a graphical representation of its associated item.
<a href="#">android:label</a>	A user-legible name for the given item.
<a href="#">android:logo</a>	A Drawable resource providing an extended graphical logo for its associated item.
<a href="#">android:priority</a>	Specify the relative importance or ability in handling a particular Intent.

### See Also

[AndroidManifestIntentFilter\\_icon](#)  
[AndroidManifestIntentFilter\\_label](#)  
[AndroidManifestIntentFilter\\_logo](#)  
[AndroidManifestIntentFilter\\_priority](#)

## public static final int[] **AndroidManifestMetaData**

The meta-data tag is used to attach additional arbitrary data to an application component. The data can later be retrieved programmatically from the [ComponentInfo.metaData](#) (`/reference/android/content/pm/PackageManager.html#metaData`) field. There is no meaning given to this data by the system. You may supply the data through either the value or resource attribute; if both are given, then resource will be used.

It is highly recommended that you avoid supplying related data as multiple separate meta-data entries.

Instead, if you have complex data to associate with a component, then use the `resource` attribute to assign an XML resource that the client can parse to retrieve the complete data.

Includes the following attributes:

Attribute	Description
<code>android:name</code>	A unique name for the given item.
<code>android:resource</code>	Resource identifier to assign to this piece of named meta-data.
<code>android:value</code>	Concrete value to assign to this piece of named meta-data.

**See Also**

[AndroidManifestMetaData\\_name](#)  
[AndroidManifestMetaData\\_resource](#)  
[AndroidManifestMetaData\\_value](#)

## public static final int[] **AndroidManifestOriginalPackage**

Private tag to declare the original package name that this package is based on. Only used for packages installed in the system image. If given, and different than the actual package name, and the given original package was previously installed on the device but the new one was not, then the data for the old one will be renamed to be for the new package.

This appears as a child tag of the root [manifest](#)  
[\(/reference/android/R.styleable.html#AndroidManifest\)](#) tag.

Includes the following attributes:

Attribute	Description
<code>android:name</code>	A unique name for the given item.

**See Also**

[AndroidManifestOriginalPackage\\_name](#)

## public static final int[] **AndroidManifestPackageVerifier**

Attributes relating to a package verifier.

Includes the following attributes:

Attribute	Description
<code>android:name</code>	Specifies the Java-style package name that defines this package verifier.
<code>android:publicKey</code>	The Base64 encoded public key of the package verifier's signature.

**See Also**

[AndroidManifestPackageVerifier\\_name](#)  
[AndroidManifestPackageVerifier\\_publicKey](#)

## public static final int[] **AndroidManifestPathPermission**

Attributes that can be supplied in an `AndroidManifest.xml` path-permission tag, a child of the [provider](#)  
[\(/reference/android/R.styleable.html#AndroidManifestProvider\)](#) tag, describing a permission that allows access to a specific path in the provider. This tag can be specified multiple time to supply multiple paths.

Includes the following attributes:

Attribute	Description
<code>android:path</code>	Specify a URI path that must exactly match, as per <a href="#">PatternMatcher</a> with <a href="#">PATTERN_LITERAL</a> .
<code>android:pathPattern</code>	Specify a URI path that matches a simple pattern, as per <a href="#">PatternMatcher</a> with <a href="#">PATTERN_SIMPLE_GLOB</a> .
<code>android:pathPrefix</code>	Specify a URI path that must be a prefix to match, as per <a href="#">PatternMatcher</a> with <a href="#">PATTERN_PREFIX</a> .
<code>android:permission</code>	Specify a permission that a client is required to have in order to use the associated object.
<code>android:readPermission</code>	A specific <a href="#">permission</a> name for read-only access to a <a href="#">ContentProvider</a> .
<code>android:writePermission</code>	A specific <a href="#">permission</a> name for write access to a <a href="#">ContentProvider</a> .

**See Also**

[AndroidManifestPathPermission\\_path](#)  
[AndroidManifestPathPermission\\_pathPattern](#)  
[AndroidManifestPathPermission\\_pathPrefix](#)

```
AndroidManifestPermission permission  
AndroidManifestPermission readPermission  
AndroidManifestPermission writePermission
```

## public static final int[] **AndroidManifestPermission**

The `permission` tag declares a security permission that can be used to control access from other packages to specific components or features in your package (or other packages). See the [Security and Permissions](#) ([/guide/topics/security/security.html](#)) document for more information on permissions.

This appears as a child tag of the root [manifest](#) ([/reference/android/R.styleable.html#AndroidManifest](#)) tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:description</a>	Descriptive text for the associated data.
<a href="#">android:icon</a>	A Drawable resource providing a graphical representation of its associated item.
<a href="#">android:label</a>	A user-legible name for the given item.
<a href="#">android:logo</a>	A Drawable resource providing an extended graphical logo for its associated item.
<a href="#">android:name</a>	Required public name of the permission, which other components and packages will use when referring to this permission.
<a href="#">android:permissionFlags</a>	Flags indicating more context for a permission.
<a href="#">android:permissionGroup</a>	Specified the name of a group that this permission is associated with. Characterizes the potential risk implied in a permission and indicates the <a href="#">android:protectionLevel</a> procedure the system should follow when determining whether to grant the permission to an application requesting it.

### See Also

[AndroidManifestPermission\\_description](#)  
[AndroidManifestPermission\\_icon](#)  
[AndroidManifestPermission\\_label](#)  
[AndroidManifestPermission\\_logo](#)  
[AndroidManifestPermission\\_name](#)  
[AndroidManifestPermission\\_permissionFlags](#)  
[AndroidManifestPermission\\_permissionGroup](#)  
[AndroidManifestPermission\\_protectionLevel](#)

## public static final int[] **AndroidManifestPermissionGroup**

The `permission-group` tag declares a logical grouping of related permissions.

Note that this tag does not declare a permission itself, only a namespace in which further permissions can be placed. See the [<permission>](#) ([/reference/android/R.styleable.html#AndroidManifestPermission](#)) tag for more information.

This appears as a child tag of the root [manifest](#)

Android APIs      API level: 17      [m1#AndroidManifest](#) tag.

Includes the following attributes:	
<a href="#">android</a>	
<a href="#">android.accessibilityservice</a>	
<a href="#">android.accounts</a>	
<a href="#">android.animation</a>	
<a href="#">android.app</a>	
<a href="#">android.app.admin</a>	
<a href="#">android.app.backup</a>	
<a href="#">android.appwidget</a>	
<a href="#">android.content</a>	
<a href="#">android.content.pm</a>	
<a href="#">android.content.res</a>	
<a href="#">android.permissionGroupFlags</a>	Flags indicating more context for a permission group.
<a href="#">android.database.sqlite</a>	Specify the relative importance or ability in handling a particular Intent.

### Manifest

[See Also](#)  
[Manifest.permission](#)  
[Manifest.permissionGroup](#), [description](#)  
R [AndroidManifestPermissionGroup.icon](#)  
R [AndroidManifestPermissionGroup.label](#)  
R [AndroidManifestPermissionGroup.logo](#)  
R [Array](#) [AndroidManifestPermissionGroup.name](#)  
R [attr](#) [AndroidManifestPermissionGroup.permissionGroupFlags](#)

```

R.boolean<AndroidManifestPermissionGroup> priority
R.color
R.dimen
R.integer static final int[] AndroidManifestPermissionTree
R.drawable
R.fraction A permission-tree tag declares the base of a tree of permission values: it declares that this package
R.id has ownership of the given permission name, as well as all names underneath it (separated by '.'). This
R.integer allows you to use the PackageManager.addPermission\(\)
R.interpolator
R.layout
R.menu
R.mipmap
R.plurals
R.raw
R.string
    □ This appears as a child tag of the root manifest
    Use Tree Navigation /reference/android/R.styleable.html#AndroidManifestPermissionTree

```

Includes the following attributes:

Attribute	Description
<a href="#">android:icon</a>	A Drawable resource providing a graphical representation of its associated item.
<a href="#">android:label</a>	A user-legible name for the given item.
<a href="#">android:logo</a>	A Drawable resource providing an extended graphical logo for its associated item.
<a href="#">android:name</a>	Required public name of the permission tree, which is the base name of all permissions under it.

#### See Also

[AndroidManifestPermissionTree\\_icon](#)  
[AndroidManifestPermissionTree\\_label](#)  
[AndroidManifestPermissionTree\\_logo](#)  
[AndroidManifestPermissionTree\\_name](#)

## public static final int[] **AndroidManifestProtectedBroadcast**

Private tag to declare system protected broadcast actions.

This appears as a child tag of the root [manifest](#)  
[/reference/android/R.styleable.html#AndroidManifest](#) tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:name</a>	A unique name for the given item.

#### See Also

[AndroidManifestProtectedBroadcast\\_name](#)

## public static final int[] **AndroidManifestProvider**

The provider tag declares a [ContentProvider](#) ([/reference/android/content/ContentProvider.html](#)) class that is available as part of the package's application components, supplying structured access to data managed by the application.

This appears as a child tag of the [application](#)  
[/reference/android/R.styleable.html#AndroidManifestApplication](#) tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:authorities</a>	Specify the authorities under which this content provider can be found.
<a href="#">android:description</a>	Descriptive text for the associated data.
<a href="#">android:enabled</a>	Specify whether this provider is enabled or not (that is, can be instantiated by the system).
<a href="#">android:exported</a>	Flag indicating whether the given application component is available to other applications.
<a href="#">android:grantUriPermissions</a>	If true, the <a href="#">Context.grantUriPermission</a> or corresponding Intent flags can be used to allow others to access specific URLs in the content provider, even if they do not have an explicit read or write permission.
<a href="#">android:icon</a>	A Drawable resource providing a graphical representation of its associated item.
<a href="#">android:initOrder</a>	Specify the order in which content providers hosted by a process are instantiated when that process is created.

<a href="#">android:label</a>	A user-legible name for the given item.
<a href="#">android:logo</a>	A Drawable resource providing an extended graphical logo for its associated item.
<a href="#">android:multiprocess</a>	Specify whether a component is allowed to have multiple instances of itself running in different processes.
<a href="#">android:name</a>	Required name of the class implementing the provider, deriving from <a href="#">ContentProvider</a> .
<a href="#">android:permission</a>	Specify a permission that a client is required to have in order to use the associated object.
<a href="#">android:process</a>	Specify a specific process that the associated code is to run in.
<a href="#">android:readPermission</a>	A specific <a href="#">permission</a> name for read-only access to a <a href="#">ContentProvider</a> .
<a href="#">android:singleUser</a>	If set to true, a single instance of this component will run for all users.
<a href="#">android:syncable</a>	Flag indicating whether this content provider would like to participate in data synchronization.
<a href="#">android:writePermission</a>	A specific <a href="#">permission</a> name for write access to a <a href="#">ContentProvider</a> .

## See Also

[AndroidManifestProviderAuthorities](#)  
[AndroidManifestProviderDescription](#)  
[AndroidManifestProviderEnabled](#)  
[AndroidManifestProviderExported](#)  
[AndroidManifestProviderGrantUriPermissions](#)  
[AndroidManifestProviderIcon](#)  
[AndroidManifestProviderInitOrder](#)  
[AndroidManifestProviderLabel](#)  
[AndroidManifestProviderLogo](#)  
[AndroidManifestProviderMultiprocess](#)  
[AndroidManifestProviderName](#)  
[AndroidManifestProviderPermission](#)  
[AndroidManifestProviderProcess](#)  
[AndroidManifestProviderReadPermission](#)  
[AndroidManifestProviderSingleUser](#)  
[AndroidManifestProviderSyncable](#)  
[AndroidManifestProviderWritePermission](#)

## public static final int[] **AndroidManifestReceiver**

The receiver tag declares an [BroadcastReceiver](#) ([/reference/android/content/BroadcastReceiver.html](#)) class that is available as part of the package's application components, allowing the application to receive actions or data broadcast by other applications even if it is not currently running.

Zero or more [intent-filter](#) ([/reference/android/R.styleable.html#AndroidManifestIntentFilter](#)) tags can be included inside of a receiver, to specify the Intents it will receive. If none are specified, the receiver will only be run when an Intent is broadcast that is directed at its specific class name. The receiver tag appears as a child tag of the [application](#) ([/reference/android/R.styleable.html#AndroidManifestApplication](#)) tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:description</a>	Descriptive text for the associated data.
<a href="#">android:enabled</a>	Specify whether the receiver is enabled or not (that is, can be instantiated by the system).
<a href="#">android:exported</a>	Flag indicating whether the given application component is available to other applications.
<a href="#">android:icon</a>	A Drawable resource providing a graphical representation of its associated item.
<a href="#">android:label</a>	A user-legible name for the given item.
<a href="#">android:logo</a>	A Drawable resource providing an extended graphical logo for its associated item.
<a href="#">android:name</a>	Required name of the class implementing the receiver, deriving from <a href="#">BroadcastReceiver</a> .
<a href="#">android:permission</a>	Specify a permission that a client is required to have in order to use the associated object.
<a href="#">android:process</a>	Specify a specific process that the associated code is to run in.
<a href="#">android:singleUser</a>	If set to true, a single instance of this component will run for all users.

## See Also

[AndroidManifestReceiver\\_description](#)  
[AndroidManifestReceiver\\_enabled](#)  
[AndroidManifestReceiver\\_exported](#)  
[AndroidManifestReceiver\\_icon](#)  
[AndroidManifestReceiver\\_label](#)  
[AndroidManifestReceiver\\_logo](#)  
[AndroidManifestReceiver\\_name](#)  
[AndroidManifestReceiver\\_permission](#)  
[AndroidManifestReceiver\\_process](#)  
[AndroidManifestReceiver\\_singleUser](#)

## public static final int[] AndroidManifestService

The service tag declares a [Service](#) ([/reference/android/app/Service.html](#)) class that is available as part of the package's application components, implementing long-running background operations or a rich communication API that can be called by other packages.

Zero or more [intent-filter](#) ([/reference/android/R.styleable.html#AndroidManifestIntentFilter](#)) tags can be included inside of a service, to specify the Intents that can connect with it. If none are specified, the service can only be accessed by direct specification of its class name. The service tag appears as a child tag of the [application](#) ([/reference/android/R.styleable.html#AndroidManifestApplication](#)) tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:description</a>	Descriptive text for the associated data.
<a href="#">android:enabled</a>	Specify whether the service is enabled or not (that is, can be instantiated by the system).
<a href="#">android:exported</a>	Flag indicating whether the given application component is available to other applications.
<a href="#">android:icon</a>	A Drawable resource providing a graphical representation of its associated item.
<a href="#">android:isolatedProcess</a>	If set to true, this service will run under a special process that is isolated from the rest of the system.
<a href="#">android:label</a>	A user-legible name for the given item.
<a href="#">android:logo</a>	A Drawable resource providing an extended graphical logo for its associated item.
<a href="#">android:name</a>	Required name of the class implementing the service, deriving from <a href="#">Service</a> .
<a href="#">android:permission</a>	Specify a permission that a client is required to have in order to use the associated object.
<a href="#">android:process</a>	Specify a specific process that the associated code is to run in.
<a href="#">android:singleUser</a>	If set to true, a single instance of this component will run for all users.
<a href="#">android:stopWithTask</a>	If set to true, this service will be automatically stopped when the user remove a task rooted in an activity owned by the application.

## See Also

[AndroidManifestService\\_description](#)  
[AndroidManifestService\\_enabled](#)  
[AndroidManifestService\\_exported](#)  
[AndroidManifestService\\_icon](#)  
[AndroidManifestService\\_isolatedProcess](#)  
[AndroidManifestService\\_label](#)  
[AndroidManifestService\\_logo](#)  
[AndroidManifestService\\_name](#)  
[AndroidManifestService\\_permission](#)  
[AndroidManifestService\\_process](#)  
[AndroidManifestService\\_singleUser](#)  
[AndroidManifestService\\_stopWithTask](#)

## public static final int[] AndroidManifestSupportsScreens

The supports-screens specifies the screen dimensions an application supports. By default a modern application supports all screen sizes and must explicitly disable certain screen sizes here; older applications are assumed to only support the traditional normal (HVGA) screen size. Note that screen size is a separate axis from density, and is determined as the available pixels to an application after density scaling has been applied.

This appears as a child tag of the [manifest](#) ([/reference/android/R.styleable.html#AndroidManifest](#)) tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:anyDensity</a>	Indicates whether the application can accommodate any screen density.
<a href="#">android:compatibleWidthLimitDp</a>	Starting with <a href="#">HONEYCOMB_MR2</a> , this is the new way to specify the largest screens an application is compatible with.
<a href="#">android:largeScreens</a>	Indicates whether the application supports larger screen form-factors.
<a href="#">android:largestWidthLimitDp</a>	Starting with <a href="#">HONEYCOMB_MR2</a> , this is the new way to specify the screens an application is compatible with.
<a href="#">android:normalScreens</a>	Indicates whether an application supports the normal screen form-factors.
<a href="#">android:requiresSmallestWidthDp</a>	Starting with <a href="#">HONEYCOMB_MR2</a> , this is the new way to specify the minimum screen size an application is compatible with.
<a href="#">android:resizeable</a>	Indicates whether the application can resize itself to newer screen sizes.
<a href="#">android:smallScreens</a>	Indicates whether the application supports smaller screen form-factors.
<a href="#">android:xlargeScreens</a>	Indicates whether the application supports extra large screen form-factors.

**See Also**

[AndroidManifestSupportsScreens\\_anyDensity](#)  
[AndroidManifestSupportsScreens\\_compatibleWidthLimitDp](#)  
[AndroidManifestSupportsScreens\\_largeScreens](#)  
[AndroidManifestSupportsScreens\\_largestWidthLimitDp](#)  
[AndroidManifestSupportsScreens\\_normalScreens](#)  
[AndroidManifestSupportsScreens\\_requiresSmallestWidthDp](#)  
[AndroidManifestSupportsScreens\\_resizeable](#)  
[AndroidManifestSupportsScreens\\_smallScreens](#)  
[AndroidManifestSupportsScreens\\_xlargeScreens](#)

## public static final int[] **AndroidManifestUsesConfiguration**

The uses-configuration tag specifies a specific hardware configuration value used by the application. For example an application might specify that it requires a physical keyboard or a particular navigation method like trackball. Multiple such attribute values can be specified by the application.

This appears as a child tag of the root [manifest](#)  
[/reference/android/R.styleable.html#AndroidManifest](#) tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:reqFiveWayNav</a>	Application's requirement for five way navigation
<a href="#">android:reqHardKeyboard</a>	Application's requirement for a hard keyboard
<a href="#">android:reqKeyboardType</a>	The input method preferred by an application.
<a href="#">android:reqNavigation</a>	The navigation device preferred by an application.
<a href="#">android:reqTouchScreen</a>	The type of touch screen used by an application.

**See Also**

[AndroidManifestUsesConfiguration\\_reqFiveWayNav](#)  
[AndroidManifestUsesConfiguration\\_reqHardKeyboard](#)  
[AndroidManifestUsesConfiguration\\_reqKeyboardType](#)  
[AndroidManifestUsesConfiguration\\_reqNavigation](#)  
[AndroidManifestUsesConfiguration\\_reqTouchScreen](#)

## public static final int[] **AndroidManifestUsesFeature**

The uses-feature tag specifies a specific feature used by the application. For example an application might specify that it requires specific version of OpenGL. Multiple such attribute values can be specified by the application.

This appears as a child tag of the root [manifest](#)  
[/reference/android/R.styleable.html#AndroidManifest](#) tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:glesVersion</a>	The GLES driver version number needed by an application.
<a href="#">android:name</a>	The name of the feature that is being used.

[android:required](#) Specify whether this feature is required for the application.

**See Also**

[AndroidManifestUsesFeature\\_glesVersion](#)  
[AndroidManifestUsesFeature\\_name](#)  
[AndroidManifestUsesFeature\\_required](#)

## public static final int[] **AndroidManifestUsesLibrary**

The uses-libraries specifies a shared library that this package requires to be linked against. Specifying this flag tells the system to include this library's code in your class loader.

This appears as a child tag of the [application](#)

[\(/reference/android/R.styleable.html#AndroidManifestApplication\)](#) tag.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:name</u></a>	Required name of the library you use.
<a href="#"><u>android:required</u></a>	Specify whether this library is required for the application.

**See Also**

[AndroidManifestUsesLibrary\\_name](#)  
[AndroidManifestUsesLibrary\\_required](#)

## public static final int[] **AndroidManifestUsesPermission**

The uses-permission tag requests a [<permission>](#)  
[\(/reference/android/R.styleable.html#AndroidManifestPermission\)](#) that the containing package must be granted in order for it to operate correctly. See the [Security and Permissions](#)  
[\(/guide/topics/security/security.html\)](#) document for more information on permissions. Also available is a [list of permissions](#) [\(/reference/android/Manifest.permission.html\)](#) included with the base platform.

This appears as a child tag of the root [manifest](#)  
[\(/reference/android/R.styleable.html#AndroidManifest\)](#) tag. Specify whether this permission is required for the application. The default is true, meaning the application requires the permission, and it must always be granted when it is installed. If you set this to false, then in some cases the application may be installed with it being granted the permission, and it will need to request the permission later if it needs it.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:name</u></a>	Required name of the permission you use, as published with the corresponding name attribute of a <a href="#"><u>&lt;permission&gt;</u></a> tag; often this is one of the <a href="#"><u>standard system permissions</u></a> .

**See Also**

[AndroidManifestUsesPermission\\_name](#)

## public static final int[] **AndroidManifestUsesSdk**

The uses-sdk tag describes the SDK features that the containing package must be running on to operate correctly.

This appears as a child tag of the root [manifest](#)  
[\(/reference/android/R.styleable.html#AndroidManifest\)](#) tag.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:maxSdkVersion</u></a>	This is the maximum SDK version number that an application works on.
<a href="#"><u>android:minSdkVersion</u></a>	This is the minimum SDK version number that the application requires.
<a href="#"><u>android:targetSdkVersion</u></a>	This is the SDK version number that the application is targeting.

**See Also**

[AndroidManifestUsesSdk\\_maxSdkVersion](#)  
[AndroidManifestUsesSdk\\_minSdkVersion](#)  
[AndroidManifestUsesSdk\\_targetSdkVersion](#)

## public static final int[] **AnimatedRotateDrawable**

Attributes that can be used with a AnimatedRotateDrawable.

Includes the following attributes:

Attribute	Description
<a href="#">android:drawable</a>	Reference to a drawable resource to use for the frame.
<a href="#">android:pivotX</a>	
<a href="#">android:pivotY</a>	
<a href="#">android:visible</a>	Provides initial visibility state of the drawable; the default value is false.

#### See Also

[AnimatedRotateDrawable drawable](#)  
[AnimatedRotateDrawable pivotX](#)  
[AnimatedRotateDrawable pivotY](#)  
[AnimatedRotateDrawable visible](#)

## public static final int[] Animation

Attributes that can be used with a Animation.

Includes the following attributes:

Attribute	Description
<a href="#">android:background</a>	Special background behind animation.
<a href="#">android:detachWallpaper</a>	Special option for window animations: if this window is on top of a wallpaper, don't animate the wallpaper with it.
<a href="#">android:duration</a>	Amount of time (in milliseconds) for the animation to run.
<a href="#">android:fillAfter</a>	When set to true, the animation transformation is applied after the animation is over.
<a href="#">android:fillBefore</a>	When set to true or when fillEnabled is not set to true, the animation transformation is applied before the animation has started.
<a href="#">android:fillEnabled</a>	When set to true, the value of fillBefore is taken into account.
<a href="#">android:interpolator</a>	Defines the interpolator used to smooth the animation movement in time.
<a href="#">android:repeatCount</a>	Defines how many times the animation should repeat.
<a href="#">android:repeatMode</a>	Defines the animation behavior when it reaches the end and the repeat count is greater than 0 or infinite.
<a href="#">android:startOffset</a>	Delay in milliseconds before the animation runs, once start time is reached.
<a href="#">android:zAdjustment</a>	Allows for an adjustment of the Z ordering of the content being animated for the duration of the animation.

#### See Also

[Animation\\_background](#)  
[Animation\\_detachWallpaper](#)  
[Animation\\_duration](#)  
[Animation\\_fillAfter](#)  
[Animation\\_fillBefore](#)  
[Animation\\_fillEnabled](#)  
[Animation\\_interpolator](#)  
[Animation\\_repeatCount](#)  
[Animation\\_repeatMode](#)  
[Animation\\_startOffset](#)  
[Animation\\_zAdjustment](#)

## public static final int[] AnimationDrawable

Drawable used to render several animated frames.

Includes the following attributes:

Attribute	Description
<a href="#">android:oneshot</a>	If true, the animation will only run a single time and then stop.
<a href="#">android:variablePadding</a>	If true, allows the drawable's padding to change based on the current state that is selected.
<a href="#">android:visible</a>	Provides initial visibility state of the drawable; the default value is false.

#### See Also

[AnimationDrawable\\_oneshot](#)  
[AnimationDrawable\\_variablePadding](#)  
[AnimationDrawable\\_visible](#)

## public static final int[] AnimationDrawableItem

Represents a single frame inside an AnimationDrawable.

Includes the following attributes:

Attribute	Description
<a href="#">android:drawable</a>	Reference to a drawable resource to use for the frame.
<a href="#">android:duration</a>	Amount of time (in milliseconds) to display this frame.

**See Also**

[AnimationDrawableItem.drawable](#)  
[AnimationDrawableItem.duration](#)

## public static final int[] AnimationSet

Attributes that can be used with a AnimationSet.

Includes the following attributes:

Attribute	Description
<a href="#">android:duration</a>	Amount of time (in milliseconds) to display this frame.
<a href="#">android:fillAfter</a>	When set to true, the animation transformation is applied after the animation is over.
<a href="#">android:fillBefore</a>	When set to true or when fillEnabled is not set to true, the animation transformation is applied before the animation has started.
<a href="#">android:repeatMode</a>	Defines the animation behavior when it reaches the end and the repeat count is greater than 0 or infinite.
<a href="#">android:shareInterpolator</a>	
<a href="#">android:startOffset</a>	Delay in milliseconds before the animation runs, once start time is reached.

**See Also**

[AnimationSet.duration](#)  
[AnimationSet.fillAfter](#)  
[AnimationSet.fillBefore](#)  
[AnimationSet.repeatMode](#)  
[AnimationSet.shareInterpolator](#)  
[AnimationSet.startOffset](#)

## public static final int[] Animator

Attributes that can be used with a Animator.

Includes the following attributes:

Attribute	Description
<a href="#">android:duration</a>	Amount of time (in milliseconds) for the animation to run.
<a href="#">android:interpolator</a>	Defines the interpolator used to smooth the animation movement in time.
<a href="#">android:repeatCount</a>	Defines how many times the animation should repeat.
<a href="#">android:repeatMode</a>	Defines the animation behavior when it reaches the end and the repeat count is greater than 0 or infinite.
<a href="#">android:startOffset</a>	Delay in milliseconds before the animation runs, once start time is reached.
<a href="#">android:valueFrom</a>	Value the animation starts from.
<a href="#">android:valueTo</a>	Value the animation animates to.
<a href="#">android:valueType</a>	The type of valueFrom and valueTo.

**See Also**

[Animator.duration](#)  
[Animator.interpolator](#)  
[Animator.repeatCount](#)  
[Animator.repeatMode](#)  
[Animator.startOffset](#)  
[Animator.valueFrom](#)  
[Animator.valueTo](#)  
[Animator.valueType](#)

## public static final int[] AnimatorSet

Attributes that can be used with a AnimatorSet.

Includes the following attributes:

Attribute	Description
<a href="#">android:ordering</a>	Name of the property being animated.

**See Also**

[AnimatorSet\\_ordering](#)

**public static final int[] AnticipateInterpolator**

Attributes that can be used with a AnticipateInterpolator.

Includes the following attributes:

Attribute	Description
<a href="#">android:tension</a>	This is the amount of tension.

**See Also**

[AnticipateInterpolator\\_tension](#)

**public static final int[] AnticipateOvershootInterpolator**

Attributes that can be used with a AnticipateOvershootInterpolator.

Includes the following attributes:

Attribute	Description
<a href="#">android:extraTension</a>	This is the amount by which to multiply the tension.
<a href="#">android:tension</a>	This is the amount of tension.

**See Also**

[AnticipateOvershootInterpolator\\_extraTension](#)

[AnticipateOvershootInterpolator\\_tension](#)

**public static final int[] AppWidgetProviderInfo**

Use appwidget-provider as the root tag of the XML resource that describes an AppWidget provider. See [android.appwidget \(/reference/android/appwidget/package-summary.html\)](#) package for more info.

Includes the following attributes:

Attribute	Description
<a href="#">android:autoAdvanceViewId</a>	The view id of the AppWidget subview which should be auto-advanced.
<a href="#">android:configure</a>	A class name in the AppWidget's package to be launched to configure.
<a href="#">android:initialKeyguardLayout</a>	A resource id of a layout.
<a href="#">android:initialLayout</a>	A resource id of a layout.
<a href="#">android:minHeight</a>	Minimum height of the AppWidget.
<a href="#">android:minResizeHeight</a>	Minimum height that the AppWidget can be resized to.
<a href="#">android:minResizeWidth</a>	Minimum width that the AppWidget can be resized to.
<a href="#">android:minWidth</a>	Minimum width of the AppWidget.
<a href="#">android:previewImage</a>	A preview of what the AppWidget will look like after it's configured.
<a href="#">android:resizeMode</a>	Optional parameter which indicates if and how this widget can be resized.
<a href="#">android:updatePeriodMillis</a>	Update period in milliseconds, or 0 if the AppWidget will update itself.
<a href="#">android:widgetCategory</a>	Optional parameter which indicates where this widget can be shown, ie.

**See Also**

[AppWidgetProviderInfo\\_autoAdvanceViewId](#)  
[AppWidgetProviderInfo\\_configure](#)  
[AppWidgetProviderInfo\\_initialKeyguardLayout](#)  
[AppWidgetProviderInfo\\_initialLayout](#)  
[AppWidgetProviderInfo\\_minHeight](#)  
[AppWidgetProviderInfo\\_minResizeHeight](#)  
[AppWidgetProviderInfo\\_minResizeWidth](#)  
[AppWidgetProviderInfo\\_minWidth](#)  
[AppWidgetProviderInfo\\_previewImage](#)  
[AppWidgetProviderInfo\\_resizeMode](#)  
[AppWidgetProviderInfo\\_updatePeriodMillis](#)  
[AppWidgetProviderInfo\\_widgetCategory](#)

**public static final int[] AutoCompleteTextView**

Attributes that can be used with a AutoCompleteTextView.

Includes the following attributes:

Attribute	Description
<a href="#">android:completionHint</a>	Defines the hint displayed in the drop down menu.
<a href="#">android:completionHintView</a>	Defines the hint view displayed in the drop down menu.
<a href="#">android:completionThreshold</a>	Defines the number of characters that the user must type before completion suggestions are displayed in a drop down menu.
<a href="#">android:dropDownAnchor</a>	View to anchor the auto-complete dropdown to.
<a href="#">android:dropDownHeight</a>	Specifies the basic height of the dropdown.
<a href="#">android:dropDownHorizontalOffset</a>	Amount of pixels by which the drop down should be offset horizontally.
<a href="#">android:dropDownSelector</a>	Selector in a drop down list.
<a href="#">android:dropDownVerticalOffset</a>	Amount of pixels by which the drop down should be offset vertically.
<a href="#">android:dropDownWidth</a>	Specifies the basic width of the dropdown.
<a href="#">android:inputType</a>	The type of data being placed in a text field, used to help an input method decide how to let the user enter text.

#### See Also

[AutoCompleteTextView\\_completionHint](#)  
[AutoCompleteTextView\\_completionHintView](#)  
[AutoCompleteTextView\\_completionThreshold](#)  
[AutoCompleteTextView\\_dropDownAnchor](#)  
[AutoCompleteTextView\\_dropDownHeight](#)  
[AutoCompleteTextView\\_dropDownHorizontalOffset](#)  
[AutoCompleteTextView\\_dropDownSelector](#)  
[AutoCompleteTextView\\_dropDownVerticalOffset](#)  
[AutoCompleteTextView\\_dropDownWidth](#)  
[AutoCompleteTextView\\_inputType](#)

## public static final int[] BitmapDrawable

Drawable used to draw bitmaps.

Includes the following attributes:

Attribute	Description
<a href="#">android:antialias</a>	Enables or disables antialiasing.
<a href="#">android:dither</a>	Enables or disables dithering of the bitmap if the bitmap does not have the same pixel configuration as the screen (for instance: a ARGB 8888 bitmap with an RGB 565 screen).
<a href="#">android:filter</a>	Enables or disables bitmap filtering.
<a href="#">android:gravity</a>	Defines the gravity for the bitmap.
<a href="#">android:src</a>	Identifier of the bitmap file.
<a href="#">android:titleMode</a>	Defines the tile mode.

#### See Also

[BitmapDrawable\\_antialias](#)  
[BitmapDrawable\\_dither](#)  
[BitmapDrawable\\_filter](#)  
[BitmapDrawable\\_gravity](#)  
[BitmapDrawable\\_src](#)  
[BitmapDrawable\\_tileMode](#)

## public static final int[] Button

Attributes that can be used with a Button.

## public static final int[] CalendarView

Attributes that can be used with a CalendarView.

Includes the following attributes:

Attribute	Description
<a href="#">android:dateTextAppearance</a>	The text appearance for the calendar dates.
<a href="#">android:firstDayOfWeek</a>	The first day of week according to <a href="#">Calendar</a> .
<a href="#">android:focusedMonthDateColor</a>	The color for the dates of the focused month.

<a href="#">android:maxDate</a>	The minimal date shown by this calendar view in mm/dd/yyyy format.
<a href="#">android:minDate</a>	The minimal date shown by this calendar view in mm/dd/yyyy format.
<a href="#">android:selectedDateVerticalBar</a>	Drawable for the vertical bar shown at the beginning and at the end of the selected date.
<a href="#">android:selectedWeekBackgroundColor</a>	The background color for the selected week.
<a href="#">android:showWeekNumber</a>	Whether do show week numbers.
<a href="#">android:shownWeekCount</a>	The number of weeks to be shown.
<a href="#">android:unfocusedMonthTextColor</a>	The color for the dates of an unfocused month.
<a href="#">android:weekDayTextAppearance</a>	The text appearance for the week day abbreviation of the calendar header.
<a href="#">android:weekNumberColor</a>	The color for the week numbers.
<a href="#">android:weekSeparatorLineColor</a>	The color for the separator line between weeks.

#### See Also

[CalendarView\\_dateTextAppearance](#)  
[CalendarView\\_firstDayOfWeek](#)  
[CalendarView\\_focusedMonthTextColor](#)  
[CalendarView\\_maxDate](#)  
[CalendarView\\_minDate](#)  
[CalendarView\\_selectedDateVerticalBar](#)  
[CalendarView\\_selectedWeekBackgroundColor](#)  
[CalendarView\\_showWeekNumber](#)  
[CalendarView\\_shownWeekCount](#)  
[CalendarView\\_unfocusedMonthTextColor](#)  
[CalendarView\\_weekDayTextAppearance](#)  
[CalendarView\\_weekNumberColor](#)  
[CalendarView\\_weekSeparatorLineColor](#)

## public static final int[] CheckBoxPreference

Base attributes available to CheckBoxPreference.

Includes the following attributes:

Attribute	Description
<a href="#">android:disableDependentsState</a>	The state (true for on, or false for off) that causes dependents to be disabled.
<a href="#">android:summaryOff</a>	The summary for the Preference in a PreferenceActivity screen when the CheckBoxPreference is unchecked.
<a href="#">android:summaryOn</a>	The summary for the Preference in a PreferenceActivity screen when the CheckBoxPreference is checked.

#### See Also

[CheckBoxPreference\\_disableDependentsState](#)  
[CheckBoxPreference\\_summaryOff](#)  
[CheckBoxPreference\\_summaryOn](#)

## public static final int[] CheckedTextView

Attributes that can be used with a CheckedTextView.

Includes the following attributes:

Attribute	Description
<a href="#">android:checkMark</a>	Drawable used for the check mark graphic.
<a href="#">android:checked</a>	Indicates the initial checked state of this text.

#### See Also

[CheckedTextView\\_checkMark](#)  
[CheckedTextView\\_checked](#)

## public static final int[] Chronometer

Attributes that can be used with a Chronometer.

Includes the following attributes:

Attribute	Description
<a href="#">android:format</a>	Format string: if specified, the Chronometer will display this string, with the first "%s" replaced by the current timer value in "MM:SS" or "H:MM:SS" form.

**See Also**

[Chronometer\\_format](#)

**public static final int[] ClipDrawable**

Attributes that can be used with a ClipDrawable.

Includes the following attributes:

Attribute	Description
<a href="#">android:clipOrientation</a>	The orientation for the clip.
<a href="#">android:drawable</a>	Reference to a drawable resource to draw with the specified scale.
<a href="#">android:gravity</a>	Specifies where to clip within the drawable.

**See Also**

[ClipDrawable\\_clipOrientation](#)  
[ClipDrawable\\_drawable](#)  
[ClipDrawable\\_gravity](#)

**public static final int[] ColorDrawable**

Drawable used to draw a single color.

Includes the following attributes:

Attribute	Description
<a href="#">android:color</a>	The color to use.

**See Also**

[ColorDrawable\\_color](#)

**public static final int[] CompoundButton**

Attributes that can be used with a CompoundButton.

Includes the following attributes:

Attribute	Description
<a href="#">android:button</a>	Drawable used for the button graphic (e.
<a href="#">android:checked</a>	Indicates the initial checked state of this button.

**See Also**

[CompoundButton\\_button](#)  
[CompoundButton\\_checked](#)

**public static final int[] ContactsDataKind**

Maps a specific contact data MIME-type to styling information.

Includes the following attributes:

Attribute	Description
<a href="#">android:allContactsName</a>	Resource representing the term "All Contacts" (e.
<a href="#">android:detailColumn</a>	Column in data table that contains details for this data.
<a href="#">android:detailSocialSummary</a>	Flag indicating that detail should be built from SocialProvider.
<a href="#">android:icon</a>	Icon used to represent data of this kind.
<a href="#">android:mimeType</a>	Mime-type handled by this mapping.
<a href="#">android:summaryColumn</a>	Column in data table that summarizes this data.

**See Also**

[ContactsDataKind\\_allContactsName](#)  
[ContactsDataKind\\_detailColumn](#)  
[ContactsDataKind\\_detailSocialSummary](#)  
[ContactsDataKind\\_icon](#)  
[ContactsDataKindMimeType](#)  
[ContactsDataKind\\_summaryColumn](#)

**public static final int[] CycleInterpolator**

Attributes that can be used with a CycleInterpolator.

Includes the following attributes:

Attribute	Description
-----------	-------------

[android:cycles](#)

**See Also**

[CycleInterpolator cycles](#)

## public static final int[] **DatePicker**

Attributes that can be used with a DatePicker.

Includes the following attributes:

Attribute	Description
<a href="#">android:calendarViewShown</a>	Whether the calendar view is shown.
<a href="#">android:endYear</a>	The last year (inclusive), for example "2010".
<a href="#">android:maxDate</a>	The maximal date shown by this calendar view in mm/dd/yyyy format.
<a href="#">android:minDate</a>	The minimal date shown by this calendar view in mm/dd/yyyy format.
<a href="#">android:spinnersShown</a>	Whether the spinners are shown.
<a href="#">android:startYear</a>	The first year (inclusive), for example "1940".

**See Also**

[DatePicker\\_calendarViewShown](#)  
[DatePicker\\_endYear](#)  
[DatePicker\\_maxDate](#)  
[DatePicker\\_minDate](#)  
[DatePicker\\_spinnersShown](#)  
[DatePicker\\_startYear](#)

## public static final int[] **DecelerateInterpolator**

Attributes that can be used with a DecelerateInterpolator.

Includes the following attributes:

Attribute	Description
<a href="#">android:factor</a>	This is the amount of acceleration to add when easing out.

**See Also**

[DecelerateInterpolator\\_factor](#)

## public static final int[] **DeviceAdmin**

Use device-admin as the root tag of the XML resource that describes a [DeviceAdminReceiver](#) ([/reference/android/app/admin/DeviceAdminReceiver.html](#)), which is referenced from its [DEVICE\\_ADMIN\\_META\\_DATA](#) ([/reference/android/app/admin/DeviceAdminReceiver.html#DEVICE\\_ADMIN\\_META\\_DATA](#)) meta-data entry. Described here are the attributes that can be included in that tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:visible</a>	Control whether the admin is visible to the user, even when it is not enabled.

**See Also**

[DeviceAdmin\\_visible](#)

## public static final int[] **DialogPreference**

Base attributes available to DialogPreference.

Includes the following attributes:

Attribute	Description
<a href="#">android:dialogIcon</a>	The icon for the dialog.
<a href="#">android:dialogLayout</a>	A layout to be used as the content View for the dialog.
<a href="#">android:dialogMessage</a>	The message in the dialog.
<a href="#">android:dialogTitle</a>	The title in the dialog.
<a href="#">android:negativeButtonText</a>	The negative button text for the dialog.
<a href="#">android:positiveButtonText</a>	The positive button text for the dialog.

**See Also**

[DialogPreference\\_dialogIcon](#)  
[DialogPreference\\_dialogLayout](#)

[DialogPreference\\_dialogMessage](#)  
[DialogPreference\\_dialogTitle](#)  
[DialogPreference\\_negativeButtonText](#)  
[DialogPreference\\_positiveButtonText](#)

## public static final int[] Drawable

Base attributes that are available to all Drawable objects.

Includes the following attributes:

Attribute	Description
<a href="#">android:visible</a>	Provides initial visibility state of the drawable; the default value is false.

### See Also

[Drawable\\_visible](#)

## public static final int[] DrawableCorners

Describes the corners for the rectangle shape of a GradientDrawable. This can be used to render rounded corners.

Includes the following attributes:

Attribute	Description
<a href="#">android:bottomLeftRadius</a>	Radius of the bottom left corner.
<a href="#">android:bottomRightRadius</a>	Radius of the bottom right corner.
<a href="#">android:radius</a>	Defines the radius of the four corners.
<a href="#">android:topLeftRadius</a>	Radius of the top left corner.
<a href="#">android:topRightRadius</a>	Radius of the top right corner.

### See Also

[DrawableCorners\\_bottomLeftRadius](#)  
[DrawableCorners\\_bottomRightRadius](#)  
[DrawableCorners\\_radius](#)  
[DrawableCorners\\_topLeftRadius](#)  
[DrawableCorners\\_topRightRadius](#)

## public static final int[] DrawableStates

Drawable states. The mapping of Drawable states to a particular drawables is specified in the "state" elements of a Widget's "selector" element. Possible values:

- "state\_focused"
- "state\_window\_focused"
- "state\_enabled"
- "state\_checked"
- "state\_selected"
- "state\_active"
- "state\_single"
- "state\_first"
- "state\_mid"
- "state\_last"
- "state\_only"
- "state\_pressed"
- "state\_ACTIVATED"
- "state\_error"
- "state\_circle"
- "state\_rect"
- "state\_grow"
- "state\_move"
- "state\_hovered"
- "state\_drag\_can\_accept"
- "state\_drag\_hovered"
- "state\_accessibility\_focused"

Includes the following attributes:

Attribute	Description
<a href="#">android:state_accelerated</a>	State value for <a href="#">StateListDrawable</a> , indicating that the Drawable is in a view that is hardware accelerated. State value for <a href="#">StateListDrawable</a> , set when a view or its

<a href="#">android:stateActivated</a>	parent has been "activated" meaning the user has currently marked it as being of interest.
<a href="#">android:stateActive</a>	State value for <a href="#">StateListDrawable</a> .
<a href="#">android:stateCheckable</a>	State identifier indicating that the object <i>may</i> display a check mark.
<a href="#">android:stateChecked</a>	State identifier indicating that the object is currently checked.
<a href="#">android:stateDragCanAccept</a>	State for <a href="#">StateListDrawable</a> indicating that the Drawable is in a view that is capable of accepting a drop of the content currently being manipulated in a drag-and-drop operation.
<a href="#">android:stateDragHovered</a>	State for <a href="#">StateListDrawable</a> indicating that a drag operation (for which the Drawable's view is a valid recipient) is currently positioned over the Drawable.
<a href="#">android:stateEnabled</a>	State value for <a href="#">StateListDrawable</a> , set when a view is enabled.
<a href="#">android:stateFirst</a>	State value for <a href="#">StateListDrawable</a> .
<a href="#">android:stateFocused</a>	State value for <a href="#">StateListDrawable</a> , set when a view has input focus.
<a href="#">android:stateHovered</a>	State value for <a href="#">StateListDrawable</a> , set when a pointer is hovering over the view.
<a href="#">android:stateLast</a>	State value for <a href="#">StateListDrawable</a> .
<a href="#">android:stateMiddle</a>	State value for <a href="#">StateListDrawable</a> .
<a href="#">android:statePressed</a>	State value for <a href="#">StateListDrawable</a> , set when the user is pressing down in a view.
<a href="#">android:stateSelected</a>	State value for <a href="#">StateListDrawable</a> , set when a view (or one of its parents) is currently selected.
<a href="#">android:stateSingle</a>	State value for <a href="#">StateListDrawable</a> .
<a href="#">android:stateWindowFocused</a>	State value for <a href="#">StateListDrawable</a> , set when a view's window has input focus.

#### See Also

[DrawableStates.stateAccelerated](#)  
[DrawableStates.stateActivated](#)  
[DrawableStates.stateActive](#)  
[DrawableStates.stateCheckable](#)  
[DrawableStates.stateChecked](#)  
[DrawableStates.stateDragCanAccept](#)  
[DrawableStates.stateDragHovered](#)  
[DrawableStates.stateEnabled](#)  
[DrawableStates.stateFirst](#)  
[DrawableStates.stateFocused](#)  
[DrawableStates.stateHovered](#)  
[DrawableStates.stateLast](#)  
[DrawableStates.stateMiddle](#)  
[DrawableStates.statePressed](#)  
[DrawableStates.stateSelected](#)  
[DrawableStates.stateSingle](#)  
[DrawableStates.stateWindowFocused](#)

## public static final int[] Dream

Use dream as the root tag of the XML resource that describes an [DreamService](#) ([/reference/android/service/dreams/DreamService.html](#)), which is referenced from its [DREAM\\_META\\_DATA](#) ([/reference/android/service/dreams/DreamService.html#DREAM\\_META\\_DATA](#)) meta-data entry. Described here are the attributes that can be included in that tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:settingsActivity</a>	Component name of an activity that allows the user to modify the settings for this dream.

#### See Also

[Dream.settingsActivity](#)

## public static final int[] EditText

Attributes that can be used with a EditText.

## public static final int[] ExpandableListChildIndicatorState

State array representing an expandable list child's indicator.

Includes the following attributes:

Attribute	Description
<code>android:state_last</code>	State identifier indicating the child is the last child within its group.

**See Also**

[ExpandableListChildIndicatorState\\_state\\_last](#)

## public static final int[] **ExpandableListGroupIndicatorState**

State array representing an expandable list group's indicator.

Includes the following attributes:

Attribute	Description
<code>android:state_empty</code>	State identifier indicating the group is empty (has no children).
<code>android:state_expanded</code>	State identifier indicating the group is expanded.

**See Also**

[ExpandableListGroupIndicatorState\\_state\\_empty](#)

[ExpandableListGroupIndicatorState\\_state\\_expanded](#)

## public static final int[] **ExpandableListView**

Attributes that can be used with a ExpandableListView.

Includes the following attributes:

Attribute	Description
<code>android:childDivider</code>	Drawable or color that is used as a divider for children.
<code>android:childIndicator</code>	Indicator shown beside the child View.
<code>android:childIndicatorLeft</code>	The left bound for a child's indicator.
<code>android:childIndicatorRight</code>	The right bound for a child's indicator.
<code>android:groupIndicator</code>	Indicator shown beside the group View.
<code>android:indicatorLeft</code>	The left bound for an item's indicator.
<code>android:indicatorRight</code>	The right bound for an item's indicator.

**See Also**

[ExpandableListView\\_childDivider](#)

[ExpandableListView\\_childIndicator](#)

[ExpandableListView\\_childIndicatorLeft](#)

[ExpandableListView\\_childIndicatorRight](#)

[ExpandableListView\\_groupIndicator](#)

[ExpandableListView\\_indicatorLeft](#)

[ExpandableListView\\_indicatorRight](#)

## public static final int[] **Extra**

An extra data value to place into a an extra/name value pair held in a Bundle, as per [Bundle \(/reference/android/os/Bundle.html\)](#).

Includes the following attributes:

Attribute	Description
<code>android:name</code>	Required name of the extra data.
<code>android:value</code>	Concrete value to put for this named extra data.

**See Also**

[Extra\\_name](#)

[Extra\\_value](#)

## public static final int[] **Fragment**

Attributes that can be used with <fragment> tags inside of the layout of an Activity. This instantiates the given [Fragment \(/reference/android/app/Fragment.html\)](#) and inserts its content view into the current location in the layout.

Includes the following attributes:

Attribute	Description
<code>android:id</code>	Supply an identifier name for the top-level view, to later retrieve it with <code>View.findViewById()</code> or <code>Activity.findViewById()</code> .
<code>android:name</code>	Supply the name of the fragment class to instantiate.

[android:tag](#) Supply a tag for the top-level view containing a String, to be retrieved later with [View.getTag\(\)](#) or searched for with [View.findViewByIdWithTag\(\)](#).

**See Also**

[Fragment\\_id](#)  
[Fragment\\_name](#)  
[Fragment\\_tag](#)

**public static final int[] FragmentAnimation**

Fragment animation class attributes.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:fragmentCloseEnterAnimation</u></a>	
<a href="#"><u>android:fragmentCloseExitAnimation</u></a>	
<a href="#"><u>android:fragmentFadeEnterAnimation</u></a>	
<a href="#"><u>android:fragmentFadeExitAnimation</u></a>	
<a href="#"><u>android:fragmentOpenEnterAnimation</u></a>	
<a href="#"><u>android:fragmentOpenExitAnimation</u></a>	

**See Also**

[FragmentAnimation\\_fragmentCloseEnterAnimation](#)  
[FragmentAnimation\\_fragmentCloseExitAnimation](#)  
[FragmentAnimation\\_fragmentFadeEnterAnimation](#)  
[FragmentAnimation\\_fragmentFadeExitAnimation](#)  
[FragmentAnimation\\_fragmentOpenEnterAnimation](#)  
[FragmentAnimation\\_fragmentOpenExitAnimation](#)

**public static final int[] FragmentBreadCrumbs**

Attributes that can be used with <FragmentBreadCrumbs> tags.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:gravity</u></a>	Specifies how an object should position its content, on both the X and Y axes, within its own bounds.

**See Also**

[FragmentBreadCrumbs\\_gravity](#)

**public static final int[] FrameLayout**

Attributes that can be used with a FrameLayout.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:foreground</u></a>	Defines the drawable to draw over the content.
<a href="#"><u>android:foregroundGravity</u></a>	Defines the gravity to apply to the foreground drawable.
<a href="#"><u>android:measureAllChildren</u></a>	Determines whether to measure all children or just those in the VISIBLE or INVISIBLE state when measuring.

**See Also**

[FrameLayout\\_foreground](#)  
[FrameLayout\\_foregroundGravity](#)  
[FrameLayout\\_measureAllChildren](#)

**public static final int[] FrameLayout\_Layout**

Attributes that can be used with a FrameLayout\_Layout.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:layout_gravity</u></a>	Standard gravity constant that a child supplies to its parent.

**See Also**

[FrameLayout\\_Layout\\_layout\\_gravity](#)

**public static final int[] Gallery**

Attributes that can be used with a Gallery.

Includes the following attributes:

Attribute	Description
<a href="#">android:animationDuration</a>	Sets how long a transition animation should run (in milliseconds) when layout has changed.
<a href="#">android:gravity</a>	Specifies how an object should position its content, on both the X and Y axes, within its own bounds.
<a href="#">android:spacing</a>	
<a href="#">android:unselectedAlpha</a>	Sets the alpha on the items that are not selected.

**See Also**

[Gallery\\_animationDuration](#)  
[Gallery\\_gravity](#)  
[Gallery\\_spacing](#)  
[Gallery\\_unselectedAlpha](#)

**public static final int[] GestureOverlayView**

GestureOverlayView specific attributes. These attributes are used to configure a GestureOverlayView from XML.

Includes the following attributes:

Attribute	Description
<a href="#">android:eventsInterceptionEnabled</a>	Defines whether the overlay should intercept the motion events when a gesture is recognized.
<a href="#">android:fadeDuration</a>	Duration, in milliseconds, of the fade out effect after the user is done drawing a gesture.
<a href="#">android:fadeEnabled</a>	Defines whether the gesture will automatically fade out after being recognized.
<a href="#">android:fadeOffset</a>	Time, in milliseconds, to wait before the gesture fades out after the user is done drawing it.
<a href="#">android:gestureColor</a>	Color used to draw a gesture.
<a href="#">android:gestureStrokeAngleThreshold</a>	Minimum curve angle a stroke must contain before it is recognized as a gesture.
<a href="#">android:gestureStrokeLengthThreshold</a>	Minimum length of a stroke before it is recognized as a gesture.
<a href="#">android:gestureStrokeSquarenessThreshold</a>	Squareness threshold of a stroke before it is recognized as a gesture.
<a href="#">android:gestureStrokeType</a>	Defines the type of strokes that define a gesture.
<a href="#">android:gestureStrokeWidth</a>	Width of the stroke used to draw the gesture.
<a href="#">android:orientation</a>	Indicates whether horizontal (when the orientation is vertical) or vertical (when orientation is horizontal) strokes automatically define a gesture.
<a href="#">android:uncertainGestureColor</a>	Color used to draw the user's strokes until we are sure it's a gesture.

**See Also**

[GestureOverlayView\\_eventsInterceptionEnabled](#)  
[GestureOverlayView\\_fadeDuration](#)  
[GestureOverlayView\\_fadeEnabled](#)  
[GestureOverlayView\\_fadeOffset](#)  
[GestureOverlayView\\_gestureColor](#)  
[GestureOverlayView\\_gestureStrokeAngleThreshold](#)  
[GestureOverlayView\\_gestureStrokeLengthThreshold](#)  
[GestureOverlayView\\_gestureStrokeSquarenessThreshold](#)  
[GestureOverlayView\\_gestureStrokeType](#)  
[GestureOverlayView\\_gestureStrokeWidth](#)  
[GestureOverlayView\\_orientation](#)  
[GestureOverlayView\\_uncertainGestureColor](#)

**public static final int[] GlowPadView**

Attributes that can be used with a GlowPadView.

Includes the following attributes:

Attribute	Description
<a href="#">android:directionDescriptions</a>	Reference to an array resource that be used to announce the directions with targets around the circle.

<a href="#">android:gravity</a>	Specifies how an object should position its content, on both the X and Y axes, within its own bounds.
<a href="#">android:innerRadius</a>	Inner radius of glow area.
<a href="#">android:targetDescriptions</a>	Reference to an array resource that be used as description for the targets around the circle.

**See Also**

[GlowPadView\\_directionDescriptions](#)  
[GlowPadView\\_gravity](#)  
[GlowPadView\\_innerRadius](#)  
[GlowPadView\\_targetDescriptions](#)

## public static final int[] GradientDrawable

Drawable used to render a geometric shape, with a gradient or a solid color.

Includes the following attributes:

Attribute	Description
<a href="#">android:dither</a>	Enables or disables dithering.
<a href="#">android:innerRadius</a>	Inner radius of the ring.
<a href="#">android:innerRadiusRatio</a>	Inner radius of the ring expressed as a ratio of the ring's width.
<a href="#">android:shape</a>	Indicates what shape to fill with a gradient.
<a href="#">android:thickness</a>	Thickness of the ring.
<a href="#">android:thicknessRatio</a>	Thickness of the ring expressed as a ratio of the ring's width.
<a href="#">android:useLevel</a>	Indicates whether the drawable's level affects the way the gradient is drawn.
<a href="#">android:visible</a>	Indicates whether the drawable should initially be visible.

**See Also**

[GradientDrawable\\_dither](#)  
[GradientDrawable\\_innerRadius](#)  
[GradientDrawable\\_innerRadiusRatio](#)  
[GradientDrawable\\_shape](#)  
[GradientDrawable\\_thickness](#)  
[GradientDrawable\\_thicknessRatio](#)  
[GradientDrawable\\_useLevel](#)  
[GradientDrawable\\_visible](#)

## public static final int[] GradientDrawableGradient

Used to describe the gradient used to fill the shape of a GradientDrawable.

Includes the following attributes:

Attribute	Description
<a href="#">android:angle</a>	Angle of the gradient.
<a href="#">android:centerColor</a>	Optional center color.
<a href="#">android:centerX</a>	X coordinate of the origin of the gradient within the shape.
<a href="#">android:centerY</a>	Y coordinate of the origin of the gradient within the shape.
<a href="#">android:endColor</a>	End color of the gradient.
<a href="#">android:gradientRadius</a>	Radius of the gradient, used only with radial gradient.
<a href="#">android:startColor</a>	Start color of the gradient.
<a href="#">android:type</a>	Type of gradient.
<a href="#">android:useLevel</a>	

**See Also**

[GradientDrawableGradient\\_angle](#)  
[GradientDrawableGradient\\_centerColor](#)  
[GradientDrawableGradient\\_centerX](#)  
[GradientDrawableGradient\\_centerY](#)  
[GradientDrawableGradient\\_endColor](#)  
[GradientDrawableGradient\\_gradientRadius](#)  
[GradientDrawableGradient\\_startColor](#)  
[GradientDrawableGradient\\_type](#)  
[GradientDrawableGradient\\_useLevel](#)

## public static final int[] GradientDrawablePadding

Used to specify the optional padding of a GradientDrawable.

Includes the following attributes:

Attribute	Description
<code>android:bottom</code>	Amount of bottom padding inside the gradient shape.
<code>android:left</code>	Amount of left padding inside the gradient shape.
<code>android:right</code>	Amount of right padding inside the gradient shape.
<code>android:top</code>	Amount of top padding inside the gradient shape.

**See Also**

[GradientDrawablePadding\\_bottom](#)  
[GradientDrawablePadding\\_left](#)  
[GradientDrawablePadding\\_right](#)  
[GradientDrawablePadding\\_top](#)

**public static final int[] GradientDrawableSize**

Used to specify the size of the shape for GradientDrawable.

Includes the following attributes:

Attribute	Description
<code>android:height</code>	Height of the gradient shape.
<code>android:width</code>	Width of the gradient shape.

**See Also**

[GradientDrawableSize\\_height](#)  
[GradientDrawableSize\\_width](#)

**public static final int[] GradientDrawableSolid**

Used to fill the shape of GradientDrawable with a solid color.

Includes the following attributes:

Attribute	Description
<code>android:color</code>	Solid color for the gradient shape.

**See Also**

[GradientDrawableSolid\\_color](#)

**public static final int[] GradientDrawableStroke**

Used to describe the optional stroke of a GradientDrawable.

Includes the following attributes:

Attribute	Description
<code>android:color</code>	Color of the gradient shape's stroke.
<code>android:dashGap</code>	Gap between dashes in the stroke.
<code>android:dashWidth</code>	Length of a dash in the stroke.
<code>android:width</code>	Width of the gradient shape's stroke.

**See Also**

[GradientDrawableStroke\\_color](#)  
[GradientDrawableStroke\\_dashGap](#)  
[GradientDrawableStroke\\_dashWidth](#)  
[GradientDrawableStroke\\_width](#)

**public static final int[] GridLayout**

Attributes that can be used with a GridLayout.

Includes the following attributes:

Attribute	Description
<code>android:alignmentMode</code>	When set to alignMargins, causes alignment to take place between the outer boundary of a view, as defined by its margins.
<code>android:columnCount</code>	The maximum number of columns to create when automatically positioning children.
<code>android:columnOrderPreserved</code>	When set to true, forces column boundaries to appear in the same order as column indices.
<code>android:orientation</code>	The orientation property is not used during layout.
	The maximum number of rows to create when automatically

<a href="#">android:rowCount</a>	positioning children.
<a href="#">android:rowOrderPreserved</a>	When set to true, forces row boundaries to appear in the same order as row indices.
<a href="#">android:useDefaultMargins</a>	When set to true, tells GridLayout to use default margins when none are specified in a view's layout parameters.

**See Also**

[GridLayout\\_alignmentMode](#)  
[GridLayout\\_columnCount](#)  
[GridLayout\\_columnOrderPreserved](#)  
[GridLayout\\_orientation](#)  
[GridLayout\\_rowCount](#)  
[GridLayout\\_rowOrderPreserved](#)  
[GridLayout\\_useDefaultMargins](#)

### public static final int[] **GridLayoutAnimation**

Attributes that can be used with a GridLayoutAnimation.

Includes the following attributes:

Attribute	Description
<a href="#">android:columnDelay</a>	Fraction of the animation duration used to delay the beginning of the animation of each column.
<a href="#">android:direction</a>	Direction of the animation in the grid.
<a href="#">android:directionPriority</a>	Priority of the rows and columns.
<a href="#">android:rowDelay</a>	Fraction of the animation duration used to delay the beginning of the animation of each row.

**See Also**

[GridLayoutAnimation\\_columnDelay](#)  
[GridLayoutAnimation\\_direction](#)  
[GridLayoutAnimation\\_directionPriority](#)  
[GridLayoutAnimation\\_rowDelay](#)

### public static final int[] **GridLayout\_Layout**

Attributes that can be used with a GridLayout\_Layout.

Includes the following attributes:

Attribute	Description
<a href="#">android:layout_column</a>	The column boundary delimiting the left of the group of cells occupied by this view.
<a href="#">android:layout_columnSpan</a>	The column span: the difference between the right and left boundaries delimiting the group of cells occupied by this view.
<a href="#">android:layout_gravity</a>	Gravity specifies how a component should be placed in its group of cells.
<a href="#">android:layout_row</a>	The row boundary delimiting the top of the group of cells occupied by this view.
<a href="#">android:layout_rowSpan</a>	The row span: the difference between the bottom and top boundaries delimiting the group of cells occupied by this view.

**See Also**

[GridLayout\\_Layout\\_layout\\_column](#)  
[GridLayout\\_Layout\\_layout\\_columnSpan](#)  
[GridLayout\\_Layout\\_layout\\_gravity](#)  
[GridLayout\\_Layout\\_layout\\_row](#)  
[GridLayout\\_Layout\\_layout\\_rowSpan](#)

### public static final int[] **GridView**

Attributes that can be used with a GridView.

Includes the following attributes:

Attribute	Description
<a href="#">android:columnWidth</a>	Specifies the fixed width for each column.
<a href="#">android:gravity</a>	Specifies the gravity within each cell.
<a href="#">android:horizontalSpacing</a>	Defines the default horizontal spacing between columns.
<a href="#">android:numColumns</a>	Defines how many columns to show.
<a href="#">android:stretchMode</a>	Defines how columns should stretch to fill the available empty space,

if any.  
[android:verticalSpacing](#) Defines the default vertical spacing between rows.

**See Also**

[GridView\\_columnWidth](#)  
[GridView\\_gravity](#)  
[GridView\\_horizontalSpacing](#)  
[GridView\\_numColumns](#)  
[GridView\\_stretchMode](#)  
[GridView\\_verticalSpacing](#)

**public static final int[] HorizontalScrollView**

Attributes that can be used with a HorizontalScrollView.

Includes the following attributes:

<b>Attribute</b>	<b>Description</b>
<u><a href="#">android:fillViewport</a></u>	Defines whether the scrollview should stretch its content to fill the viewport.

**See Also**

[HorizontalScrollView\\_fillViewport](#)

**public static final int[] Icon**

Attributes that can be used with a Icon.

Includes the following attributes:

<b>Attribute</b>	<b>Description</b>
<u><a href="#">android:icon</a></u>	A Drawable resource providing a graphical representation of its associated item.
<u><a href="#">android:mimeType</a></u>	Specify a MIME type that is handled, as per <u><a href="#">IntentFilter.addDataType()</a></u> .

**See Also**

[Icon\\_icon](#)  
[Icon\\_mimeType](#)

**public static final int[] IconDefault**

Attributes that can be used with a IconDefault.

Includes the following attributes:

<b>Attribute</b>	<b>Description</b>
<u><a href="#">android:icon</a></u>	A Drawable resource providing a graphical representation of its associated item.

**See Also**

[IconDefault\\_icon](#)

**public static final int[] IconMenuItemView**

Attributes that can be used with a IconMenuItemView.

Includes the following attributes:

<b>Attribute</b>	<b>Description</b>
<u><a href="#">android:maxItemsPerRow</a></u>	Defines the maximum number of items per row.
<u><a href="#">android:maxRows</a></u>	Defines the maximum number of rows displayed.
<u><a href="#">android:moreIcon</a></u>	'More' icon.
<u><a href="#">android:rowHeight</a></u>	Defines the height of each row.

**See Also**

[IconMenuItemView\\_maxItemsPerRow](#)  
[IconMenuItemView\\_maxRows](#)  
[IconMenuItemView\\_moreIcon](#)  
[IconMenuItemView\\_rowHeight](#)

**public static final int[] ImageSwitcher**

Attributes that can be used with a ImageSwitcher.

**public static final int[] ImageView**

Attributes that can be used with a ImageView.

Includes the following attributes:

Attribute	Description
<a href="#">android:adjustViewBounds</a>	Set this to true if you want the ImageView to adjust its bounds to preserve the aspect ratio of its drawable.
<a href="#">android:baseline</a>	The offset of the baseline within this view.
<a href="#">android:baselineAlignBottom</a>	If true, the image view will be baseline aligned with based on its bottom edge.
<a href="#">android:cropToPadding</a>	If true, the image will be cropped to fit within its padding.
<a href="#">android:maxHeight</a>	An optional argument to supply a maximum height for this view.
<a href="#">android: maxWidth</a>	An optional argument to supply a maximum width for this view.
<a href="#">android:scaleType</a>	Controls how the image should be resized or moved to match the size of this ImageView.
<a href="#">android:src</a>	Sets a drawable as the content of this ImageView.
<a href="#">android:tint</a>	Set a tinting color for the image.

**See Also**

[ImageView\\_adjustViewBounds](#)  
[ImageView\\_baseline](#)  
[ImageView\\_baselineAlignBottom](#)  
[ImageView\\_cropToPadding](#)  
[ImageView\\_maxHeight](#)  
[ImageView\\_maxWidth](#)  
[ImageView\\_scaleType](#)  
[ImageView\\_src](#)  
[ImageView\\_tint](#)

**public static final int[] InputExtras**

An input-extras is a container for extra data to supply to an input method. Contains one or more more [/reference/android/R.styleable.html#Extra](#) tags.

**public static final int[] InputMethod**

Use input-method as the root tag of the XML resource that describes an [InputMethod](#) [/reference/android/view/inputmethod/InputMethod.html](#) service, which is referenced from its [SERVICE\\_META\\_DATA](#) [/reference/android/view/inputmethod/InputMethod.html#SERVICE\\_META\\_DATA](#) meta-data entry. Described here are the attributes that can be included in that tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:isDefault</a>	Set to true in all of the configurations for which this input method should be considered an option as the default.
<a href="#">android:settingsActivity</a>	Component name of an activity that allows the user to modify the settings for this service.

**See Also**

[InputMethod\\_isDefault](#)  
[InputMethod\\_settingsActivity](#)

**public static final int[] InputMethodService**

Attributes that can be used with a InputMethodService.

Includes the following attributes:

Attribute	Description
<a href="#">android:imeExtractEnterAnimation</a>	Animation to use when showing the fullscreen extract UI after it had previously been hidden.
<a href="#">android:imeExtractExitAnimation</a>	Animation to use when hiding the fullscreen extract UI after it had previously been shown.
<a href="#">android:imeFullscreenBackground</a>	Background to use for entire input method when it is being shown in fullscreen mode with the extract view, to ensure that it completely covers the application.

**See Also**

[InputMethodService\\_imeExtractEnterAnimation](#)  
[InputMethodService\\_imeExtractExitAnimation](#)  
[InputMethodService\\_imeFullscreenBackground](#)

**public static final int[] InputMethod\_Subtype**

This is the subtype of InputMethod. Subtype can describe locales (e.g. en\_US, fr\_FR...) and modes (e.g. voice, keyboard...), and is used for IME switch. This subtype allows the system to call the specified subtype of the IME directly.

Includes the following attributes:

Attribute	Description
<a href="#">android:icon</a>	The icon of the subtype.
<a href="#">android:imeSubtypeExtraValue</a>	The extra value of the subtype.
<a href="#">android:imeSubtypeLocale</a>	The locale of the subtype.
<a href="#">android:imeSubtypeMode</a>	The mode of the subtype.
<a href="#">android:isAuxiliary</a>	Set true if the subtype is auxiliary.
<a href="#">android:label</a>	The name of the subtype.
<a href="#">android:overridesImplicitlyEnabledSubtype</a>	Set true when this subtype should be selected by default if no other subtypes are selected explicitly.
<a href="#">android:subtypeId</a>	The unique id for the subtype.

**See Also**

[InputMethod\\_Subtype\\_icon](#)  
[InputMethod\\_Subtype\\_imeSubtypeExtraValue](#)  
[InputMethod\\_Subtype\\_imeSubtypeLocale](#)  
[InputMethod\\_Subtype\\_imeSubtypeMode](#)  
[InputMethod\\_Subtype\\_isAuxiliary](#)  
[InputMethod\\_Subtype\\_label](#)  
[InputMethod\\_Subtype\\_overridesImplicitlyEnabledSubtype](#)  
[InputMethod\\_Subtype\\_subtypeId](#)

**public static final int[] InsetDrawable**

Attributes that can be used with a InsetDrawable.

Includes the following attributes:

Attribute	Description
<a href="#">android:drawable</a>	Reference to a drawable resource to use for the frame.
<a href="#">android:insetBottom</a>	
<a href="#">android:insetLeft</a>	
<a href="#">android:insetRight</a>	
<a href="#">android:insetTop</a>	
<a href="#">android:visible</a>	Provides initial visibility state of the drawable; the default value is false.

**See Also**

[InsetDrawable\\_drawable](#)  
[InsetDrawable\\_insetBottom](#)  
[InsetDrawable\\_insetLeft](#)  
[InsetDrawable\\_insetRight](#)  
[InsetDrawable\\_insetTop](#)  
[InsetDrawable\\_visible](#)

**public static final int[] Intent**

Declaration of an [Intent](#) ([reference/android/content/Intent.html](#)) object in XML. May also include zero or more [and {@link #Extra \(@reference/android/R.styleable.html#IntentCategory\)}](#) tags.

Includes the following attributes:

Attribute	Description
<a href="#">android:action</a>	The action name to assign to the Intent, as per <a href="#">Intent.setAction()</a> .
<a href="#">android:data</a>	The data URI to assign to the Intent, as per <a href="#">Intent.setData()</a> .
<a href="#">android:mimeType</a>	The MIME type name to assign to the Intent, as per <a href="#">Intent.setType()</a> .
<a href="#">android:targetClass</a>	The class part of the ComponentName to assign to the Intent, as per <a href="#">Intent.setComponent()</a> .
<a href="#">android:targetPackage</a>	The package part of the ComponentName to assign to the Intent, as per <a href="#">Intent.setComponent()</a> .

**See Also**

[Intent\\_action](#)  
[Intent\\_data](#)  
[Intent\\_mimeType](#)  
[Intent\\_targetClass](#)

## Intent\_targetPackage

public static final int[] IntentCategory

A category to add to an Intent, as per [Intent.addCategory\(\)](#)  
[\(/reference/android/content/Intent.html#addCategory\(java.lang.String\)\)](#).

Includes the following attributes:

Attribute	Description
<a href="#">android:name</a>	Required name of the category.

### See Also

[IntentCategory\\_name](#)

public static final int[] Keyboard

Attributes that can be used with a Keyboard.

Includes the following attributes:

Attribute	Description
<a href="#">android:horizontalGap</a>	Default horizontal gap between keys.
<a href="#">android:keyHeight</a>	Default height of a key, in pixels or percentage of display width.
<a href="#">android:keyWidth</a>	Default width of a key, in pixels or percentage of display width.
<a href="#">android:verticalGap</a>	Default vertical gap between rows of keys.

### See Also

[Keyboard\\_horizontalGap](#)  
[Keyboard\\_keyHeight](#)  
[Keyboard\\_keyWidth](#)  
[Keyboard\\_verticalGap](#)

public static final int[] KeyboardLayout

Use keyboard-layouts as the root tag of the XML resource that describes a collection of keyboard layouts provided by an application. Each keyboard layout is declared by a keyboard-layout tag with these attributes. The XML resource that contains the keyboard layouts must be referenced from its [META\\_DATA\\_KEYBOARD\\_LAYOUTS](#)  
[\(/reference/android/hardware/input/InputManager.html#META\\_DATA\\_KEYBOARD\\_LAYOUTS\)](#) meta-data entry used with broadcast receivers for [ACTION\\_QUERY\\_KEYBOARD\\_LAYOUTS](#)  
[\(/reference/android/hardware/input/InputManager.html#ACTION\\_QUERY\\_KEYBOARD\\_LAYOUTS\)](#).

Includes the following attributes:

Attribute	Description
<a href="#">android:keyboardLayout</a>	The key character map file resource.
<a href="#">android:label</a>	The display label of the keyboard layout.
<a href="#">android:name</a>	The name of the keyboard layout, must be unique in the receiver.

### See Also

[KeyboardLayout\\_keyboardLayout](#)  
[KeyboardLayout\\_label](#)  
[KeyboardLayout\\_name](#)

public static final int[] KeyboardView

Attributes that can be used with a KeyboardView.

Includes the following attributes:

Attribute	Description
<a href="#">android:keyBackground</a>	Image for the key.
<a href="#">android:keyPreviewHeight</a>	Height of the key press feedback popup.
<a href="#">android:keyPreviewLayout</a>	Layout resource for key press feedback.
<a href="#">android:keyPreviewOffset</a>	Vertical offset of the key press feedback from the key.
<a href="#">android:keyTextColor</a>	Color to use for the label in a key.
<a href="#">android:keyTextSize</a>	Size of the text for character keys.
<a href="#">android:labelTextSize</a>	Size of the text for custom keys with some text and no icon.
<a href="#">android:popupLayout</a>	Layout resource for popup keyboards.
<a href="#">android:shadowColor</a>	Place a shadow of the specified color behind the text.

[android:shadowRadius](#) Radius of the shadow.  
[android:verticalCorrection](#) Amount to offset the touch Y coordinate by, for bias correction.

**See Also**

[KeyboardView keyBackground](#)  
[KeyboardView keyPreviewHeight](#)  
[KeyboardView keyPreviewLayout](#)  
[KeyboardView keyPreviewOffset](#)  
[KeyboardView keyTextColor](#)  
[KeyboardView keyTextSize](#)  
[KeyboardView labelTextSize](#)  
[KeyboardView popupLayout](#)  
[KeyboardView shadowColor](#)  
[KeyboardView shadowRadius](#)  
[KeyboardView verticalCorrection](#)

**public static final int[] KeyboardViewPreviewState**

Attributes that can be used with a KeyboardViewPreviewState.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:state_long_pressable</u></a>	State for <code>KeyboardView</code> key preview background.

**See Also**

[KeyboardViewPreviewState\\_state\\_long\\_pressable](#)

**public static final int[] Keyboard\_Key**

Attributes that can be used with a Keyboard\_Key.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:codes</u></a>	The unicode value or comma-separated values that this key outputs.
<a href="#"><u>android:iconPreview</u></a>	The icon to show in the popup preview.
<a href="#"><u>android:isModifier</u></a>	Whether this is a modifier key such as Alt or Shift.
<a href="#"><u>android:isRepeatable</u></a>	Whether long-pressing on this key will make it repeat.
<a href="#"><u>android:isSticky</u></a>	Whether this is a toggle key.
<a href="#"><u>android:keyEdgeFlags</u></a>	Key edge flags.
<a href="#"><u>android:keyIcon</u></a>	The icon to display on the key instead of the label.
<a href="#"><u>android:keyLabel</u></a>	The label to display on the key.
<a href="#"><u>android:keyOutputText</u></a>	The string of characters to output when this key is pressed.
<a href="#"><u>android:keyboardMode</u></a>	Mode of the keyboard.
<a href="#"><u>android:popupCharacters</u></a>	The characters to display in the popup keyboard.
<a href="#"><u>android:popupKeyboard</u></a>	The XML keyboard layout of any popup keyboard.

**See Also**

[Keyboard\\_Key\\_codes](#)  
[Keyboard\\_Key\\_iconPreview](#)  
[Keyboard\\_Key\\_isModifier](#)  
[Keyboard\\_Key\\_isRepeatable](#)  
[Keyboard\\_Key\\_isSticky](#)  
[Keyboard\\_Key\\_keyEdgeFlags](#)  
[Keyboard\\_Key\\_keyIcon](#)  
[Keyboard\\_Key\\_keyLabel](#)  
[Keyboard\\_Key\\_keyOutputText](#)  
[Keyboard\\_Key\\_keyboardMode](#)  
[Keyboard\\_Key\\_popupCharacters](#)  
[Keyboard\\_Key\\_popupKeyboard](#)

**public static final int[] Keyboard\_Row**

Attributes that can be used with a Keyboard\_Row.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:keyboardMode</u></a>	Mode of the keyboard.
<a href="#"><u>android:rowEdgeFlags</u></a>	Row edge flags.

**See Also**

[Keyboard\\_Row\\_keyboardMode](#)  
[Keyboard\\_Row\\_rowEdgeFlags](#)

**public static final int[] KeyguardGlowStripView**

Attributes that can be used with a KeyguardGlowStripView.

Includes the following attributes:

**Attribute Description**

**public static final int[] KeyguardSecurityViewFlipper\_Layout**

Attributes that can be used with a KeyguardSecurityViewFlipper\_Layout.

Includes the following attributes:

**Attribute Description**

**public static final int[] LayerDrawable**

Drawable used to render several drawables stacked on top of each other. Each child drawable can be controlled individually.

Includes the following attributes:

Attribute	Description
<a href="#">android:opacity</a>	Indicates the opacity of the layer.

**See Also**

[LayerDrawable\\_opacity](#)

**public static final int[] LayerDrawableItem**

Describes an item (or child) of a LayerDrawable.

Includes the following attributes:

Attribute	Description
<a href="#">android:bottom</a>	Bottom coordinate of the layer.
<a href="#">android:drawable</a>	Drawable used to render the layer.
<a href="#">android:id</a>	Identifier of the layer.
<a href="#">android:left</a>	Left coordinate of the layer.
<a href="#">android:right</a>	Right coordinate of the layer.
<a href="#">android:top</a>	Top coordinate of the layer.

**See Also**

[LayerDrawableItem\\_bottom](#)  
[LayerDrawableItem\\_drawable](#)  
[LayerDrawableItem\\_id](#)  
[LayerDrawableItem\\_left](#)  
[LayerDrawableItem\\_right](#)  
[LayerDrawableItem\\_top](#)

**public static final int[] LayoutAnimation**

Attributes that can be used with a LayoutAnimation.

Includes the following attributes:

Attribute	Description
<a href="#">android:animation</a>	Animation to use on each child.
<a href="#">android:animationOrder</a>	The order in which the animations will be started.
<a href="#">android:delay</a>	Fraction of the animation duration used to delay the beginning of the animation of each child.
<a href="#">android:interpolator</a>	Interpolator used to interpolate the delay between the start of each animation.

**See Also**

[LayoutAnimation\\_animation](#)  
[LayoutAnimation\\_animationOrder](#)  
[LayoutAnimation\\_delay](#)

[LayoutAnimation\\_interpolator](#)

**public static final int[] LevelListDrawableItem**

Attributes that can be used with a LevelListDrawableItem.

Includes the following attributes:

Attribute	Description
<a href="#">android:drawable</a>	Reference to a drawable resource to use for the frame.
<a href="#">android:maxLevel</a>	The maximum level allowed for this item.
<a href="#">android:minLevel</a>	The minimum level allowed for this item.

**See Also**

[LevelListDrawableItem.drawable](#)  
[LevelListDrawableItem.maxLevel](#)  
[LevelListDrawableItem.minLevel](#)

**public static final int[] LinearLayout**

Attributes that can be used with a LinearLayout.

Includes the following attributes:

Attribute	Description
<a href="#">android:baselineAligned</a>	When set to false, prevents the layout from aligning its children's baselines.
<a href="#">android:baselineAlignedChildIndex</a>	When a linear layout is part of another layout that is baseline aligned, it can specify which of its children to baseline align to (that is, which child TextView).
<a href="#">android:divider</a>	Drawable to use as a vertical divider between buttons.
<a href="#">android:dividerPadding</a>	Size of padding on either end of a divider.
<a href="#">android:gravity</a>	Specifies how an object should position its content, on both the X and Y axes, within its own bounds.
<a href="#">android:measureWithLargestChild</a>	When set to true, all children with a weight will be considered having the minimum size of the largest child.
<a href="#">android:orientation</a>	Should the layout be a column or a row? Use "horizontal" for a row, "vertical" for a column.
<a href="#">android:showDividers</a>	Setting for which dividers to show.
<a href="#">android:weightSum</a>	Defines the maximum weight sum.

**See Also**

[LinearLayout.baselineAligned](#)  
[LinearLayout.baselineAlignedChildIndex](#)  
[LinearLayout.divider](#)  
[LinearLayout.dividerPadding](#)  
[LinearLayout.gravity](#)  
[LinearLayout.measureWithLargestChild](#)  
[LinearLayout.orientation](#)  
[LinearLayout.showDividers](#)  
[LinearLayout.weightSum](#)

**public static final int[] LinearLayout\_Layout**

Attributes that can be used with a LinearLayout\_Layout.

Includes the following attributes:

Attribute	Description
<a href="#">android:layout_gravity</a>	Standard gravity constant that a child supplies to its parent.
<a href="#">android:layout_height</a>	Specifies the basic height of the view.
<a href="#">android:layout_weight</a>	
<a href="#">android:layout_width</a>	Specifies the basic width of the view.

**See Also**

[LinearLayout\\_Layout.layout\\_gravity](#)  
[LinearLayout\\_Layout.layout\\_height](#)  
[LinearLayout\\_Layout.layout\\_weight](#)  
[LinearLayout\\_Layout.layout\\_width](#)

**public static final int[] ListPreference**

Base attributes available to ListPreference.

Includes the following attributes:

Attribute	Description
<a href="#">android:entries</a>	The human-readable array to present as a list.
<a href="#">android:entryValues</a>	The array to find the value to save for a preference when an entry from entries is selected.

**See Also**

[ListPreference\\_entries](#)  
[ListPreference\\_entryValues](#)

## public static final int[] ListView

Attributes that can be used with a ListView.

Includes the following attributes:

Attribute	Description
<a href="#">android:divider</a>	Drawable or color to draw between list items.
<a href="#">android:dividerHeight</a>	Height of the divider.
<a href="#">android:entries</a>	Reference to an array resource that will populate the ListView.
<a href="#">android:footerDividersEnabled</a>	When set to false, the ListView will not draw the divider before each footer view.
<a href="#">android:headerDividersEnabled</a>	When set to false, the ListView will not draw the divider after each header view.
<a href="#">android:overScrollFooter</a>	Drawable to draw below list content.
<a href="#">android:overScrollHeader</a>	Drawable to draw above list content.

**See Also**

[ListView\\_divider](#)  
[ListView\\_dividerHeight](#)  
[ListView\\_entries](#)  
[ListView\\_footerDividersEnabled](#)  
[ListView\\_headerDividersEnabled](#)  
[ListView\\_overScrollFooter](#)  
[ListView\\_overScrollHeader](#)

## public static final int[] LockPatternView

Attributes that can be used with a LockPatternView.

Includes the following attributes:

**Attribute Description**

## public static final int[] MapView

The set of attributes for a MapView.

Includes the following attributes:

Attribute	Description
<a href="#">android:apiKey</a>	Value is a string that specifies the Maps API Key to use.

**See Also**

[MapView\\_apiKey](#)

## public static final int[] MediaRouteButton

Attributes that can be used with a MediaRouteButton.

Includes the following attributes:

Attribute	Description
<a href="#">android:mediaRouteTypes</a>	The types of media routes the button and its resulting chooser will filter by.
<a href="#">android:minHeight</a>	
<a href="#">android:minWidth</a>	

**See Also**

[MediaRouteButton\\_mediaRouteTypes](#)  
[MediaRouteButton\\_minHeight](#)

[MediaRouteButton\\_minWidth](#)

**public static final int[] Menu**

Base attributes that are available to all Menu objects.

**public static final int[] MenuGroup**

Base attributes that are available to all groups.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:checkableBehavior</u></a>	Whether the items are capable of displaying a check mark.
<a href="#"><u>android:enabled</u></a>	Whether the items are enabled.
<a href="#"><u>android:id</u></a>	The ID of the group.
<a href="#"><u>android:menuCategory</u></a>	The category applied to all items within this group.
<a href="#"><u>android:orderInCategory</u></a>	The order within the category applied to all items within this group.
<a href="#"><u>android:visible</u></a>	Whether the items are shown/visible.

**See Also**

[MenuGroup\\_checkableBehavior](#)  
[MenuGroup\\_enabled](#)  
[MenuGroup\\_id](#)  
[MenuGroup\\_menuCategory](#)  
[MenuGroup\\_orderInCategory](#)  
[MenuGroup\\_visible](#)

**public static final int[] MenuItem**

Base attributes that are available to all Item objects.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:actionLayout</u></a>	An optional layout to be used as an action view.
<a href="#"><u>android:actionProviderClass</u></a>	The name of an optional ActionProvider class to instantiate an action view and perform operations such as default action for that menu item.
<a href="#"><u>android:actionViewClass</u></a>	The name of an optional View class to instantiate and use as an action view.
<a href="#"><u>android:alphabeticShortcut</u></a>	The alphabetic shortcut key.
<a href="#"><u>android:checkable</u></a>	Whether the item is capable of displaying a check mark.
<a href="#"><u>android:checked</u></a>	Whether the item is checked.
<a href="#"><u>android:enabled</u></a>	Whether the item is enabled.
<a href="#"><u>android:icon</u></a>	The icon associated with this item.
<a href="#"><u>android:id</u></a>	The ID of the item.
<a href="#"><u>android:menuCategory</u></a>	The category applied to the item.
<a href="#"><u>android:numericShortcut</u></a>	The numeric shortcut key.
<a href="#"><u>android:onClick</u></a>	Name of a method on the Context used to inflate the menu that will be called when the item is clicked.
<a href="#"><u>android:orderInCategory</u></a>	The order within the category applied to the item.
<a href="#"><u>android:showAsAction</u></a>	How this item should display in the Action Bar, if present.
<a href="#"><u>android:title</u></a>	The title associated with the item.
<a href="#"><u>android:titleCondensed</u></a>	The condensed title associated with the item.
<a href="#"><u>android:visible</u></a>	Whether the item is shown/visible.

**See Also**

[MenuItem\\_actionLayout](#)  
[MenuItem\\_actionProviderClass](#)  
[MenuItem\\_actionViewClass](#)  
[MenuItem\\_alphabeticShortcut](#)  
[MenuItem\\_checkable](#)  
[MenuItem\\_checked](#)  
[MenuItem\\_enabled](#)  
[MenuItem\\_icon](#)  
[MenuItem\\_id](#)  
[MenuItem\\_menuCategory](#)  
[MenuItem\\_numericShortcut](#)  
[MenuItem\\_onClick](#)

[MenuItem\\_orderInCategory](#)  
[MenuItem\\_showAsAction](#)  
[MenuItem\\_title](#)  
[MenuItem\\_titleCondensed](#)  
[MenuItem\\_visible](#)

### public static final int[] MenuItemCheckedFocusedState

State array representing a menu item that is currently focused and checked.

Includes the following attributes:

Attribute	Description
<a href="#">android:state_checkable</a>	State identifier indicating that the object <i>may</i> display a check mark.
<a href="#">android:state_checked</a>	State identifier indicating that the object is currently checked.
<a href="#">android:state_focused</a>	State value for <a href="#">StateListDrawable</a> set when a view has input focus.

#### See Also

[MenuItemCheckedFocusedState\\_state\\_checkable](#)  
[MenuItemCheckedFocusedState\\_state\\_checked](#)  
[MenuItemCheckedFocusedState\\_state\\_focused](#)

### public static final int[] MenuItemCheckedState

State array representing a menu item that is currently checked.

Includes the following attributes:

Attribute	Description
<a href="#">android:state_checkable</a>	State identifier indicating that the object <i>may</i> display a check mark.
<a href="#">android:state_checked</a>	State identifier indicating that the object is currently checked.

#### See Also

[MenuItemCheckedState\\_state\\_checkable](#)  
[MenuItemCheckedState\\_state\\_checked](#)

### public static final int[] MenuItemUncheckedFocusedState

State array representing a menu item that is focused and checkable but is not currently checked.

Includes the following attributes:

Attribute	Description
<a href="#">android:state_checkable</a>	State identifier indicating that the object <i>may</i> display a check mark.
<a href="#">android:state_focused</a>	State value for <a href="#">StateListDrawable</a> set when a view has input focus.

#### See Also

[MenuItemUncheckedFocusedState\\_state\\_checkable](#)  
[MenuItemUncheckedFocusedState\\_state\\_focused](#)

### public static final int[] MenuItemUncheckedState

State array representing a menu item that is checkable but is not currently checked.

Includes the following attributes:

Attribute	Description
<a href="#">android:state_checkable</a>	State identifier indicating that the object <i>may</i> display a check mark.

#### See Also

[MenuItemUncheckedState\\_state\\_checkable](#)

### public static final int[] MenuView

Attributes that can be used with a MenuView.

Includes the following attributes:

Attribute	Description
<a href="#">android:headerBackground</a>	Default background for the menu header.
<a href="#">android:horizontalDivider</a>	Default horizontal divider between rows of menu items.
<a href="#">android:itemBackground</a>	Default background for each menu item.
<a href="#">android:itemIconDisabledAlpha</a>	Default disabled icon alpha for each menu item that shows an icon.

<a href="#">android:itemTextAppearance</a>	Default appearance of menu item text.
<a href="#">android:verticalDivider</a>	Default vertical divider between menu items.
<a href="#">android:windowAnimationStyle</a>	Default animations for the menu.

**See Also**

<a href="#">MenuView_headerBackground</a>
<a href="#">MenuView_horizontalDivider</a>
<a href="#">MenuView_itemBackground</a>
<a href="#">MenuView_itemIconDisabledAlpha</a>
<a href="#">MenuView_itemTextAppearance</a>
<a href="#">MenuView_verticalDivider</a>
<a href="#">MenuView_windowAnimationStyle</a>

## public static final int[] MipmapDrawableItem

Attributes that can be used with a MipmapDrawableItem.

Includes the following attributes:

Attribute	Description
<a href="#">android:drawable</a>	Reference to a drawable resource to use for the frame.

**See Also**

<a href="#">MipmapDrawableItem.drawable</a>
---

## public static final int[] MultiPaneChallengeLayout

Attributes that can be used with a MultiPaneChallengeLayout.

Includes the following attributes:

Attribute	Description
<a href="#">android:orientation</a>	Influences how layout_centerWithinArea behaves

**See Also**

<a href="#">MultiPaneChallengeLayout.orientation</a>
--

## public static final int[] MultiPaneChallengeLayout\_Layout

Attributes that can be used with a MultiPaneChallengeLayout\_Layout.

Includes the following attributes:

Attribute	Description
<a href="#">android:layout_gravity</a>	Standard gravity constant that a child supplies to its parent.

**See Also**

<a href="#">MultiPaneChallengeLayout_Layout.layout_gravity</a>
--

## public static final int[] MultiSelectListPreference

Attributes that can be used with a MultiSelectListPreference.

Includes the following attributes:

Attribute	Description
<a href="#">android:entries</a>	The human-readable array to present as a list.
<a href="#">android:entryValues</a>	The array to find the value to save for a preference when an entry from entries is selected.

**See Also**

<a href="#">MultiSelectListPreference.entries</a>
<a href="#">MultiSelectListPreference.entryValues</a>

## public static final int[] MultiWaveView

Attributes that can be used with a MultiWaveView.

Includes the following attributes:

Attribute	Description
<a href="#">android:directionDescriptions</a>	Reference to an array resource that be used to announce the directions with targets around the circle.
<a href="#">android:targetDescriptions</a>	Reference to an array resource that be used as description for the targets around the circle.

**See Also**

[MultiWaveView directionDescriptions](#)  
[MultiWaveView targetDescriptions](#)

**public static final int[] NinePatchDrawable**

Drawable used to draw 9-patches.

Includes the following attributes:

Attribute	Description
<a href="#">android:dither</a>	Enables or disables dithering of the bitmap if the bitmap does not have the same pixel configuration as the screen (for instance: a ARGB 8888 bitmap with an RGB 565 screen).
<a href="#">android:src</a>	Identifier of the bitmap file.

**See Also**

[NinePatchDrawable\\_dither](#)  
[NinePatchDrawable\\_src](#)

**public static final int[] NumPadKey**

Attributes that can be used with a NumPadKey.

Includes the following attributes:

**Attribute Description**

**public static final int[] OvershootInterpolator**

Attributes that can be used with a OvershootInterpolator.

Includes the following attributes:

Attribute	Description
<a href="#">android:tension</a>	This is the amount of tension.

**See Also**

[OvershootInterpolator\\_tension](#)

**public static final int[] PagedView**

PagedView specific attributes. These attributes are used to customize a PagedView view in XML files.

Includes the following attributes:

**Attribute Description**

**public static final int[] Pointer**

Attributes that can be used with a Pointer.

Includes the following attributes:

**Attribute Description**

**public static final int[] PointerIcon**

Attributes that can be used with a PointerIcon.

Includes the following attributes:

**Attribute Description**

**public static final int[] PopupWindow**

Attributes that can be used with a PopupWindow.

Includes the following attributes:

Attribute	Description
<a href="#">android:popupAnimationStyle</a>	
<a href="#">android:popupBackground</a>	

**See Also**

[PopupWindow\\_popupAnimationStyle](#)  
[PopupWindow\\_popupBackground](#)

### public static final int[] PopupWindowBackgroundState

Attributes that can be used with a PopupWindowBackgroundState.

Includes the following attributes:

Attribute	Description
<a href="#">android:state_above_anchor</a>	State identifier indicating the popup will be above the anchor.

#### See Also

[PopupWindowBackgroundState\\_state\\_above\\_anchor](#)

### public static final int[] Preference

Base attributes available to Preference.

Includes the following attributes:

Attribute	Description
<a href="#">android:defaultValue</a>	The default value for the preference, which will be set either if persistence is off or persistence is on and the preference is not found in the persistent storage.
<a href="#">android:dependency</a>	The key of another Preference that this Preference will depend on.
<a href="#">android:enabled</a>	Whether the Preference is enabled.
<a href="#">android:fragment</a>	When used inside of a modern PreferenceActivity, this declares a new PreferenceFragment to be shown when the user selects this item.
<a href="#">android:icon</a>	The optional icon for the preference
<a href="#">android:key</a>	The key to store the Preference value.
<a href="#">android:layout</a>	The layout for the Preference in a PreferenceActivity screen.
<a href="#">android:order</a>	The order for the Preference (lower values are to be ordered first).
<a href="#">android:persistent</a>	Whether the Preference stores its value to the shared preferences.
<a href="#">android:selectable</a>	Whether the Preference is selectable.
<a href="#">android:shouldDisableView</a>	Whether the view of this Preference should be disabled when this Preference is disabled.
<a href="#">android:summary</a>	The summary for the Preference in a PreferenceActivity screen.
<a href="#">android:title</a>	The title for the Preference in a PreferenceActivity screen.
<a href="#">android:widgetLayout</a>	The layout for the controllable widget portion of a Preference.

#### See Also

[Preference\\_defaultValue](#)  
[Preference\\_dependency](#)  
[Preference\\_enabled](#)  
[Preference\\_fragment](#)  
[Preference\\_icon](#)  
[Preference\\_key](#)  
[Preference\\_layout](#)  
[Preference\\_order](#)  
[Preference\\_persistent](#)  
[Preference\\_selectable](#)  
[Preference\\_shouldDisableView](#)  
[Preference\\_summary](#)  
[Preference\\_title](#)  
[Preference\\_widgetLayout](#)

### public static final int[] PreferenceFrameLayout

Attributes that can be used with a PreferenceFrameLayout.

Includes the following attributes:

#### Attribute Description

### public static final int[] PreferenceFrameLayout\_Layout

Attributes that can be used with a PreferenceFrameLayout\_Layout.

Includes the following attributes:

#### Attribute Description

## **public static final int[] PreferenceGroup**

Base attributes available to PreferenceGroup.

Includes the following attributes:

Attribute	Description
<a href="#">android:orderingFromXml</a>	Whether to order the Preference under this group as they appear in the XML file.

### See Also

[PreferenceGroup\\_orderingFromXml](#)

## **public static final int[] PreferenceHeader**

Attribute for a header describing the item shown in the top-level list from which the selects the set of preference to dig in to.

Includes the following attributes:

Attribute	Description
<a href="#">android:breadCrumbShortTitle</a>	The short title for the bread crumb of this item.
<a href="#">android:breadCrumbTitle</a>	The title for the bread crumb of this item.
<a href="#">android:fragment</a>	The fragment that is displayed when the user selects this item.
<a href="#">android:icon</a>	An icon for the item.
<a href="#">android:id</a>	Identifier value for the header.
<a href="#">android:summary</a>	The summary for the item.
<a href="#">android:title</a>	The title of the item that is shown to the user.

### See Also

[PreferenceHeader\\_breadCrumbShortTitle](#)  
[PreferenceHeader\\_breadCrumbTitle](#)  
[PreferenceHeader\\_fragment](#)  
[PreferenceHeader\\_icon](#)  
[PreferenceHeader\\_id](#)  
[PreferenceHeader\\_summary](#)  
[PreferenceHeader\\_title](#)

## **public static final int[] ProgressBar**

This field is deprecated.

Not used by the framework.

Attributes that can be used with a ProgressBar.

Includes the following attributes:

Attribute	Description
<a href="#">android:animationResolution</a>	Timeout between frames of animation in milliseconds
<a href="#">android:indeterminate</a>	Allows to enable the indeterminate mode.
<a href="#">android:indeterminateBehavior</a>	Defines how the indeterminate mode should behave when the progress reaches max.
<a href="#">android:indeterminateDrawable</a>	Drawable used for the indeterminate mode.
<a href="#">android:indeterminateDuration</a>	Duration of the indeterminate animation.
<a href="#">android:indeterminateOnly</a>	Restricts to ONLY indeterminate mode (state-keeping progress mode will not work).
<a href="#">android:interpolator</a>	
<a href="#">android:max</a>	Defines the maximum value the progress can take.
<a href="#">android:maxHeight</a>	An optional argument to supply a maximum height for this view.
<a href="#">android:maxWidth</a>	An optional argument to supply a maximum width for this view.
<a href="#">android:minHeight</a>	
<a href="#">android:minWidth</a>	
<a href="#">android:progress</a>	Defines the default progress value, between 0 and max.
<a href="#">android:progressDrawable</a>	Drawable used for the progress mode.
<a href="#">android:secondaryProgress</a>	Defines the secondary progress value, between 0 and max.

### See Also

[ProgressBar\\_animationResolution](#)  
[ProgressBar\\_indeterminate](#)  
[ProgressBar\\_indeterminateBehavior](#)  
[ProgressBar\\_indeterminateDrawable](#)

[ProgressBar\\_ineterminateDuration](#)  
[ProgressBar\\_ineterminateOnly](#)  
[ProgressBar\\_interpolator](#)  
[ProgressBar\\_max](#)  
[ProgressBar\\_maxHeight](#)  
[ProgressBar\\_maxWidth](#)  
[ProgressBar\\_minHeight](#)  
[ProgressBar\\_minWidth](#)  
[ProgressBar\\_progress](#)  
[ProgressBar\\_progressDrawable](#)  
[ProgressBar\\_secondaryProgress](#)

## public static final int[] **PropertyAnimator**

Attributes that can be used with a PropertyAnimator.

Includes the following attributes:

Attribute	Description
<a href="#">android:propertyName</a>	Name of the property being animated.

### See Also

[PropertyAnimator.propertyName](#)

## public static final int[] **QuickContactBadge**

Attributes that can be used with a QuickContactBadge.

Includes the following attributes:

### Attribute Description

## public static final int[] **RadioGroup**

Attributes that can be used with a RadioGroup.

Includes the following attributes:

Attribute	Description
<a href="#">android:checkedButton</a>	The id of the child radio button that should be checked by default within this radio group.
<a href="#">android:orientation</a>	Should the radio group be a column or a row? Use "horizontal" for a row, "vertical" for a column.

### See Also

[RadioGroup.checkedButton](#)

[RadioGroup.orientation](#)

## public static final int[] **RatingBar**

Attributes that can be used with a RatingBar.

Includes the following attributes:

Attribute	Description
<a href="#">android:isIndicator</a>	Whether this rating bar is an indicator (and non-changeable by the user).
<a href="#">android:numStars</a>	The number of stars (or rating items) to show.
<a href="#">android:rating</a>	The rating to set by default.
<a href="#">android:stepSize</a>	The step size of the rating.

### See Also

[RatingBar.isIndicator](#)

[RatingBar.numStars](#)

[RatingBar.rating](#)

[RatingBar.stepSize](#)

## public static final int[] **RecognitionService**

Use `recognition-service` as the root tag of the XML resource that describes a [RecognitionService](#) ([/reference/android/speech/RecognitionService.html](#)), which is reference from its [SERVICE\\_META\\_DATA](#) ([/reference/android/speech/RecognitionService.html#SERVICE\\_META\\_DATA](#)) meta-data entry. Described here are the attributes that can be included in that tag.

Includes the following attributes:

Attribute	Description
<code>android:settingsActivity</code>	Component name of an activity that allows the user to modify the settings for this service.

**See Also**

[RecognitionService.settingsActivity](#)

### public static final int[] **RelativeLayout**

Attributes that can be used with a RelativeLayout.

Includes the following attributes:

Attribute	Description
<code>android:gravity</code>	Specifies how an object should position its content, on both the X and Y axes, within its own bounds.
<code>android:ignoreGravity</code>	Indicates what view should not be affected by gravity.

**See Also**

[RelativeLayout\\_gravity](#)  
[RelativeLayout\\_ignoreGravity](#)

### public static final int[] **RelativeLayout\_Layout**

Attributes that can be used with a RelativeLayout\_Layout.

Includes the following attributes:

Attribute	Description
<code>android:layout_above</code>	Positions the bottom edge of this view above the given anchor view ID.
<code>android:layout_alignBaseline</code>	Positions the baseline of this view on the baseline of the given anchor view ID.
<code>android:layout_alignBottom</code>	Makes the bottom edge of this view match the bottom edge of the given anchor view ID.
<code>android:layout_alignEnd</code>	Makes the end edge of this view match the end edge of the given anchor view ID.
<code>android:layout_alignLeft</code>	Makes the left edge of this view match the left edge of the given anchor view ID.
<code>android:layout_alignParentBottom</code>	If true, makes the bottom edge of this view match the bottom edge of the parent.
<code>android:layout_alignParentEnd</code>	If true, makes the end edge of this view match the end edge of the parent.
<code>android:layout_alignParentLeft</code>	If true, makes the left edge of this view match the left edge of the parent.
<code>android:layout_alignParentRight</code>	If true, makes the right edge of this view match the right edge of the parent.
<code>android:layout_alignParentStart</code>	If true, makes the start edge of this view match the start edge of the parent.
<code>android:layout_alignParentTop</code>	If true, makes the top edge of this view match the top edge of the parent.
<code>android:layout_alignRight</code>	Makes the right edge of this view match the right edge of the given anchor view ID.
<code>android:layout_alignStart</code>	Makes the start edge of this view match the start edge of the given anchor view ID.
<code>android:layout_alignTop</code>	Makes the top edge of this view match the top edge of the given anchor view ID.
<code>android:layout_alignWithParentIfMissing</code>	If set to true, the parent will be used as the anchor when the anchor cannot be found for layout_toLeftOf, layout_toRightOf, etc.
<code>android:layout_below</code>	Positions the top edge of this view below the given anchor view ID.
<code>android:layout_centerHorizontal</code>	If true, centers this child horizontally within its parent.
<code>android:layout_centerInParent</code>	If true, centers this child horizontally and vertically within its parent.
<code>android:layout_centerVertical</code>	If true, centers this child vertically within its parent.
<code>android:layout_toEndOf</code>	Positions the start edge of this view to the end of the given anchor view ID.
<code>android:layout_toLeftOf</code>	Positions the right edge of this view to the left of the given anchor view ID.

<a href="#">android:layout_toRightOf</a>	Positions the left edge of this view to the right of the given anchor view ID.
<a href="#">android:layout_toStartOf</a>	Positions the end edge of this view to the start of the given anchor view ID.

#### See Also

[RelativeLayout\\_Layout\\_layout\\_above](#)  
[RelativeLayout\\_Layout\\_layout\\_alignBaseline](#)  
[RelativeLayout\\_Layout\\_layout\\_alignBottom](#)  
[RelativeLayout\\_Layout\\_layout\\_alignEnd](#)  
[RelativeLayout\\_Layout\\_layout\\_alignLeft](#)  
[RelativeLayout\\_Layout\\_layout\\_alignParentBottom](#)  
[RelativeLayout\\_Layout\\_layout\\_alignParentEnd](#)  
[RelativeLayout\\_Layout\\_layout\\_alignParentLeft](#)  
[RelativeLayout\\_Layout\\_layout\\_alignParentRight](#)  
[RelativeLayout\\_Layout\\_layout\\_alignParentStart](#)  
[RelativeLayout\\_Layout\\_layout\\_alignParentTop](#)  
[RelativeLayout\\_Layout\\_layout\\_alignRight](#)  
[RelativeLayout\\_Layout\\_layout\\_alignStart](#)  
[RelativeLayout\\_Layout\\_layout\\_alignTop](#)  
[RelativeLayout\\_Layout\\_layout\\_alignWithParentIfMissing](#)  
[RelativeLayout\\_Layout\\_layout\\_below](#)  
[RelativeLayout\\_Layout\\_layout\\_centerHorizontal](#)  
[RelativeLayout\\_Layout\\_layout\\_centerInParent](#)  
[RelativeLayout\\_Layout\\_layout\\_centerVertical](#)  
[RelativeLayout\\_Layout\\_layout\\_toEndOf](#)  
[RelativeLayout\\_Layout\\_layout\\_toLeftOf](#)  
[RelativeLayout\\_Layout\\_layout\\_toRightOf](#)  
[RelativeLayout\\_Layout\\_layout\\_toStartOf](#)

## public static final int[] RingtonePreference

Base attributes available to RingtonePreference.

Includes the following attributes:

Attribute	Description
<a href="#">android:ringtoneType</a>	Which ringtone type(s) to show in the picker.
<a href="#">android:showDefault</a>	Whether to show an item for a default sound.
<a href="#">android:showSilent</a>	Whether to show an item for 'Silent'.

#### See Also

[RingtonePreference\\_ringtoneType](#)  
[RingtonePreference\\_showDefault](#)  
[RingtonePreference\\_showSilent](#)

## public static final int[] RotateAnimation

Attributes that can be used with a RotateAnimation.

Includes the following attributes:

Attribute	Description
<a href="#">android:fromDegrees</a>	
<a href="#">android:pivotX</a>	
<a href="#">android:pivotY</a>	
<a href="#">android:toDegrees</a>	

#### See Also

[RotateAnimation\\_fromDegrees](#)  
[RotateAnimation\\_pivotX](#)  
[RotateAnimation\\_pivotY](#)  
[RotateAnimation\\_toDegrees](#)

## public static final int[] RotateDrawable

Drawable used to rotate another drawable.

Includes the following attributes:

Attribute	Description
<a href="#">android:drawable</a>	Reference to a drawable resource to use for the frame.
<a href="#">android:fromDegrees</a>	

[android:pivotX](#)  
[android:pivotY](#)  
[android:toDegrees](#)  
[android:visible](#) Provides initial visibility state of the drawable; the default value is false.

**See Also**

[RotateDrawable\\_drawable](#)  
[RotateDrawable\\_fromDegrees](#)  
[RotateDrawable\\_pivotX](#)  
[RotateDrawable\\_pivotY](#)  
[RotateDrawable\\_toDegrees](#)  
[RotateDrawable\\_visible](#)

**public static final int[] ScaleAnimation**

Attributes that can be used with a ScaleAnimation.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:fromXScale</u></a>	
<a href="#"><u>android:fromYScale</u></a>	
<a href="#"><u>android:pivotX</u></a>	
<a href="#"><u>android:pivotY</u></a>	
<a href="#"><u>android:toXScale</u></a>	
<a href="#"><u>android:toYScale</u></a>	

**See Also**

[ScaleAnimation\\_fromXScale](#)  
[ScaleAnimation\\_fromYScale](#)  
[ScaleAnimation\\_pivotX](#)  
[ScaleAnimation\\_pivotY](#)  
[ScaleAnimation\\_toXScale](#)  
[ScaleAnimation\\_toYScale](#)

**public static final int[] ScaleDrawable**

Attributes that can be used with a ScaleDrawable.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:drawable</u></a>	Reference to a drawable resource to draw with the specified scale.
<a href="#"><u>android:scaleGravity</u></a>	Specifies where the drawable is positioned after scaling.
<a href="#"><u>android:scaleHeight</u></a>	Scale height, expressed as a percentage of the drawable's bound.
<a href="#"><u>android:scaleWidth</u></a>	Scale width, expressed as a percentage of the drawable's bound.
<a href="#"><u>android:useIntrinsicSizeAsMinimum</u></a>	Use the drawable's intrinsic width and height as minimum size values.

**See Also**

[ScaleDrawable\\_drawable](#)  
[ScaleDrawable\\_scaleGravity](#)  
[ScaleDrawable\\_scaleHeight](#)  
[ScaleDrawable\\_scaleWidth](#)  
[ScaleDrawable\\_useIntrinsicSizeAsMinimum](#)

**public static final int[] ScrollView**

Attributes that can be used with a ScrollView.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:fillViewport</u></a>	Defines whether the scrollview should stretch its content to fill the viewport.

**See Also**

[ScrollView\\_fillViewport](#)

**public static final int[] SearchView**

Attributes that can be used with a SearchView.

Includes the following attributes:

Attribute	Description
<a href="#">android:iconifiedByDefault</a>	The default state of the SearchView.
<a href="#">android:imeOptions</a>	The IME options to set on the query text field.
<a href="#">android:inputType</a>	The input type to set on the query text field.
<a href="#">android:maxWidth</a>	An optional maximum width of the SearchView.
<a href="#">android:queryHint</a>	An optional query hint string to be displayed in the empty query field.

**See Also**

[SearchView.iconifiedByDefault](#)  
[SearchView.imeOptions](#)  
[SearchView.inputType](#)  
[SearchView.maxWidth](#)  
[SearchView.queryHint](#)

**public static final int[] Searchable**

Searchable activities and applications must provide search configuration information in an XML file, typically called searchable.xml. This file is referenced in your manifest. For a more in-depth discussion of search configuration, please refer to [SearchManager](#) ([/reference/android/app/SearchManager.html](#)).

Includes the following attributes:

Attribute	Description
<a href="#">android:autoUrlDetect</a>	If provided and true, URLs entered in the search dialog while searching within this activity would be detected and treated as URLs (show a 'go' button in the keyboard and invoke the browser directly when user launches the URL instead of passing the URL to the activity).
<a href="#">android:hint</a>	If supplied, this string will be displayed as a hint to the user.
<a href="#">android:icon</a>	This is deprecated.
<a href="#">android:imeOptions</a>	Additional features you can enable in an IME associated with an editor to improve the integration with your application.
<a href="#">android:includeInGlobalSearch</a>	If provided and true, this searchable activity will be included in any global lists of search targets.
<a href="#">android:inputType</a>	The type of data being placed in a text field, used to help an input method decide how to let the user enter text.
<a href="#">android:label</a>	This is the user-displayed name of the searchable activity.
<a href="#">android:queryAfterZeroResults</a>	If provided and true, this searchable activity will be invoked for all queries in a particular session.
<a href="#">android:searchButtonText</a>	If supplied, this string will be displayed as the text of the "Search" button.
<a href="#">android:searchMode</a>	Additional features are controlled by mode bits in this field.
<a href="#">android:searchSettingsDescription</a>	If provided, this string will be used to describe the searchable item in the searchable items settings within system search settings.
<a href="#">android:searchSuggestAuthority</a>	If provided, this is the trigger indicating that the searchable activity provides suggestions as well.
<a href="#">android:searchSuggestIntentAction</a>	If provided, and not overridden by an action in the selected suggestion, this string will be placed in the action field of the Intent when the user clicks a suggestion.
<a href="#">android:searchSuggestIntentData</a>	If provided, and not overridden by an action in the selected suggestion, this string will be placed in the data field of the Intent when the user clicks a suggestion.
<a href="#">android:searchSuggestPath</a>	If provided, this will be inserted in the suggestions query Uri, after the authority you have provide but before the standard suggestions path.
<a href="#">android:searchSuggestSelection</a>	If provided, suggestion queries will be passed into your query function as the selection parameter.
<a href="#">android:searchSuggestThreshold</a>	If provided, this is the minimum number of characters needed to trigger search suggestions.
<a href="#">android:voiceLanguage</a>	If provided, this specifies the spoken language to be expected, and that it will be different than the one set in the getDefault().
<a href="#">android:voiceLanguageModel</a>	If provided, this specifies the language model that should be used by the voice recognition system.

<a href="#">android:voiceMaxResults</a>	If provided, enforces the maximum number of results to return, including the "best" result which will always be provided as the SEARCH intent's primary query.
<a href="#">android:voicePromptText</a>	If provided, this specifies a prompt that will be displayed during voice input.
<a href="#">android:voiceSearchMode</a>	Voice search features are controlled by mode bits in this field.

#### See Also

[Searchable\\_autoUrlDetect](#)  
[Searchable\\_hint](#)  
[Searchable\\_icon](#)  
[Searchable\\_imeOptions](#)  
[Searchable\\_includeInGlobalSearch](#)  
[Searchable\\_inputType](#)  
[Searchable\\_label](#)  
[Searchable\\_queryAfterZeroResults](#)  
[Searchable\\_searchButtonText](#)  
[Searchable\\_searchMode](#)  
[Searchable\\_searchSettingsDescription](#)  
[Searchable\\_searchSuggestAuthority](#)  
[Searchable\\_searchSuggestIntentAction](#)  
[Searchable\\_searchSuggestIntentData](#)  
[Searchable\\_searchSuggestPath](#)  
[Searchable\\_searchSuggestSelection](#)  
[Searchable\\_searchSuggestThreshold](#)  
[Searchable\\_voiceLanguage](#)  
[Searchable\\_voiceLanguageMode1](#)  
[Searchable\\_voiceMaxResults](#)  
[Searchable\\_voicePromptText](#)  
[Searchable\\_voiceSearchMode](#)

## public static final int[] SearchableActionKey

In order to process special action keys during search, you must define them using one or more "ActionKey" elements in your Searchable metadata. For a more in-depth discussion of action code handling, please refer to [SearchManager \(/reference/android/app/SearchManager.html\)](#).

Includes the following attributes:

Attribute	Description
<a href="#">android:keycode</a>	This attribute denotes the action key you wish to respond to.
<a href="#">android:queryActionMsg</a>	If you wish to handle an action key during normal search query entry, you must define an action string here.
<a href="#">android:suggestActionMsg</a>	If you wish to handle an action key while a suggestion is being displayed <i>and selected</i> , there are two ways to handle this.
<a href="#">android:suggestActionMsgColumn</a>	If you wish to handle an action key while a suggestion is being displayed <i>and selected</i> , but you do not wish to enable this action key for every suggestion, then you can use this attribute to control it on a suggestion-by-suggestion basis.

#### See Also

[SearchableActionKey\\_keycode](#)  
[SearchableActionKey\\_queryActionMsg](#)  
[SearchableActionKey\\_suggestActionMsg](#)  
[SearchableActionKey\\_suggestActionMsgColumn](#)

## public static final int[] SeekBar

Attributes that can be used with a SeekBar.

Includes the following attributes:

Attribute	Description
<a href="#">android:thumb</a>	Draws the thumb on a seekbar.
<a href="#">android:thumbOffset</a>	An offset for the thumb that allows it to extend out of the range of the track.

#### See Also

[SeekBar\\_thumb](#)  
[SeekBar\\_thumbOffset](#)

## public static final int[] SelectionModeDrawables

Attributes that can be used with a SelectionModeDrawables.

Includes the following attributes:

Attribute	Description
<a href="#">android:actionModeCopyDrawable</a>	Drawable to use for the Copy action button in Contextual Action Bar
<a href="#">android:actionModeCutDrawable</a>	Drawable to use for the Cut action button in Contextual Action Bar
<a href="#">android:actionModePasteDrawable</a>	Drawable to use for the Paste action button in Contextual Action Bar
<a href="#">android:actionModeSelectAllDrawable</a>	Drawable to use for the Select all action button in Contextual Action Bar

**See Also**

[SelectionModeDrawables\\_actionModeCopyDrawable](#)  
[SelectionModeDrawables\\_actionModeCutDrawable](#)  
[SelectionModeDrawables\\_actionModePasteDrawable](#)  
[SelectionModeDrawables\\_actionModeSelectAllDrawable](#)

**public static final int[] ShapeDrawable**

Drawable used to draw shapes.

Includes the following attributes:

Attribute	Description
<a href="#">android:color</a>	Defines the color of the shape.
<a href="#">android:dither</a>	Enables or disables dithering.
<a href="#">android:height</a>	Defines the height of the shape.
<a href="#">android:width</a>	Defines the width of the shape.

**See Also**

[ShapeDrawable\\_color](#)  
[ShapeDrawable\\_dither](#)  
[ShapeDrawable\\_height](#)  
[ShapeDrawable\\_width](#)

**public static final int[] ShapeDrawablePadding**

Defines the padding of a ShapeDrawable.

Includes the following attributes:

Attribute	Description
<a href="#">android:bottom</a>	Bottom padding.
<a href="#">android:left</a>	Left padding.
<a href="#">android:right</a>	Right padding.
<a href="#">android:top</a>	Top padding.

**See Also**

[ShapeDrawablePadding\\_bottom](#)  
[ShapeDrawablePadding\\_left](#)  
[ShapeDrawablePadding\\_right](#)  
[ShapeDrawablePadding\\_top](#)

**public static final int[] SizeAdaptiveLayout**

Attributes that can be used with a SizeAdaptiveLayout.

**public static final int[] SizeAdaptiveLayout\_Layout**

Attributes that can be used with a SizeAdaptiveLayout\_Layout.

Includes the following attributes:

**Attribute Description**

**public static final int[] SlidingChallengeLayout\_Layout**

Attributes that can be used with a SlidingChallengeLayout\_Layout.

Includes the following attributes:

#### Attribute Description

public static final int[] **SlidingDrawer**

SlidingDrawer specific attributes. These attributes are used to configure a SlidingDrawer from XML.

Includes the following attributes:

Attribute	Description
<a href="#">android:allowSingleTap</a>	Indicates whether the drawer can be opened/closed by a single tap on the handle.
<a href="#">android:animateOnClick</a>	Indicates whether the drawer should be opened/closed with an animation when the user clicks the handle.
<a href="#">android:bottomOffset</a>	Extra offset for the handle at the bottom of the SlidingDrawer.
<a href="#">android:content</a>	Identifier for the child that represents the drawer's content.
<a href="#">android:handle</a>	Identifier for the child that represents the drawer's handle.
<a href="#">android:orientation</a>	Orientation of the SlidingDrawer.
<a href="#">android:topOffset</a>	Extra offset for the handle at the top of the SlidingDrawer.

#### See Also

[SlidingDrawer\\_allowSingleTap](#)  
[SlidingDrawer\\_animateOnClick](#)  
[SlidingDrawer\\_bottomOffset](#)  
[SlidingDrawer\\_content](#)  
[SlidingDrawer\\_handle](#)  
[SlidingDrawer\\_orientation](#)  
[SlidingDrawer\\_topOffset](#)

public static final int[] **SlidingTab**

Attributes that can be used with a SlidingTab.

Includes the following attributes:

Attribute	Description
<a href="#">android:orientation</a>	Use "horizontal" for a row, "vertical" for a column.

#### See Also

[SlidingTab\\_orientation](#)

public static final int[] **SpellChecker**

Use spell-checker as the root tag of the XML resource that describes an [SpellCheckerService](#) service, which is referenced from its [SERVICE\\_META\\_DATA](#) meta-data entry. Described here are the attributes that can be included in that tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:label</a>	The name of the spell checker.
<a href="#">android:settingsActivity</a>	Component name of an activity that allows the user to modify the settings for this service.

#### See Also

[SpellChecker\\_label](#)  
[SpellChecker\\_settingsActivity](#)

public static final int[] **SpellChecker\_Subtype**

This is the subtype of the spell checker. Subtype can describe locales (e.g. en\_US, fr\_FR...)

Includes the following attributes:

Attribute	Description
<a href="#">android:label</a>	The name of the subtype.
<a href="#">android:subtypeExtraValue</a>	The extra value of the subtype.
<a href="#">android:subtypeLocale</a>	The locale of the subtype.

#### See Also

[SpellChecker\\_Subtype\\_label](#)  
[SpellChecker\\_Subtype\\_subtypeExtraValue](#)

## [SpellChecker\\_Subtype\\_subtypeLocale](#)

### **public static final int[] Spinner**

Attributes that can be used with a Spinner.

Includes the following attributes:

Attribute	Description
<a href="#">android:dropDownHorizontalOffset</a>	Horizontal offset from the spinner widget for positioning the dropdown in spinnerMode="dropdown".
<a href="#">android:dropDownSelector</a>	List selector to use for spinnerMode="dropdown" display.
<a href="#">android:dropDownVerticalOffset</a>	Vertical offset from the spinner widget for positioning the dropdown in spinnerMode="dropdown".
<a href="#">android:dropDownWidth</a>	Width of the dropdown in spinnerMode="dropdown".
<a href="#">android:gravity</a>	Gravity setting for positioning the currently selected item.
<a href="#">android:popupBackground</a>	Background drawable to use for the dropdown in spinnerMode="dropdown".
<a href="#">android:prompt</a>	The prompt to display when the spinner's dialog is shown.
<a href="#">android:spinnerMode</a>	Display mode for spinner options.

#### See Also

[Spinner\\_dropDownHorizontalOffset](#)  
[Spinner\\_dropDownSelector](#)  
[Spinner\\_dropDownVerticalOffset](#)  
[Spinner\\_dropDownWidth](#)  
[Spinner\\_gravity](#)  
[Spinner\\_popupBackground](#)  
[Spinner\\_prompt](#)  
[Spinner\\_spinnerMode](#)

### **public static final int[] StackView**

Attributes that can be used with a StackView.

Includes the following attributes:

#### Attribute Description

### **public static final int[] StateListDrawable**

Drawable used to render several states. Each state is represented by a child drawable.

Includes the following attributes:

Attribute	Description
<a href="#">android:constantSize</a>	If true, the drawable's reported internal size will remain constant as the state changes; the size is the maximum of all of the states.
<a href="#">android:dither</a>	Enables or disables dithering of the bitmap if the bitmap does not have the same pixel configuration as the screen (for instance: a ARGB 8888 bitmap with an RGB 565 screen).
<a href="#">android:enterFadeDuration</a>	Amount of time (in milliseconds) to fade in a new state drawable.
<a href="#">android:exitFadeDuration</a>	Amount of time (in milliseconds) to fade out an old state drawable.
<a href="#">android:variablePadding</a>	If true, allows the drawable's padding to change based on the current state that is selected.
<a href="#">android:visible</a>	Indicates whether the drawable should be initially visible.

#### See Also

[StateListDrawable\\_constantSize](#)  
[StateListDrawable\\_dither](#)  
[StateListDrawable\\_enterFadeDuration](#)  
[StateListDrawable\\_exitFadeDuration](#)  
[StateListDrawable\\_variablePadding](#)  
[StateListDrawable\\_visible](#)

### **public static final int[] Storage**

Attributes that can be used with a Storage.

Includes the following attributes:

#### Attribute Description

## **public static final int[] SuggestionSpan**

Attributes that can be used with a SuggestionSpan.

Includes the following attributes:

### **Attribute Description**

## **public static final int[] Switch**

Attributes that can be used with a Switch.

Includes the following attributes:

<b>Attribute</b>	<b>Description</b>
<a href="#"><u>android:switchMinWidth</u></a>	Minimum width for the switch component
<a href="#"><u>android:switchPadding</u></a>	Minimum space between the switch and caption text
<a href="#"><u>android:switchTextAppearance</u></a>	TextAppearance style for text displayed on the switch thumb.
<a href="#"><u>android:textOff</u></a>	Text to use when the switch is in the unchecked/"off" state.
<a href="#"><u>android:textOn</u></a>	Text to use when the switch is in the checked/"on" state.
<a href="#"><u>android:thumb</u></a>	Drawable to use as the "thumb" that switches back and forth.
<a href="#"><u>android:thumbTextPadding</u></a>	Amount of padding on either side of text within the switch thumb.
<a href="#"><u>android:track</u></a>	Drawable to use as the "track" that the switch thumb slides within.

### **See Also**

[Switch\\_switchMinWidth](#)  
[Switch\\_switchPadding](#)  
[Switch\\_switchTextAppearance](#)  
[Switch\\_textOff](#)  
[Switch\\_textOn](#)  
[Switch\\_thumb](#)  
[Switch\\_thumbTextPadding](#)  
[Switch\\_track](#)

## **public static final int[] SwitchPreference**

Attributes that can be used with a SwitchPreference.

Includes the following attributes:

<b>Attribute</b>	<b>Description</b>
<a href="#"><u>android:disableDependentsState</u></a>	The state (true for on, or false for off) that causes dependents to be disabled.
<a href="#"><u>android:summaryOff</u></a>	The summary for the Preference in a PreferenceActivity screen when the SwitchPreference is unchecked.
<a href="#"><u>android:summaryOn</u></a>	The summary for the Preference in a PreferenceActivity screen when the SwitchPreference is checked.
<a href="#"><u>android:switchTextOff</u></a>	The text used on the switch itself when in the "off" state.
<a href="#"><u>android:switchTextOn</u></a>	The text used on the switch itself when in the "on" state.

### **See Also**

[SwitchPreference\\_disableDependentsState](#)  
[SwitchPreference\\_summaryOff](#)  
[SwitchPreference\\_summaryOn](#)  
[SwitchPreference\\_switchTextOff](#)  
[SwitchPreference\\_switchTextOn](#)

## **public static final int[] SyncAdapter**

Use account-authenticator as the root tag of the XML resource that describes an account authenticator.

Includes the following attributes:

<b>Attribute</b>	<b>Description</b>
<a href="#"><u>android:accountType</u></a>	The account type this authenticator handles.
<a href="#"><u>android:allowParallelSyncs</u></a>	Set to true to tell the SyncManager that this SyncAdapter supports multiple simultaneous syncs for the same account type and authority.
<a href="#"><u>android:contentAuthority</u></a>	the authority of a content provider.
<a href="#"><u>android:isAlwaysSyncable</u></a>	Set to true to tell the SyncManager to automatically call setIsSyncable(

[android:settingsActivity](#) If provided, specifies the action of the settings activity for this SyncAdapter.

[android:supportsUploading](#)

[android:userVisible](#)

**See Also**

[SyncAdapter\\_accountType](#)

[SyncAdapter\\_allowParallelSyncs](#)

[SyncAdapter\\_contentAuthority](#)

[SyncAdapter\\_isAlwaysSyncable](#)

[SyncAdapter\\_settingsActivity](#)

[SyncAdapter\\_supportsUploading](#)

[SyncAdapter\\_userVisible](#)

**public static final int[] TabWidget**

Attributes that can be used with a TabWidget.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:divider</u></a>	Drawable used to draw the divider between tabs.
<a href="#"><u>android:tabStripEnabled</u></a>	Determines whether the strip under the tab indicators is drawn or not.
<a href="#"><u>android:tabStripLeft</u></a>	Drawable used to draw the left part of the strip underneath the tabs.
<a href="#"><u>android:tabStripRight</u></a>	Drawable used to draw the right part of the strip underneath the tabs.

**See Also**

[TabWidget\\_divider](#)

[TabWidget\\_tabStripEnabled](#)

[TabWidget\\_tabStripLeft](#)

[TabWidget\\_tabStripRight](#)

**public static final int[] TableLayout**

Attributes that can be used with a TableLayout.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:collapseColumns</u></a>	The zero-based index of the columns to collapse.
<a href="#"><u>android:shrinkColumns</u></a>	The zero-based index of the columns to shrink.
<a href="#"><u>android:stretchColumns</u></a>	The zero-based index of the columns to stretch.

**See Also**

[TableLayout\\_collapseColumns](#)

[TableLayout\\_shrinkColumns](#)

[TableLayout\\_stretchColumns](#)

**public static final int[] TableRow**

Attributes that can be used with a TableRow.

**public static final int[] TableRow\_Cell**

Attributes that can be used with a TableRow\_Cell.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:layout_column</u></a>	The index of the column in which this child should be.
<a href="#"><u>android:layout_span</u></a>	Defines how many columns this child should span.

**See Also**

[TableRow\\_Cell\\_layout\\_column](#)

[TableRow\\_Cell\\_layout\\_span](#)

**public static final int[] TextAppearance**

Attributes that can be used with a TextAppearance.

Includes the following attributes:

Attribute	Description
-----------	-------------

<a href="#">android:fontFamily</a>	Font family (named by string) for the text.
<a href="#">android:textAllCaps</a>	Present the text in ALL CAPS.
<a href="#">android:textColor</a>	Text color.
<a href="#">android:textColorHighlight</a>	Color of the text selection highlight.
<a href="#">android:textColorHint</a>	Color of the hint text.
<a href="#">android:textColorLink</a>	Color of the links.
<a href="#">android:textSize</a>	Size of the text.
<a href="#">android:textStyle</a>	Style (bold, italic, bolditalic) for the text.
<a href="#">android:typeface</a>	Typeface (normal, sans, serif, monospace) for the text.

#### See Also

[TextAppearance\\_fontFamily](#)  
[TextAppearance\\_textAllCaps](#)  
[TextAppearance\\_textColor](#)  
[TextAppearance\\_textColorHighlight](#)  
[TextAppearance\\_textColorHint](#)  
[TextAppearance\\_textColorLink](#)  
[TextAppearance\\_textSize](#)  
[TextAppearance\\_textStyle](#)  
[TextAppearance\\_typeface](#)

## public static final int[] **TextClock**

Attributes that can be used with a TextClock.

Includes the following attributes:

Attribute	Description
<a href="#">android:format12Hour</a>	Specifies the formatting pattern used to show the time and/or date in 12-hour mode.
<a href="#">android:format24Hour</a>	Specifies the formatting pattern used to show the time and/or date in 24-hour mode.
<a href="#">android:timeZone</a>	Specifies the time zone to use.

#### See Also

[TextClock\\_format12Hour](#)  
[TextClock\\_format24Hour](#)  
[TextClock\\_timeZone](#)

## public static final int[] **TextSwitcher**

Attributes that can be used with a TextSwitcher.

## public static final int[] **TextToSpeechEngine**

Use `tts-engine` as the root tag of the XML resource that describes a text to speech engine implemented as a subclass of [TextToSpeechService](#) ([/reference/android/speech/tts/TextToSpeechService.html](#)). The XML resource must be referenced from its `SERVICE META DATA` ([/reference/android/speech/tts/TextToSpeech.Engine.html#SERVICE\\_META\\_DATA](#)) meta-data entry.

Includes the following attributes:

Attribute	Description
<a href="#">android:settingsActivity</a>	Component name of an activity that allows the user to modify the settings for this service.

#### See Also

[TextToSpeechEngine\\_settingsActivity](#)

## public static final int[] **TextView**

Attributes that can be used with a TextView.

Includes the following attributes:

Attribute	Description
<a href="#">android:autoLink</a>	Controls whether links such as urls and email addresses are automatically found and converted to clickable links.
<a href="#">android:autoText</a>	If set, specifies that this TextView has a textual input method and automatically corrects some common spelling errors.

<u><a href="#">android:bufferType</a></u>	Determines the minimum type that getText() will return.
<u><a href="#">android:capitalize</a></u>	If set, specifies that this TextView has a textual input method and should automatically capitalize what the user types.
<u><a href="#">android:cursorVisible</a></u>	Makes the cursor visible (the default) or invisible.
<u><a href="#">android:digits</a></u>	If set, specifies that this TextView has a numeric input method and that these specific characters are the ones that it will accept.
<u><a href="#">android:drawableBottom</a></u>	The drawable to be drawn below the text.
<u><a href="#">android:drawableEnd</a></u>	The drawable to be drawn to the end of the text.
<u><a href="#">android:drawableLeft</a></u>	The drawable to be drawn to the left of the text.
<u><a href="#">android:drawablePadding</a></u>	The padding between the drawables and the text.
<u><a href="#">android:drawableRight</a></u>	The drawable to be drawn to the right of the text.
<u><a href="#">android:drawableStart</a></u>	The drawable to be drawn to the start of the text.
<u><a href="#">android:drawableTop</a></u>	The drawable to be drawn above the text.
<u><a href="#">android:editable</a></u>	If set, specifies that this TextView has an input method.
<u><a href="#">android:editorExtras</a></u>	Reference to an <u><a href="#">&lt;input-extras&gt;</a></u> XML resource containing additional data to supply to an input method, which is private to the implementation of the input method.
<u><a href="#">android:ellipsize</a></u>	If set, causes words that are longer than the view is wide to be ellipsized instead of broken in the middle.
<u><a href="#">android:ems</a></u>	Makes the TextView be exactly this many ems wide.
<u><a href="#">android:enabled</a></u>	Specifies whether the widget is enabled.
<u><a href="#">android:fontFamily</a></u>	Font family (named by string) for the text.
<u><a href="#">android:freezesText</a></u>	If set, the text view will include its current complete text inside of its frozen icicle in addition to metadata such as the current cursor position.
<u><a href="#">android:gravity</a></u>	Specifies how to align the text by the view's x-and/or y-axis when the text is smaller than the view.
<u><a href="#">android:height</a></u>	Makes the TextView be exactly this many pixels tall.
<u><a href="#">android:hint</a></u>	Hint text to display when the text is empty.
<u><a href="#">android:imeActionId</a></u>	Supply a value for <u><a href="#">EditorInfo.actionId</a></u> used when an input method is connected to the text view.
<u><a href="#">android:imeActionLabel</a></u>	Supply a value for <u><a href="#">EditorInfo.actionLabel</a></u> used when an input method is connected to the text view.
<u><a href="#">android:imeOptions</a></u>	Additional features you can enable in an IME associated with an editor to improve the integration with your application.
<u><a href="#">android:includeFontPadding</a></u>	Leave enough room for ascenders and descenders instead of using the font ascent and descent strictly.
<u><a href="#">android:inputMethod</a></u>	If set, specifies that this TextView should use the specified input method (specified by fully-qualified class name).
<u><a href="#">android:inputType</a></u>	The type of data being placed in a text field, used to help an input method decide how to let the user enter text.
<u><a href="#">android:lineSpacingExtra</a></u>	Extra spacing between lines of text.
<u><a href="#">android:lineSpacingMultiplier</a></u>	Extra spacing between lines of text, as a multiplier.
<u><a href="#">android:lines</a></u>	Makes the TextView be exactly this many lines tall.
<u><a href="#">android:linksClickable</a></u>	If set to false, keeps the movement method from being set to the link movement method even if autoLink causes links to be found.
<u><a href="#">android:marqueeRepeatLimit</a></u>	The number of times to repeat the marquee animation.
<u><a href="#">android:maxEms</a></u>	Makes the TextView be at most this many ems wide.
<u><a href="#">android:maxHeight</a></u>	Makes the TextView be at most this many pixels tall.
<u><a href="#">android:maxLength</a></u>	Set an input filter to constrain the text length to the specified number.
<u><a href="#">android:maxLines</a></u>	Makes the TextView be at most this many lines tall.
<u><a href="#">android:maxWidth</a></u>	Makes the TextView be at most this many pixels wide.
<u><a href="#">android:minEms</a></u>	Makes the TextView be at least this many ems wide.
<u><a href="#">android:minHeight</a></u>	Makes the TextView be at least this many pixels tall.

<a href="#">android:minLines</a>	Makes the TextView be at least this many lines tall.
<a href="#">android:minWidth</a>	Makes the TextView be at least this many pixels wide.
<a href="#">android:numeric</a>	If set, specifies that this TextView has a numeric input method.
<a href="#">android:password</a>	Whether the characters of the field are displayed as password dots instead of themselves.
<a href="#">android:phoneNumber</a>	If set, specifies that this TextView has a phone number input method.
<a href="#">android:privateImeOptions</a>	An addition content type description to supply to the input method attached to the text view, which is private to the implementation of the input method.
<a href="#">android:scrollHorizontally</a>	Whether the text is allowed to be wider than the view (and therefore can be scrolled horizontally).
<a href="#">android:selectAllOnFocus</a>	If the text is selectable, select it all when the view takes focus.
<a href="#">android:shadowColor</a>	Place a shadow of the specified color behind the text.
<a href="#">android:shadowDx</a>	Horizontal offset of the shadow.
<a href="#">android:shadowDy</a>	Vertical offset of the shadow.
<a href="#">android:shadowRadius</a>	Radius of the shadow.
<a href="#">android:singleLine</a>	Constrains the text to a single horizontally scrolling line instead of letting it wrap onto multiple lines, and advances focus instead of inserting a newline when you press the enter key.
<a href="#">android:text</a>	Text to display.
<a href="#">android:textAllCaps</a>	Present the text in ALL CAPS.
<a href="#">android:textAppearance</a>	Base text color, typeface, size, and style.
<a href="#">android:textColor</a>	Text color.
<a href="#">android:textColorHighlight</a>	Color of the text selection highlight.
<a href="#">android:textColorHint</a>	Color of the hint text.
<a href="#">android:textColorLink</a>	Text color for links.
<a href="#">android:textCursorDrawable</a>	Reference to a drawable that will be drawn under the insertion cursor.
<a href="#">android:textEditNoPasteWindowLayout</a>	Variation of editTextPasteWindowLayout displayed when the clipboard is empty.
<a href="#">android:textEditPasteWindowLayout</a>	The layout of the view that is displayed on top of the cursor to paste inside a EditText field.
<a href="#">android:textEditSideNoPasteWindowLayout</a>	Variation of editTextSidePasteWindowLayout displayed when the clipboard is empty.
<a href="#">android:textEditSidePasteWindowLayout</a>	Used instead of editTextPasteWindowLayout when the window is moved on the side of the insertion cursor because it would be clipped if it were positioned on top.
<a href="#">android:textEditSuggestionItemLayout</a>	Layout of the TextView item that will populate the suggestion popup window.
<a href="#">android:textIsSelectable</a>	Indicates that the content of a non-editable text can be selected.
<a href="#">android:textScaleX</a>	Sets the horizontal scaling factor for the text.
<a href="#">android:textSelectHandle</a>	Reference to a drawable that will be used to display a text selection anchor for positioning the cursor within text.
<a href="#">android:textSelectHandleLeft</a>	Reference to a drawable that will be used to display a text selection anchor on the left side of a selection region.
<a href="#">android:textSelectHandleRight</a>	Reference to a drawable that will be used to display a text selection anchor on the right side of a selection region.
<a href="#">android:textSize</a>	Size of the text.
<a href="#">android:textStyle</a>	Style (bold, italic, bolditalic) for the text.
<a href="#">android:typeface</a>	Typeface (normal, sans, serif, monospace) for the text.
<a href="#">android:width</a>	Makes the TextView be exactly this many pixels wide.

#### See Also

[TextView\\_autoLink](#)  
[TextView\\_autoText](#)

TextView\_bufferType  
TextView\_capitalize  
TextView\_cursorVisible  
TextView\_digits  
TextView\_drawableBottom  
TextView\_drawableEnd  
TextView\_drawableLeft  
TextView\_drawablePadding  
TextView\_drawableRight  
TextView\_drawableStart  
TextView\_drawableTop  
TextView\_editable  
TextView\_editorExtras  
TextView\_ellipsize  
TextView\_em  
TextView\_enabled  
TextView\_fontFamily  
TextView\_freezesText  
TextView\_gravity  
TextView\_height  
TextView\_hint  
TextView imeActionId  
TextView imeActionLabel  
TextView imeOptions  
TextView\_includeFontPadding  
TextView\_inputMethod  
TextView\_inputType  
TextView\_lineSpacingExtra  
TextView\_lineSpacingMultiplier  
TextView\_lines  
TextView\_linksClickable  
TextView\_marqueeRepeatLimit  
TextView\_maxEm  
TextView\_maxHeight  
TextView\_maxLength  
TextView\_maxLines  
TextView\_maxWidth  
TextView\_minEm  
TextView\_minHeight  
TextView\_minLines  
TextView\_minWidth  
TextView\_numeric  
TextView\_password  
TextView\_phoneNumber  
TextView\_privateImeOptions  
TextView\_scrollHorizontally  
TextView\_selectAllOnFocus  
TextView\_shadowColor  
TextView\_shadowDx  
TextView\_shadowDy  
TextView\_shadowRadius  
TextView\_singleLine  
TextView\_text  
TextView\_textAllCaps  
TextView\_textAppearance  
TextView\_textColor  
TextView\_textColorHighlight  
TextView\_textColorHint  
TextView\_textColorLink  
TextView\_textCursorDrawable  
TextView\_textEditNoPasteWindowLayout  
TextView\_textEditPasteWindowLayout  
TextView\_textEditSideNoPasteWindowLayout  
TextView\_textEditSidePasteWindowLayout  
TextView\_textEditSuggestionItemLayout  
TextView\_textIsSelectable  
TextView\_textScaleX  
TextView\_textSelectHandle  
TextView\_textSelectHandleLeft  
TextView\_textSelectHandleRight  
TextView\_textSize  
TextView\_textStyle  
TextView\_typeface

## TextView\_width

### **public static final int[] TextViewAppearance**

Attributes that can be used with a TextViewAppearance.

Includes the following attributes:

Attribute	Description
<a href="#">android:textAppearance</a>	Base text color, typeface, size, and style.

#### **See Also**

[TextViewAppearance\\_textAppearance](#)

### **public static final int[] TextViewMultiLineBackgroundState**

Attributes that can be used with a TextViewMultiLineBackgroundState.

Includes the following attributes:

Attribute	Description
<a href="#">android:state_multiline</a>	State identifier indicating a TextView has a multi-line layout.

#### **See Also**

[TextViewMultiLineBackgroundState\\_state\\_multiline](#)

### **public static final int[] Theme**

These are the standard attributes that make up a complete theme.

Includes the following attributes:

Attribute	Description
<a href="#">android:absListViewStyle</a>	Default AbsListView style.
<a href="#">android:actionBarDivider</a>	Custom divider drawable to use for elements in the action bar.
<a href="#">android:actionBarItemBackground</a>	Custom item state list drawable background for action bar items.
<a href="#">android: actionBarSize</a>	Size of the Action Bar, including the contextual bar used to present Action Modes.
<a href="#">android: actionBarSplitStyle</a>	Reference to a style for the split Action Bar.
<a href="#">android: actionBarStyle</a>	Reference to a style for the Action Bar
<a href="#">android: actionBarTabBarStyle</a>	Default style for tabs within an action bar
<a href="#">android: actionBarTabStyle</a>	Default style for tabs within an action bar
<a href="#">android: actionBarTextTheme</a>	Reference to a theme that should be used to inflate widgets and layouts destined for the action bar.
<a href="#">android: actionBarWidgetTheme</a>	Default action button style.
<a href="#">android: actionBarStyle</a>	Default ActionBar dropdown style.
<a href="#">android: actionDropDownStyle</a>	TextAppearance style that will be applied to text that appears within action menu items.
<a href="#">android: actionMenuTextAppearance</a>	Color for text that appears within action menu items.
<a href="#">android: actionMenuTextColor</a>	Background drawable to use for action mode UI
<a href="#">android: actionModeBackground</a>	Background drawable to use for the close action mode button
<a href="#">android: actionModeCloseButtonStyle</a>	Drawable to use for the Copy action button in Contextual Action Bar
<a href="#">android: actionModeCloseDrawable</a>	Drawable to use for the Cut action button in Contextual Action Bar
<a href="#">android: actionModeCopyDrawable</a>	Drawable to use for the Paste action button in Contextual Action Bar
<a href="#">android: actionModeCutDrawable</a>	Drawable to use for the Select all action button in Contextual Action Bar
<a href="#">android: actionModePasteDrawable</a>	Drawable to use for the Select all action button in Contextual Action Bar
<a href="#">android: actionModeSelectAllDrawable</a>	Background drawable to use for action mode UI in the lower split bar

<code>android:actionModeStyle</code>	
<code>android:actionOverflowButtonStyle</code>	
<code>android:activatedBackgroundIndicator</code>	Drawable used as a background for activated items.
<code>android:alertDialogIcon</code>	Icon drawable to use for alerts
<code>android:alertDialogStyle</code>	
<code>android:alertDialogTheme</code>	Theme to use for alert dialogs spawned from this theme.
<code>android:autoCompleteTextViewStyle</code>	Default AutoCompleteTextView style.
<code>android:backgroundDimAmount</code>	Default background dim amount when a menu, dialog, or something similar pops up.
<code>android:backgroundDimEnabled</code>	Control whether dimming behind the window is enabled.
<code>android:borderlessButtonStyle</code>	Style for buttons without an explicit border, often used in groups.
<code>android:buttonBarButtonStyle</code>	Style for buttons within button bars
<code>android:buttonBarStyle</code>	Style for button bars
<code>android:buttonStyle</code>	Normal Button style.
<code>android:buttonStyleInset</code>	Button style to inset into an EditText.
<code>android:buttonStyleSmall</code>	Small Button style.
<code>android:buttonStyleToggle</code>	ToggleButton style.
<code>android:calendarViewStyle</code>	The CalendarView style.
<code>android:candidatesTextStyleSpans</code>	A styled string, specifying the style to be used for showing inline candidate text when composing with an input method.
<code>android:checkBoxPreferenceStyle</code>	Default style for CheckBoxPreference.
<code>android:checkboxStyle</code>	Default Checkbox style.
<code>android:checkedTextVisualStyle</code>	Default CheckedTextview style.
<code>android:colorActivatedHighlight</code>	Default highlight color for items that are activated.
<code>android:colorBackground</code>	Color that matches (as closely as possible) the window background.
<code>android:colorBackgroundCacheHint</code>	This is a hint for a solid color that can be used for caching rendered views.
<code>android:colorFocusedHighlight</code>	Default highlight color for items that are focused.
<code>android:colorForeground</code>	Default color of foreground imagery.
<code>android:colorForegroundInverse</code>	Default color of foreground imagery on an inverted background.
<code>android:colorLongPressedHighlight</code>	Default highlight color for items that are long-pressed.
<code>android:colorMultiSelectHighlight</code>	Default highlight color for items in multiple selection mode.
<code>android:colorPressedHighlight</code>	Default highlight color for items that are pressed.
<code>android:datePickerStyle</code>	The DatePicker style.
<code>android:detailsElementBackground</code>	Background that can be used behind parts of a UI that provide details on data the user is selecting.
<code>android:dialogPreferenceStyle</code>	Default style for DialogPreference.
<code>android:dialogTheme</code>	Theme to use for dialogs spawned from this theme.
<code>android:disabledAlpha</code>	Default disabled alpha for widgets that set enabled/disabled alpha programmatically.
<code>android:dividerHorizontal</code>	Drawable to use for generic horizontal dividers.
<code>android:dividerVertical</code>	Drawable to use for generic vertical dividers.
<code>android:dropDownHintAppearance</code>	Default style for drop down hints.
<code>android:dropDownItemStyle</code>	Default style for drop down items.
<code>android:dropDownListStyle</code>	Default ListView style for drop downs.
<code>android:dropDownSpinnerStyle</code>	Default dropdown Spinner style.
<code>android:editTextBackground</code>	EditText background drawable.
<code>android:editTextColor</code>	EditText text foreground color.
<code>android:editTextPreferenceStyle</code>	Default style for EditTextPreference.

<code>android:editTextStyle</code>	Default EditText style.
<code>android:expandableListPreferredChildIndicatorLeft</code>	The preferred left bound for an expandable list child's indicator.
<code>android:expandableListPreferredChildIndicatorRight</code>	The preferred right bound for an expandable list child's indicator.
<code>android:expandableListPreferredChildPaddingLeft</code>	The preferred left padding for an expandable list item that is a child.
<code>android:expandableListPreferredItemIndicatorLeft</code>	The preferred left bound for an expandable list item's indicator.
<code>android:expandableListPreferredItemIndicatorRight</code>	The preferred right bound for an expandable list item's indicator.
<code>android:expandableListPreferredItemPaddingLeft</code>	The preferred left padding for an expandable list item (for child-specific layouts, use <code>expandableListPreferredChildPaddingLeft</code> ).
<code>android:expandableListViewStyle</code>	Default ExpandableListView style.
<code>android:expandableListViewWhiteStyle</code>	ExpandableListView with white background.
<code>android:fastScrollOverlayPosition</code>	Position of the fast scroll index overlay window.
<code>android:fastScrollPreviewBackgroundLeft</code>	Drawable to use as the fast scroll index preview window background when shown on the left.
<code>android:fastScrollPreviewBackgroundRight</code>	Drawable to use as the fast scroll index preview window background when shown on the right.
<code>android:fastScrollTextColor</code>	Text color for the fast scroll index overlay.
<code>android:fastScrollThumbDrawable</code>	Drawable to use as the fast scroll thumb.
<code>android:fastScrollTrackDrawable</code>	Drawable to use as the track for the fast scroll thumb.
<code>android:galleryItemBackground</code>	The preferred background for gallery items.
<code>android:galleryStyle</code>	Default Gallery style.
<code>android:gridViewStyle</code>	Default GridView style.
<code>android:homeAsUpIndicator</code>	Specifies a drawable to use for the 'home as up' indicator.
<code>android:horizontalScrollViewStyle</code>	Default HorizontalScrollView style.
<code>android:imageButtonStyle</code>	The style resource to use for an ImageButton.
<code>android:imageWellStyle</code>	The style resource to use for an ImageButton that is an image well.
<code>android:listChoiceBackgroundIndicator</code>	Drawable used as a background for selected list items.
<code>android:listChoiceIndicatorMultiple</code>	Drawable to use for multiple choice indicators.
<code>android:listChoiceIndicatorSingle</code>	Drawable to use for single choice indicators.
<code>android:listDivider</code>	The drawable for the list divider.
<code>android:listDividerAlertDialog</code>	The list divider used in alert dialogs.
<code>android:listPopupWindowStyle</code>	Default ListPopupWindow style.
<code>android:listPreferredItemHeight</code>	The preferred list item height.
<code>android:listPreferredItemHeightLarge</code>	A larger, more robust list item height.
<code>android:listPreferredItemHeightSmall</code>	A smaller, sleeker list item height.
<code>android:listPreferredItemPaddingEnd</code>	The preferred padding along the end edge of list items.
<code>android:listPreferredItemPaddingLeft</code>	The preferred padding along the left edge of list items.
<code>android:listPreferredItemPaddingRight</code>	The preferred padding along the right edge of list items.
<code>android:listPreferredItemPaddingStart</code>	The preferred padding along the start edge of list items.
<code>android:listSeparatorTextViewStyle</code>	TextView style for list separators.
<code>android:listViewStyle</code>	Default ListView style.
<code>android:listViewWhiteStyle</code>	ListView with white background.
<code>android:mapViewStyle</code>	Default MapView style.
<code>android:mediaRouteButtonStyle</code>	Default style for the MediaRouteButton widget.

<code>android:panelBackground</code>	The background of a panel when it is inset from the left and right edges of the screen.
<code>android:panelColorBackground</code>	Color that matches (as closely as possible) the panel background.
<code>android:panelColorForeground</code>	Default color of foreground panel imagery.
<code>android:panelFullBackground</code>	The background of a panel when it extends to the left and right edges of the screen.
<code>android:panelTextAppearance</code>	Default appearance of panel text.
<code>android:popupMenuStyle</code>	Default PopupMenu style.
<code>android:popupWindowStyle</code>	Default PopupWindow style.
<code>android:preferenceCategoryStyle</code>	Default style for PreferenceCategory.
<code>android:preferenceInformationStyle</code>	Default style for informational Preference.
<code>android:preferenceLayoutChild</code>	The preference layout that has the child/tabbed effect.
<code>android:preferenceScreenStyle</code>	Default style for PreferenceScreen.
<code>android:preferenceStyle</code>	Default style for Preference.
<code>android:presentationTheme</code>	Theme to use for presentations spawned from this theme.
<code>android:progressBarStyle</code>	Default ProgressBar style.
<code>android:progressBarStyleHorizontal</code>	Horizontal ProgressBar style.
<code>android:progressBarStyleInverse</code>	Inverse ProgressBar style.
<code>android:progressBarStyleLarge</code>	Large ProgressBar style.
<code>android:progressBarStyleLargeInverse</code>	Large inverse ProgressBar style.
<code>android:progressBarStyleSmall</code>	Small ProgressBar style.
<code>android:progressBarStyleSmallInverse</code>	Small inverse ProgressBar style.
<code>android:progressBarStyleSmallTitle</code>	Small ProgressBar in title style.
<code>android:quickContactBadgeStyleLargeWindowLarge</code>	Default quickcontact badge style with large quickcontact window.
<code>android:quickContactBadgeStyleMediumWindowMedium</code>	Default quickcontact badge style with medium quickcontact window.
<code>android:quickContactBadgeStyleSmallWindowSmall</code>	Default quickcontact badge style with small quickcontact window.
<code>android:quickContactBadgeStyleWindowLarge</code>	Default quickcontact badge style with large quickcontact window.
<code>android:quickContactBadgeStyleWindowMedium</code>	Default quickcontact badge style with medium quickcontact window.
<code>android:quickContactBadgeStyleWindowSmall</code>	Default quickcontact badge style with small quickcontact window.
<code>android:radioButtonStyle</code>	Default RadioButton style.
<code>android:ratingBarStyle</code>	Default RatingBar style.
<code>android:ratingBarStyleIndicator</code>	Indicator RatingBar style.
<code>android:ratingBarStyleSmall</code>	Small indicator RatingBar style.
<code>android:ringtonePreferenceStyle</code>	Default style for RingtonePreference.
<code>android:scrollViewStyle</code>	Default ScrollView style.
<code>android:seekBarStyle</code>	Default SeekBar style.
<code>android:segmentedButtonStyle</code>	Style for segmented buttons - a container that houses several buttons with the appearance of a singel button broken into segments.
<code>android:selectableItemBackground</code>	Background drawable for standalone items that need focus/pressed states.
<code>android:spinnerDropDownItemStyle</code>	Default style for spinner drop down items.
<code>android:spinnerItemStyle</code>	Default spinner item style.
<code>android:spinnerStyle</code>	Default Spinner style.
<code>android:starStyle</code>	Default Star style.
<code>android:switchPreferenceStyle</code>	Default style for switch preferences.
<code>android:tabWidgetStyle</code>	Default TabWidget style.
<code>android:textAppearance</code>	Default appearance of text: color, typeface, size, and style.
<code>android:textAppearanceButton</code>	Text color, typeface, size, and style for the text inside of a button.
<code>android:textAppearanceInverse</code>	Default appearance of text against an inverted background: color, typeface, size, and style.
	Text color, typeface, size, and style for

<code>android:textAppearanceLarge</code>	"large" text.
<code>android:textAppearanceLargeInverse</code>	Text color, typeface, size, and style for "large" inverse text.
<code>android:textAppearanceLargePopupMenu</code>	Text color, typeface, size, and style for the text inside of a popup menu.
<code>android:textAppearanceListItem</code>	The preferred TextAppearance for the primary text of list items.
<code>android:textAppearanceListItemSmall</code>	The preferred TextAppearance for the primary text of small list items.
<code>android:textAppearanceMedium</code>	Text color, typeface, size, and style for "medium" text.
<code>android:textAppearanceMediumInverse</code>	Text color, typeface, size, and style for "medium" inverse text.
<code>android:textAppearanceSearchResultSubtitle</code>	Text color, typeface, size, and style for system search result subtitle.
<code>android:textAppearanceSearchResultTitle</code>	Text color, typeface, size, and style for system search result title.
<code>android:textAppearanceSmall</code>	Text color, typeface, size, and style for "small" text.
<code>android:textAppearanceSmallInverse</code>	Text color, typeface, size, and style for "small" inverse text.
<code>android:textAppearanceSmallPopupMenu</code>	Text color, typeface, size, and style for small text inside of a popup menu.
<code>android:textCheckMark</code>	Drawable to use for check marks.
<code>android:textCheckMarkInverse</code>	
<code>android:textColorAlertDialogListItem</code>	Color of list item text in alert dialogs.
<code>android:textColorHighlightInverse</code>	Color of highlighted text, when used in a light theme.
<code>android:textColorHintInverse</code>	Inverse hint text color.
<code>android:textColorLinkInverse</code>	Color of link text (URLs), when used in a light theme.
<code>android:textColorPrimary</code>	The most prominent text color.
<code>android:textColorPrimaryDisableOnly</code>	Bright text color.
<code>android:textColorPrimaryInverse</code>	Primary inverse text color, useful for inverted backgrounds.
<code>android:textColorPrimaryInverseDisableOnly</code>	Bright inverse text color.
<code>android:textColorPrimaryInverseNoDisable</code>	Bright inverse text color.
<code>android:textColorPrimaryNoDisable</code>	Bright text color.
<code>android:textColorSecondary</code>	Secondary text color.
<code>android:textColorSecondaryInverse</code>	Secondary inverse text color, useful for inverted backgrounds.
<code>android:textColorSecondaryInverseNoDisable</code>	Dim inverse text color.
<code>android:textColorSecondaryNoDisable</code>	Dim text color.
<code>android:textColorTertiary</code>	Tertiary text color.
<code>android:textColorTertiaryInverse</code>	Tertiary inverse text color, useful for inverted backgrounds.
<code>android:textEditNoPasteWindowLayout</code>	Variation of <code>textEditPasteWindowLayout</code> displayed when the clipboard is empty.
<code>android:textEditPasteWindowLayout</code>	The layout of the view that is displayed on top of the cursor to paste inside a <code>EditText</code> field.
<code>android:textEditSideNoPasteWindowLayout</code>	Variation of <code>textEditSidePasteWindowLayout</code> displayed when the clipboard is empty.
<code>android:textEditSidePasteWindowLayout</code>	Used instead of <code>textEditPasteWindowLayout</code> when the window is moved on the side of the insertion cursor because it would be clipped if it were positioned on top.
<code>android:textEditSuggestionItemLayout</code>	Layout of the <code>TextView</code> item that will populate the suggestion popup window.
<code>android:textSelectHandle</code>	Reference to a drawable that will be used to display a text selection anchor for positioning the cursor within text.
<code>android:textSelectHandleLeft</code>	Reference to a drawable that will be used to display a text selection anchor on the left

<a href="#">android:textSelectHandleRight</a>	side of a selection region.
<a href="#">android:textSelectHandleWindowStyle</a>	Reference to a drawable that will be used to display a text selection anchor on the right side of a selection region.
<a href="#">android:textSuggestionsWindowStyle</a>	Reference to a style that will be used for the window containing a list of possible text suggestions in an EditText.
<a href="#">android:textViewStyle</a>	Default TextView style.
<a href="#">android:webTextViewStyle</a>	Default WebTextView style.
<a href="#">android:webViewStyle</a>	Default WebView style.
<a href="#">android:windowActionBar</a>	Flag indicating whether this window should have an Action Bar in place of the usual title bar.
<a href="#">android:windowActionBarOverlay</a>	Flag indicating whether this window's Action Bar should overlay application content.
<a href="#">android:windowActionModeOverlay</a>	Flag indicating whether action modes should overlay window content when there is not reserved space for their UI (such as an Action Bar).
<a href="#">android:windowAnimationStyle</a>	Reference to a style resource holding the set of window animations to use, which can be any of the attributes defined by <a href="#">WindowAnimation</a> .
<a href="#">android:windowBackground</a>	Drawable to use as the overall window background.
<a href="#">android:windowCloseOnTouchOutside</a>	Control whether a container should automatically close itself if the user touches outside of it.
<a href="#">android:windowContentOverlay</a>	This Drawable is overlaid over the foreground of the Window's content area, usually to place a shadow below the title.
<a href="#">android:windowDisablePreview</a>	Flag allowing you to disable the preview animation for a window.
<a href="#">android:windowEnableSplitTouch</a>	Flag indicating that this window should allow touches to be split across other windows that also support split touch.
<a href="#">android:windowFrame</a>	Drawable to use as a frame around the window.
<a href="#">android:windowFullscreen</a>	Flag indicating whether this window should fill the entire screen.
<a href="#">android:windowIsFloating</a>	Flag indicating whether this is a floating window.
<a href="#">android:windowIsTranslucent</a>	Flag indicating whether this is a translucent window.
<a href="#">android:windowNoDisplay</a>	Flag indicating that this window should not be displayed at all.
<a href="#">android:windowNoTitle</a>	Flag indicating whether there should be no title on this window.
<a href="#">android:windowShowWallpaper</a>	Flag indicating that this window's background should be the user's current wallpaper.
<a href="#">android:windowSoftInputMode</a>	Defines the default soft input state that this window would like when it is displayed.
<a href="#">android:windowTitleBackgroundStyle</a>	The style resource to use for a window's title area.
<a href="#">android:windowTitleSize</a>	The style resource to use for a window's title bar height.
<a href="#">android:windowTitleStyle</a>	The style resource to use for a window's title text.
<a href="#">android:yesNoPreferenceStyle</a>	Default style for YesNoPreference.

## See Also

[Theme\\_absListViewStyle](#)  
[Theme\\_actionBarDivider](#)  
[Theme\\_actionBarItemBackground](#)

Theme\_actionBarSize  
Theme\_actionBarSplitStyle  
Theme\_actionBarStyle  
Theme\_actionBarTabBarStyle  
Theme\_actionBarTabStyle  
Theme\_actionBarTabTextStyle  
Theme\_actionBarWidgetTheme  
ThemeActionButtonStyle  
Theme\_actionDropDownStyle  
Theme\_actionMenuTextAppearance  
Theme\_actionMenuTextColor  
Theme\_actionModeBackground  
Theme\_actionModeCloseButtonStyle  
Theme\_actionModeCloseDrawable  
Theme\_actionModeCopyDrawable  
Theme\_actionModeCutDrawable  
Theme\_actionModePasteDrawable  
Theme\_actionModeSelectAllDrawable  
Theme\_actionModeSplitBackground  
Theme\_actionModeStyle  
Theme\_actionOverflowButtonStyle  
Theme\_activatedBackgroundIndicator  
Theme\_alertDialogIcon  
Theme\_alertDialogStyle  
Theme\_alertDialogTheme  
Theme\_autoCompleteTextViewStyle  
Theme\_backgroundDimAmount  
Theme\_backgroundDimEnabled  
Theme\_borderlessButtonStyle  
Theme\_buttonBarButtonStyle  
Theme\_buttonBarStyle  
Theme\_buttonStyle  
Theme\_buttonStyleInset  
Theme\_buttonStyleSmall  
Theme\_buttonStyleToggle  
Theme\_calendarViewStyle  
Theme\_candidatesTextStyleSpans  
Theme\_checkBoxPreferenceStyle  
Theme\_checkboxStyle  
Theme\_checkedTextViewStyle  
Theme\_colorActivatedHighlight  
Theme\_colorBackground  
Theme\_colorBackgroundCacheHint  
Theme\_colorFocusedHighlight  
Theme\_colorForeground  
Theme\_colorForegroundInverse  
Theme\_colorLongPressedHighlight  
Theme\_colorMultiSelectHighlight  
Theme\_colorPressedHighlight  
Theme\_datePickerStyle  
Theme\_detailsElementBackground  
Theme\_dialogPreferenceStyle  
Theme\_dialogTheme  
Theme\_disabledAlpha  
Theme\_dividerHorizontal  
Theme\_dividerVertical  
Theme\_dropDownHintAppearance  
Theme\_dropDownItemStyle  
Theme\_dropDownListStyle  
Theme\_dropDownSpinnerStyle  
Theme\_editTextBackground  
Theme\_editTextColor  
Theme\_editTextPreferenceStyle  
Theme\_editTextStyle  
Theme\_expandableListPreferredChildIndicatorLeft  
Theme\_expandableListPreferredChildIndicatorRight  
Theme\_expandableListPreferredChildPaddingLeft  
Theme\_expandableListPreferredItemIndicatorLeft  
Theme\_expandableListPreferredItemIndicatorRight  
Theme\_expandableListPreferredItemPaddingLeft  
Theme\_expandableListViewStyle  
Theme\_expandableListViewWhiteStyle  
Theme\_fastScrollOverlayPosition

Theme\_fastScrollPreviewBackgroundLeft  
Theme\_fastScrollPreviewBackgroundRight  
Theme\_fastScrollTextColor  
Theme\_fastScrollThumbDrawable  
Theme\_fastScrollTrackDrawable  
Theme\_galleryItemBackground  
Theme\_galleryStyle  
Theme\_gridViewStyle  
Theme\_homeAsUpIndicator  
Theme\_horizontalScrollViewStyle  
Theme\_imageButtonStyle  
Theme\_imageWellStyle  
Theme\_listChoiceBackgroundIndicator  
Theme\_listChoiceIndicatorMultiple  
Theme\_listChoiceIndicatorSingle  
Theme\_listDivider  
Theme\_listDividerAlertDialog  
Theme\_listPopupWindowStyle  
Theme\_listPreferredItemHeight  
Theme\_listPreferredItemHeightLarge  
Theme\_listPreferredItemHeightSmall  
Theme\_listPreferredItemPaddingEnd  
Theme\_listPreferredItemPaddingLeft  
Theme\_listPreferredItemPaddingRight  
Theme\_listPreferredItemPaddingStart  
Theme\_listSeparatorTextViewStyle  
Theme\_listViewStyle  
Theme\_listViewWhiteStyle  
Theme\_mapViewStyle  
Theme\_mediaRouteButtonStyle  
Theme\_panelBackground  
Theme\_panelColorBackground  
Theme\_panelColorForeground  
Theme\_panelFullBackground  
Theme\_panelTextAppearance  
Theme\_popupMenuStyle  
Theme\_popupWindowStyle  
Theme\_preferenceCategoryStyle  
Theme\_preferenceInformationStyle  
Theme\_preferenceLayoutChild  
Theme\_preferencesScreenStyle  
Theme\_preferenceStyle  
Theme\_presentationTheme  
Theme\_progressBarStyle  
Theme\_progressBarStyleHorizontal  
Theme\_progressBarStyleInverse  
Theme\_progressBarStyleLarge  
Theme\_progressBarStyleLargeInverse  
Theme\_progressBarStyleSmall  
Theme\_progressBarStyleSmallInverse  
Theme\_progressBarStyleSmallTitle  
Theme\_quickContactBadgeStyleSmallWindowLarge  
Theme\_quickContactBadgeStyleSmallWindowMedium  
Theme\_quickContactBadgeStyleSmallWindowSmall  
Theme\_quickContactBadgeStyleWindowLarge  
Theme\_quickContactBadgeStyleWindowMedium  
Theme\_quickContactBadgeStyleWindowSmall  
Theme\_radioButtonStyle  
Theme\_ratingBarStyle  
Theme\_ratingBarStyleIndicator  
Theme\_ratingBarStyleSmall  
Theme\_ringtonePreferenceStyle  
Theme\_scrollViewStyle  
Theme\_seekBarStyle  
Theme\_segmentedButtonStyle  
Theme\_selectableItemBackground  
Theme\_spinnerDropDownItemStyle  
Theme\_spinnerItemStyle  
Theme\_spinnerStyle  
Theme\_starStyle  
Theme\_switchPreferenceStyle  
Theme\_tabWidgetStyle  
Theme\_textAppearance

```
Theme_textAppearanceButton
Theme_textAppearanceInverse
Theme_textAppearanceLarge
Theme_textAppearanceLargeInverse
Theme_textAppearanceLargePopupMenu
Theme_textAppearanceListItem
Theme_textAppearanceListItemSmall
Theme_textAppearanceMedium
Theme_textAppearanceMediumInverse
Theme_textAppearanceSearchResultSubtitle
Theme_textAppearanceSearchResultTitle
Theme_textAppearanceSmall
Theme_textAppearanceSmallInverse
Theme_textAppearanceSmallPopupMenu
Theme_textCheckMark
Theme_textCheckMarkInverse
Theme_textColorAlertDialogListItem
Theme_textColorHighlightInverse
Theme_textColorHintInverse
Theme_textColorLinkInverse
Theme_textColorPrimary
Theme_textColorPrimaryDisableOnly
Theme_textColorPrimaryInverse
Theme_textColorPrimaryInverseDisableOnly
Theme_textColorPrimaryInverseNoDisable
Theme_textColorPrimaryNoDisable
Theme_textColorSecondary
Theme_textColorSecondaryInverse
Theme_textColorSecondaryInverseNoDisable
Theme_textColorSecondaryNoDisable
Theme_textColorTertiary
Theme_textColorTertiaryInverse
ThemeTextEditNoPasteWindowLayout
ThemeTextEditPasteWindowLayout
ThemeTextEditSideNoPasteWindowLayout
ThemeTextEditSidePasteWindowLayout
ThemeTextEditSuggestionItemLayout
Theme_textSelectHandle
Theme_textSelectHandleLeft
Theme_textSelectHandleRight
Theme_textSelectHandleWindowSize
Theme_textSuggestionsWindowSize
Theme_textViewStyle
Theme_webTextViewStyle
Theme_webViewStyle
Theme_windowActionBar
Theme_windowActionBarOverlay
Theme_windowActionModeOverlay
Theme_windowAnimationStyle
Theme_windowBackground
Theme_windowCloseOnTouchOutside
Theme_windowContentOverlay
Theme_windowDisablePreview
Theme_windowEnableSplitTouch
Theme_windowFrame
Theme_windowFullscreen
Theme_windowIsFloating
Theme_windowIsTranslucent
Theme_windowNoDisplay
Theme_windowNoTitle
Theme_windowShowWallpaper
Theme_windowSoftInputMode
Theme_windowTitleBackgroundStyle
Theme_windowTitleSize
Theme_windowTitleStyle
Theme_yesNoPreferenceStyle
```

public static final int[] TimePicker

Attributes that can be used with a TimePicker.

Includes the following attributes:

#### Attribute Description

##### public static final int[] **ToggleButton**

Attributes that can be used with a ToggleButton.

Includes the following attributes:

Attribute	Description
<a href="#">android:disabledAlpha</a>	The alpha to apply to the indicator when disabled.
<a href="#">android:textOff</a>	The text for the button when it is not checked.
<a href="#">android:textOn</a>	The text for the button when it is checked.

#### See Also

[ToggleButton\\_disabledAlpha](#)  
[ToggleButton\\_textOff](#)  
[ToggleButton\\_textOn](#)

##### public static final int[] **TranslateAnimation**

Attributes that can be used with a TranslateAnimation.

Includes the following attributes:

Attribute	Description
<a href="#">android:fromXDelta</a>	
<a href="#">android:fromYDelta</a>	
<a href="#">android:toXDelta</a>	
<a href="#">android:toYDelta</a>	

#### See Also

[TranslateAnimation\\_fromXDelta](#)  
[TranslateAnimation\\_fromYDelta](#)  
[TranslateAnimation\\_toXDelta](#)  
[TranslateAnimation\\_toYDelta](#)

##### public static final int[] **TwoLineListIItem**

Attributes that can be used with a TwoLineListIItem.

Includes the following attributes:

Attribute	Description
<a href="#">android:mode</a>	

#### See Also

[TwoLineListIItem\\_mode](#)

##### public static final int[] **VerticalSlider\_Layout**

Attributes that can be used with a VerticalSlider\_Layout.

Includes the following attributes:

Attribute	Description
<a href="#">android:layout_scale</a>	

#### See Also

[VerticalSlider\\_Layout\\_layout\\_scale](#)

##### public static final int[] **View**

Attributes that can be used with [View](#) ([/reference/android/view/View.html](#)) or any of its subclasses.

Also see [ViewGroup\\_Layout](#) ([/reference/android/R.styleable.html#ViewGroup\\_Layout](#)) for attributes that are processed by the view's parent.

Includes the following attributes:

Attribute	Description
<a href="#">android:alpha</a>	alpha property of the view, as a value between 0 (completely transparent) and 1 (completely opaque).
<a href="#">android:background</a>	A drawable to use as the background.

<u><a href="#">android:clickable</a></u>	Defines whether this view reacts to click events.
<u><a href="#">android:contentDescription</a></u>	Defines text that briefly describes content of the view.
<u><a href="#">android:drawingCacheQuality</a></u>	Defines the quality of translucent drawing caches.
<u><a href="#">android:duplicateParentState</a></u>	When this attribute is set to true, the view gets its drawable state (focused, pressed, etc).
<u><a href="#">android:fadeScrollbars</a></u>	Defines whether to fade out scrollbars when they are not in use.
<u><a href="#">android:fadingEdge</a></u>	This attribute is deprecated and will be ignored as of API level 14 ( <a href="#">ICE_CREAM_SANDWICH</a> ).
<u><a href="#">android:fadingEdgeLength</a></u>	Defines the length of the fading edges.
<u><a href="#">android:filterTouchesWhenObscured</a></u>	Specifies whether to filter touches when the view's window is obscured by another visible window.
<u><a href="#">android:fitsSystemWindows</a></u>	Boolean internal attribute to adjust view layout based on system windows such as the status bar.
<u><a href="#">android:focusable</a></u>	Boolean that controls whether a view can take focus.
<u><a href="#">android:focusableInTouchMode</a></u>	Boolean that controls whether a view can take focus while in touch mode.
<u><a href="#">android:hapticFeedbackEnabled</a></u>	Boolean that controls whether a view should have haptic feedback enabled for events such as long presses.
<u><a href="#">android:id</a></u>	Supply an identifier name for this view, to later retrieve it with <a href="#">View.findViewById()</a> or <a href="#">Activity.findViewById()</a> .
<u><a href="#">android:importantForAccessibility</a></u>	Controls how this View is important for accessibility which is if it fires accessibility events and if it is reported to accessibility services that query the screen.
<u><a href="#">android:isScrollContainer</a></u>	Set this if the view will serve as a scrolling container, meaning that it can be resized to shrink its overall window so that there will be space for an input method.
<u><a href="#">android:keepScreenOn</a></u>	Controls whether the view's window should keep the screen on while visible.
<u><a href="#">android:labelFor</a></u>	Specifies the id of a view for which this view serves as a label for accessibility purposes.
<u><a href="#">android:layerType</a></u>	Specifies the type of layer backing this view.
<u><a href="#">android:layoutDirection</a></u>	Defines the direction of layout drawing.
<u><a href="#">android:longClickable</a></u>	Defines whether this view reacts to long click events.
<u><a href="#">android:minHeight</a></u>	Defines the minimum height of the view.
<u><a href="#">android:minWidth</a></u>	Defines the minimum width of the view.
<u><a href="#">android:nextFocusDown</a></u>	Defines the next view to give focus to when the next focus is <a href="#">FOCUS_DOWN</a> If the reference refers to a view that does not exist or is part of a hierarchy that is invisible, a <a href="#">RuntimeException</a> will result when the reference is accessed.
<u><a href="#">android:nextFocusForward</a></u>	Defines the next view to give focus to when the next focus is <a href="#">FOCUS_FORWARD</a> If the reference refers to a view that does not exist or is part of a hierarchy that is invisible, a <a href="#">RuntimeException</a> will result when the reference is accessed.
<u><a href="#">android:nextFocusLeft</a></u>	Defines the next view to give focus to when the next focus is <a href="#">FOCUS_LEFT</a> .
<u><a href="#">android:nextFocusRight</a></u>	Defines the next view to give focus to when the next focus is <a href="#">FOCUS_RIGHT</a> If the reference refers to a view that does not exist or is part of a hierarchy that is invisible, a <a href="#">RuntimeException</a> will result when the reference is accessed.
<u><a href="#">android:nextFocusUp</a></u>	Defines the next view to give focus to when the next focus is <a href="#">FOCUS_UP</a> If the reference refers to a view that does not exist or is part of a hierarchy that is invisible, a <a href="#">RuntimeException</a> will result when the reference is accessed.
<u><a href="#">android:onClick</a></u>	Name of the method in this View's context to invoke when the view is clicked.

<a href="#">android:overScrollMode</a>	Defines over-scrolling behavior.
<a href="#">android:padding</a>	Sets the padding, in pixels, of all four edges.
<a href="#">android:paddingBottom</a>	Sets the padding, in pixels, of the bottom edge; see <a href="#">padding</a> .
<a href="#">android:paddingEnd</a>	Sets the padding, in pixels, of the end edge; see <a href="#">padding</a> .
<a href="#">android:paddingLeft</a>	Sets the padding, in pixels, of the left edge; see <a href="#">padding</a> .
<a href="#">android:paddingRight</a>	Sets the padding, in pixels, of the right edge; see <a href="#">padding</a> .
<a href="#">android:paddingStart</a>	Sets the padding, in pixels, of the start edge; see <a href="#">padding</a> .
<a href="#">android:paddingTop</a>	Sets the padding, in pixels, of the top edge; see <a href="#">padding</a> .
<a href="#">android:requiresFadingEdge</a>	Defines which edges should be faded on scrolling.
<a href="#">android:rotation</a>	rotation of the view, in degrees.
<a href="#">android:rotationX</a>	rotation of the view around the x axis, in degrees.
<a href="#">android:rotationY</a>	rotation of the view around the y axis, in degrees.
<a href="#">android:saveEnabled</a>	If unset, no state will be saved for this view when it is being frozen.
<a href="#">android:scaleX</a>	scale of the view in the x direction.
<a href="#">android:scaleY</a>	scale of the view in the y direction.
<a href="#">android:scrollX</a>	The initial horizontal scroll offset, in pixels.
<a href="#">android:scrollY</a>	The initial vertical scroll offset, in pixels.
<a href="#">android:scrollbarAlwaysDrawHorizontalTrack</a>	Defines whether the horizontal scrollbar track should always be drawn.
<a href="#">android:scrollbarAlwaysDrawVerticalTrack</a>	Defines whether the vertical scrollbar track should always be drawn.
<a href="#">android:scrollbarDefaultDelayBeforeFade</a>	Defines the delay in milliseconds that a scrollbar waits before fade out.
<a href="#">android:scrollbarFadeDuration</a>	Defines the delay in milliseconds that a scrollbar takes to fade out.
<a href="#">android:scrollbarSize</a>	Sets the width of vertical scrollbars and height of horizontal scrollbars.
<a href="#">android:scrollbarStyle</a>	Controls the scrollbar style and position.
<a href="#">android:scrollbarThumbHorizontal</a>	Defines the horizontal scrollbar thumb drawable.
<a href="#">android:scrollbarThumbVertical</a>	Defines the vertical scrollbar thumb drawable.
<a href="#">android:scrollbarTrackHorizontal</a>	Defines the horizontal scrollbar track drawable.
<a href="#">android:scrollbarTrackVertical</a>	Defines the vertical scrollbar track drawable.
<a href="#">android:scrollbars</a>	Defines which scrollbars should be displayed on scrolling or not.
<a href="#">android:soundEffectsEnabled</a>	Boolean that controls whether a view should have sound effects enabled for events such as clicking and touching.
<a href="#">android:tag</a>	Supply a tag for this view containing a String, to be retrieved later with <a href="#">View.getTag()</a> or searched for with <a href="#">View.findViewWithTag()</a> .
<a href="#">android:textAlignment</a>	Defines the alignment of the text.
<a href="#">android:textDirection</a>	Defines the direction of the text.
<a href="#">android:transformPivotX</a>	x location of the pivot point around which the view will rotate and scale.
<a href="#">android:transformPivotY</a>	y location of the pivot point around which the view will rotate and scale.
<a href="#">android:translationX</a>	translation in x of the view.
<a href="#">android:translationY</a>	translation in y of the view.
<a href="#">android:verticalScrollbarPosition</a>	Determines which side the vertical scroll bar should be placed on.
<a href="#">android:visibility</a>	Controls the initial visibility of the view.

#### See Also

[View\\_alpha](#)  
[View\\_background](#)  
[View\\_clickable](#)

```
View_contentDescription
View_drawingCacheQuality
View_duplicateParentState
View_fadeScrollbars
View_fadingEdge
View_fadingEdgeLength
View_filterTouchesWhenObscured
View.fitsSystemWindows
View_focusable
View_focusableInTouchMode
View_hapticFeedbackEnabled
View_id
View_importantForAccessibility
View_isScrollContainer
View_keepScreenOn
View_labelFor
View_layerType
View_layoutDirection
View_longClickable
View_minHeight
View_minWidth
View_nextFocusDown
View_nextFocusForward
View_nextFocusLeft
View_nextFocusRight
View_nextFocusUp
View_onClick
View_overScrollMode
View_padding
View_paddingBottom
View_paddingEnd
View_paddingLeft
View_paddingRight
View_paddingStart
View_paddingTop
View_requiresFadingEdge
View_rotation
View_rotationX
View_rotationY
View_saveEnabled
View_scaleX
View_scaleY
View_scrollX
View_scrollY
View_scrollbarAlwaysDrawHorizontalTrack
View_scrollbarAlwaysDrawVerticalTrack
View_scrollbarDefaultDelayBeforeFade
View_scrollbarFadeDuration
View_scrollbarSize
View_scrollbarStyle
View_scrollbarThumbHorizontal
View_scrollbarThumbVertical
View_scrollbarTrackHorizontal
View_scrollbarTrackVertical
View_scrollers
View_soundEffectsEnabled
View_tag
View_textAlignment
View_textDirection
View_transformPivotX
View_transformPivotY
View_translationX
View_translationY
View_verticalScrollbarPosition
View_visibility
```

**public static final int[] ViewAnimator**

Attributes that can be used with a ViewAnimator.

Includes the following attributes:

Attribute	Description
-----------	-------------

[android:animateFirstView](#) Defines whether to animate the current View when the ViewAnimation is first displayed.

[android:inAnimation](#) Identifier for the animation to use when a view is shown.

[android:outAnimation](#) Identifier for the animation to use when a view is hidden.

#### See Also

[ViewAnimator\\_animateFirstView](#)

[ViewAnimator\\_inAnimation](#)

[ViewAnimator\\_outAnimation](#)

## public static final int[] **ViewDrawableStates**

Attributes that can be used with a ViewDrawableStates.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:state_accelerated</u></a>	State value for <a href="#"><u>StateListDrawable</u></a> , indicating that the Drawable is in a view that is hardware accelerated.
<a href="#"><u>android:state_activated</u></a>	State value for <a href="#"><u>StateListDrawable</u></a> , set when a view or its parent has been "activated" meaning the user has currently marked it as being of interest.
<a href="#"><u>android:state_drag_can_accept</u></a>	State for <a href="#"><u>StateListDrawable</u></a> indicating that the Drawable is in a view that is capable of accepting a drop of the content currently being manipulated in a drag-and-drop operation.
<a href="#"><u>android:state_drag_hovered</u></a>	State for <a href="#"><u>StateListDrawable</u></a> indicating that a drag operation (for which the Drawable's view is a valid recipient) is currently positioned over the Drawable.
<a href="#"><u>android:state_enabled</u></a>	State value for <a href="#"><u>StateListDrawable</u></a> , set when a view is enabled.
<a href="#"><u>android:state_focused</u></a>	State value for <a href="#"><u>StateListDrawable</u></a> , set when a view has input focus.
<a href="#"><u>android:state_hovered</u></a>	State value for <a href="#"><u>StateListDrawable</u></a> , set when a pointer is hovering over the view.
<a href="#"><u>android:state_pressed</u></a>	State value for <a href="#"><u>StateListDrawable</u></a> , set when the user is pressing down in a view.
<a href="#"><u>android:state_selected</u></a>	State value for <a href="#"><u>StateListDrawable</u></a> , set when a view (or one of its parents) is currently selected.
<a href="#"><u>android:state_window_focused</u></a>	State value for <a href="#"><u>StateListDrawable</u></a> , set when a view's window has input focus.

#### See Also

[ViewDrawableStates\\_state\\_accelerated](#)

[ViewDrawableStates\\_state\\_activated](#)

[ViewDrawableStates\\_state\\_drag\\_can\\_accept](#)

[ViewDrawableStates\\_state\\_drag\\_hovered](#)

[ViewDrawableStates\\_state\\_enabled](#)

[ViewDrawableStates\\_state\\_focused](#)

[ViewDrawableStates\\_state\\_hovered](#)

[ViewDrawableStates\\_state\\_pressed](#)

[ViewDrawableStates\\_state\\_selected](#)

[ViewDrawableStates\\_state\\_window\\_focused](#)

## public static final int[] **ViewFlipper**

Attributes that can be used with a ViewFlipper.

Includes the following attributes:

Attribute	Description
<a href="#"><u>android:autoStart</u></a>	When true, automatically start animating
<a href="#"><u>android:flipInterval</u></a>	

#### See Also

[ViewFlipper\\_autoStart](#)

[ViewFlipper\\_flipInterval](#)

## public static final int[] **ViewGroup**

Attributes that can be used with a [viewGroup](#) ([/reference/android/view/ViewGroup.html](#)) or any of its subclasses. Also see [viewGroup\\_Layout](#) ([/reference/android/R.styleable.html#ViewGroup\\_Layout](#)) for attributes that this class processes in its children.

Includes the following attributes:

Attribute	Description
<a href="#">android:addStatesFromChildren</a>	Sets whether this ViewGroup's drawable states also include its children's drawable states.
<a href="#">android:alwaysDrawnWithCache</a>	Defines whether the ViewGroup should always draw its children using their drawing cache or not.
<a href="#">android:animateLayoutChanges</a>	Defines whether changes in layout (caused by adding and removing items) should cause a LayoutTransition to run.
<a href="#">android:animationCache</a>	Defines whether layout animations should create a drawing cache for their children.
<a href="#">android:clipChildren</a>	Defines whether a child is limited to draw inside of its bounds or not.
<a href="#">android:clipToPadding</a>	Defines whether the ViewGroup will clip its drawing surface so as to exclude the padding area.
<a href="#">android:descendantFocusability</a>	Defines the relationship between the ViewGroup and its descendants when looking for a View to take focus.
<a href="#">android:layoutAnimation</a>	Defines the layout animation to use the first time the ViewGroup is laid out.
<a href="#">android:persistentDrawingCache</a>	Defines the persistence of the drawing cache.
<a href="#">android:splitMotionEvents</a>	Sets whether this ViewGroup should split MotionEvents to separate child views during touch event dispatch.

#### See Also

[ViewGroup\\_addStatesFromChildren](#)  
[ViewGroup\\_alwaysDrawnWithCache](#)  
[ViewGroup\\_animateLayoutChanges](#)  
[ViewGroup\\_animationCache](#)  
[ViewGroup\\_clipChildren](#)  
[ViewGroup\\_clipToPadding](#)  
[ViewGroup\\_descendantFocusability](#)  
[ViewGroup\\_layoutAnimation](#)  
[ViewGroup\\_persistentDrawingCache](#)  
[ViewGroup\\_splitMotionEvents](#)

## public static final int[] ViewGroup\_Layout

This is the basic set of layout attributes that are common to all layout managers. These attributes are specified with the rest of a view's normal attributes (such as [background](#) [/reference/android/R.attr.html#background](#)), but will be parsed by the view's parent and ignored by the child.

The values defined here correspond to the base layout attribute class [ViewGroup.LayoutParams](#) [\(/reference/android/view/ViewGroup.LayoutParams.html\)](#).

Includes the following attributes:

Attribute	Description
<a href="#">android:layout_height</a>	Specifies the basic height of the view.
<a href="#">android:layout_width</a>	Specifies the basic width of the view.

#### See Also

[ViewGroup\\_Layout\\_layout\\_height](#)  
[ViewGroup\\_Layout\\_layout\\_width](#)

## public static final int[] ViewGroup\_MarginLayout

This is the basic set of layout attributes for layout managers that wish to place margins around their child views. These attributes are specified with the rest of a view's normal attributes (such as [background](#) [/reference/android/R.attr.html#background](#)), but will be parsed by the view's parent and ignored by the child.

The values defined here correspond to the base layout attribute class [ViewGroup.MarginLayoutParams](#) [\(/reference/android/view/ViewGroup.MarginLayoutParams.html\)](#).

Includes the following attributes:

Attribute	Description
<a href="#">android:layout_height</a>	Specifies the basic height of the view.
<a href="#">android:layout_margin</a>	Specifies extra space on the left, top, right and bottom sides of this view.
<a href="#">android:layout_marginBottom</a>	Specifies extra space on the bottom side of this view.

<a href="#">android:layout_marginEnd</a>	Specifies extra space on the end side of this view.
<a href="#">android:layout_marginLeft</a>	Specifies extra space on the left side of this view.
<a href="#">android:layout_marginRight</a>	Specifies extra space on the right side of this view.
<a href="#">android:layout_marginStart</a>	Specifies extra space on the start side of this view.
<a href="#">android:layout_marginTop</a>	Specifies extra space on the top side of this view.
<a href="#">android:layout_width</a>	Specifies the basic width of the view.

**See Also**

[ViewGroup.MarginLayoutParams.layout\\_height](#)  
[ViewGroup.MarginLayoutParams.layout\\_margin](#)  
[ViewGroup.MarginLayoutParams.layout\\_marginBottom](#)  
[ViewGroup.MarginLayoutParams.layout\\_marginEnd](#)  
[ViewGroup.MarginLayoutParams.layout\\_marginLeft](#)  
[ViewGroup.MarginLayoutParams.layout\\_marginRight](#)  
[ViewGroup.MarginLayoutParams.layout\\_marginStart](#)  
[ViewGroup.MarginLayoutParams.layout\\_marginTop](#)  
[ViewGroup.MarginLayoutParams.layout\\_width](#)

## public static final int[] ViewStub

A [ViewStub](#) ([/reference/android/view/ViewStub.html](#)) lets you lazily include other XML layouts inside your application at runtime.

Includes the following attributes:

Attribute	Description
<a href="#">android:inflatedId</a>	Overrides the id of the inflated View with this value.
<a href="#">android:layout</a>	Supply an identifier for the layout resource to inflate when the ViewStub becomes visible or when forced to do so.

**See Also**

[ViewStub.inflatedId](#)  
[ViewStub.layout](#)

## public static final int[] ViewSwitcher

Attributes that can be used with a ViewSwitcher.

## public static final int[] VolumePreference

Base attributes available to VolumePreference.

Includes the following attributes:

Attribute	Description
<a href="#">android:streamType</a>	Different audio stream types.

**See Also**

[VolumePreference.streamType](#)

## public static final int[] Wallpaper

Use wallpaper as the root tag of the XML resource that describes an [WallpaperService](#) ([/reference/android/service/wallpaper/WallpaperService.html](#)), which is referenced from its [SERVICE\\_META\\_DATA](#) ([/reference/android/service/wallpaper/WallpaperService.html#SERVICE\\_META\\_DATA](#)) meta-data entry. Described here are the attributes that can be included in that tag.

Includes the following attributes:

Attribute	Description
<a href="#">android:author</a>	Name of the author of this component, e.
<a href="#">android:description</a>	Short description of the component's purpose or behavior.
<a href="#">android:settingsActivity</a>	Component name of an activity that allows the user to modify the settings for this service.
<a href="#">android:thumbnail</a>	Reference to a the wallpaper's thumbnail bitmap.

**See Also**

[Wallpaper.author](#)  
[Wallpaper.description](#)  
[Wallpaper.settingsActivity](#)  
[Wallpaper.thumbnail](#)

## public static final int[] **WallpaperPreviewInfo**

Use `wallpaper-preview` as the root tag of the XML resource that describes a wallpaper preview.

Includes the following attributes:

Attribute	Description
<a href="#">android:staticWallpaperPreview</a>	A resource id of a static drawable.

### See Also

[WallpaperPreviewInfo.staticWallpaperPreview](#)

## public static final int[] **Window**

The set of attributes that describe a Windows's theme.

Includes the following attributes:

Attribute	Description
<a href="#">android:backgroundDimAmount</a>	Default background dim amount when a menu, dialog, or something similar pops up.
<a href="#">android:backgroundDimEnabled</a>	Control whether dimming behind the window is enabled.
<a href="#">android:textColor</a>	Color of text (usually same as colorForeground).
<a href="#">android:windowActionBar</a>	Flag indicating whether this window should have an Action Bar in place of the usual title bar.
<a href="#">android:windowActionBarOverlay</a>	Flag indicating whether this window's Action Bar should overlay application content.
<a href="#">android:windowActionModeOverlay</a>	Flag indicating whether action modes should overlay window content when there is not reserved space for their UI (such as an Action Bar).
<a href="#">android:windowAnimationStyle</a>	Reference to a style resource holding the set of window animations to use, which can be any of the attributes defined by <a href="#">WindowAnimation</a> .
<a href="#">android:windowBackground</a>	Drawable to use as the overall window background.
<a href="#">android:windowCloseOnTouchOutside</a>	Control whether a container should automatically close itself if the user touches outside of it.
<a href="#">android:windowContentOverlay</a>	This Drawable is overlaid over the foreground of the Window's content area, usually to place a shadow below the title.
<a href="#">android:windowDisablePreview</a>	Flag allowing you to disable the preview animation for a window.
<a href="#">android:windowEnableSplitTouch</a>	Flag indicating that this window should allow touches to be split across other windows that also support split touch.
<a href="#">android:windowFrame</a>	Drawable to use as a frame around the window.
<a href="#">android:windowFullscreen</a>	Flag indicating whether this window should fill the entire screen.
<a href="#">android:windowIsFloating</a>	Flag indicating whether this is a floating window.
<a href="#">android:windowIsTranslucent</a>	Flag indicating whether this is a translucent window.
<a href="#">android:windowMinWidthMajor</a>	The minimum width the window is allowed to be, along the major axis of the screen.
<a href="#">android:windowMinWidthMinor</a>	The minimum width the window is allowed to be, along the minor axis of the screen.
<a href="#">android:windowNoDisplay</a>	Flag indicating that this window should not be displayed at all.
<a href="#">android:windowNoTitle</a>	Flag indicating whether there should be no title on this window.
<a href="#">android:windowShowWallpaper</a>	Flag indicating that this window's background should be the user's current wallpaper.
<a href="#">android:windowSoftInputMode</a>	Defines the default soft input state that this window would like when it is displayed.

### See Also

[Window.backgroundDimAmount](#)  
[Window.backgroundDimEnabled](#)  
[Window.textColor](#)  
[Window.windowActionBar](#)  
[Window.windowActionBarOverlay](#)  
[Window.windowActionModeOverlay](#)  
[Window.windowAnimationStyle](#)  
[Window.windowBackground](#)  
[Window.windowCloseOnTouchOutside](#)

```
Window windowContentOverlay
Window windowDisablePreview
Window windowEnableSplitTouch
Window windowFrame
Window windowFullscreen
Window windowIsFloating
Window windowIsTranslucent
Window windowMinWidthMajor
Window windowMinWidthMinor
Window windowNoDisplay
Window windowNoTitle
Window windowShowWallpaper
Window windowSoftInputMode
```

```
public static final int[] WindowAnimation
```

Window animation class attributes.

Includes the following attributes:

Attribute	Description
<u>android:activityCloseEnterAnimation</u>	When closing the current activity, this is the animation that is run on the next activity (which is entering the screen).
<u>android:activityCloseExitAnimation</u>	When closing the current activity, this is the animation that is run on the current activity (which is exiting the screen).
<u>android:activityOpenEnterAnimation</u>	When opening a new activity, this is the animation that is run on the next activity (which is entering the screen).
<u>android:activityOpenExitAnimation</u>	When opening a new activity, this is the animation that is run on the previous activity (which is exiting the screen).
<u>android:taskCloseEnterAnimation</u>	When closing the last activity of a task, this is the animation that is run on the activity of the next task (which is entering the screen).
<u>android:taskCloseExitAnimation</u>	When opening an activity in a new task, this is the animation that is run on the activity of the old task (which is exiting the screen).
<u>android:taskOpenEnterAnimation</u>	When opening an activity in a new task, this is the animation that is run on the activity of the new task (which is entering the screen).
<u>android:taskOpenExitAnimation</u>	When opening an activity in a new task, this is the animation that is run on the activity of the old task (which is exiting the screen).
<u>android:taskToBackEnterAnimation</u>	When sending the current task to the background, this is the animation that is run on the top activity of the task behind it (which is entering the screen).
<u>android:taskToBackExitAnimation</u>	When sending the current task to the background, this is the animation that is run on the top activity of the current task (which is exiting the screen).
<u>android:taskToFrontEnterAnimation</u>	When bringing an existing task to the foreground, this is the animation that is run on the top activity of the task being brought to the foreground (which is entering the screen).
<u>android:taskToFrontExitAnimation</u>	When bringing an existing task to the foreground, this is the animation that is run on the current foreground activity (which is exiting the screen).
<u>android:wallpaperCloseEnterAnimation</u>	When opening a new activity that hides the wallpaper, while currently showing the wallpaper, this is the animation that is run on the new activity (which is entering the screen).
<u>android:wallpaperCloseExitAnimation</u>	When opening a new activity that hides the wallpaper, while currently showing the wallpaper, this is the animation that is run on the old wallpaper activity (which is exiting the screen).
<u>android:wallpaperIntraCloseEnterAnimation</u>	When closing a foreground activity that is on top of the wallpaper when the previous activity is also
	on top of the wallpaper, this is the animation that is run on the previous activity (which is entering the screen).
	When closing a foreground activity that is on top of

<a href="#"><u>android:wallpaperIntraCloseExitAnimation</u></a>	the wallpaper when the previous activity is also on top of the wallpaper, this is the animation that is run on the current activity (which is exiting the screen).
<a href="#"><u>android:wallpaperIntraOpenEnterAnimation</u></a>	When opening a new activity that is on top of the wallpaper when the current activity is also on top of the wallpaper, this is the animation that is run on the new activity (which is entering the screen).
<a href="#"><u>android:wallpaperIntraOpenExitAnimation</u></a>	When opening a new activity that is on top of the wallpaper when the current activity is also on top of the wallpaper, this is the animation that is run on the current activity (which is exiting the screen).
<a href="#"><u>android:wallpaperOpenEnterAnimation</u></a>	When opening a new activity that shows the wallpaper, while currently not showing the wallpaper, this is the animation that is run on the new wallpaper activity (which is entering the screen).
<a href="#"><u>android:wallpaperOpenExitAnimation</u></a>	When opening a new activity that shows the wallpaper, while currently not showing the wallpaper, this is the animation that is run on the current activity (which is exiting the screen).
<a href="#"><u>android:windowEnterAnimation</u></a>	The animation used when a window is being added.
<a href="#"><u>android:windowExitAnimation</u></a>	The animation used when a window is being removed.
<a href="#"><u>android:windowHideAnimation</u></a>	The animation used when a window is going from VISIBLE to INVISIBLE.
<a href="#"><u>android:windowShowAnimation</u></a>	The animation used when a window is going from INVISIBLE to VISIBLE.

#### See Also

[WindowAnimation\\_activityCloseEnterAnimation](#)  
[WindowAnimation\\_activityCloseExitAnimation](#)  
[WindowAnimation\\_activityOpenEnterAnimation](#)  
[WindowAnimation\\_activityOpenExitAnimation](#)  
[WindowAnimation\\_taskCloseEnterAnimation](#)  
[WindowAnimation\\_taskCloseExitAnimation](#)  
[WindowAnimation\\_taskOpenEnterAnimation](#)  
[WindowAnimation\\_taskOpenExitAnimation](#)  
[WindowAnimation\\_taskToBackEnterAnimation](#)  
[WindowAnimation\\_taskToBackExitAnimation](#)  
[WindowAnimation\\_taskToFrontEnterAnimation](#)  
[WindowAnimation\\_taskToFrontExitAnimation](#)  
[WindowAnimation\\_wallpaperCloseEnterAnimation](#)  
[WindowAnimation\\_wallpaperCloseExitAnimation](#)  
[WindowAnimation\\_wallpaperIntraCloseEnterAnimation](#)  
[WindowAnimation\\_wallpaperIntraCloseExitAnimation](#)  
[WindowAnimation\\_wallpaperIntraOpenEnterAnimation](#)  
[WindowAnimation\\_wallpaperIntraOpenExitAnimation](#)  
[WindowAnimation\\_wallpaperOpenEnterAnimation](#)  
[WindowAnimation\\_wallpaperOpenExitAnimation](#)  
[WindowAnimation\\_windowEnterAnimation](#)  
[WindowAnimation\\_windowExitAnimation](#)  
[WindowAnimation\\_windowHideAnimation](#)  
[WindowAnimation\\_windowShowAnimation](#)

---

## Public Constructors

```
public R.styleable()
```