

Gallery

extends [AbsSpinner](#)

implements [GestureDetector.OnGestureListener](#)

[java.lang.Object](#)

↳ [android.view.View](#)

↳ [android.view.ViewGroup](#)

↳ [android.widget.AdapterView](#) <T extends [android.widget.Adapter](#)>

↳ [android.widget.AbsSpinner](#)

↳ [android.widget.Gallery](#)

Class Overview

A view that shows items in a center-locked, horizontally scrolling list.

The default values for the Gallery assume you will be using [Theme_galleryItemBackground](#) as the background for each View given to the Gallery from the Adapter. If you are not doing this, you may need to adjust some Gallery properties, such as the spacing.

Views given to the Gallery should use [Gallery.LayoutParams](#) as their layout parameters type.

See the [Gallery tutorial](#).

Summary

Nested Classes

class	Gallery.LayoutParams	Gallery extends LayoutParams to provide a place to hold current Transformation information/position/transformation info.
-------	--------------------------------------	--

XML Attributes

Attribute Name	Related Method	Description
----------------	----------------	-------------

android:animationDuration	setAnimationDuration(int)	Sets how long a transition animation should run (in milliseconds)
android:gravity	setGravity(int)	Specifies how to place the content of an object, both on the screen and within the object itself.
android:spacing	setSpacing(int)	
android:unselectedAlpha	setUnselectedAlpha(float)	Sets the alpha on the items that are not selected.

[\[Expand\]](#)

Inherited XML Attributes

► From class [android.widget.AbsSpinner](#)

► From class [android.view.ViewGroup](#)

► From class [android.view.View](#)

[\[Expand\]](#)

Inherited Constants

► From class [android.widget.AdapterView](#)

► From class [android.view.ViewGroup](#)

► From class [android.view.View](#)

[\[Expand\]](#)

Inherited Fields

► From class [android.view.View](#)

Public Constructors

[Gallery\(Context context\)](#)

	<code>Gallery(Context context, AttributeSet attrs)</code>
	<code>Gallery(Context context, AttributeSet attrs, int defStyle)</code>
Public Methods	
boolean	<code>dispatchKeyEvent(KeyEvent event)</code> Dispatch a key event to the next view on the focus path.
void	<code>dispatchSetSelected(boolean selected)</code> Dispatch setSelected to all of this View's children.
<code>ViewGroup.LayoutParams</code>	<code>generateLayoutParams(AttributeSet attrs)</code> Returns a new set of layout parameters based on the supplied attributes set.
boolean	<code>onDown(MotionEvent e)</code> Notified when a tap occurs with the down <code>MotionEvent</code> that triggered it.
boolean	<code>onFling(MotionEvent e1, MotionEvent e2, float velocityX, float velocityY)</code> Notified of a fling event when it occurs with the initial on down <code>MotionEvent</code> and the m
boolean	<code>onKeyDown(int keyCode, KeyEvent event)</code> Handles left, right, and clicking
boolean	<code>onKeyUp(int keyCode, KeyEvent event)</code> Default implementation of <code>KeyEvent.Callback.onKeyUp()</code> : perform clicking of the when <code>KEYCODE_DPAD_CENTER</code> or <code>KEYCODE_ENTER</code> is released.
void	<code>onLongPress(MotionEvent e)</code> Notified when a long press occurs with the initial on down <code>MotionEvent</code> that trigged it.

boolean	<p>onScroll(MotionEvent e1, MotionEvent e2, float distanceX, float distanceY)</p> <p>Notified when a scroll occurs with the initial on down MotionEvent and the current mo</p>
void	<p>onShowPress(MotionEvent e)</p> <p>The user has performed a down MotionEvent and not performed a move or up yet.</p>
boolean	<p>onSingleTapUp(MotionEvent e)</p> <p>Notified when a tap occurs with the up MotionEvent that triggered it.</p>
boolean	<p>onTouchEvent(MotionEvent event)</p> <p>Implement this method to handle touch screen motion events.</p>
void	<p>setAnimationDuration(int animationDurationMillis)</p> <p>Sets how long the transition animation should run when a child view changes position.</p>
void	<p>setCallbackDuringFling(boolean shouldCallback)</p> <p>Whether or not to callback on any getItemSelectedListener() while the items a</p>
void	<p>setGravity(int gravity)</p> <p>Describes how the child views are aligned.</p>
void	<p>setSpacing(int spacing)</p> <p>Sets the spacing between items in a Gallery</p>
void	<p>setUnselectedAlpha(float unselectedAlpha)</p> <p>Sets the alpha of items that are not selected in the Gallery.</p>
boolean	<p>showContextMenu()</p> <p>Bring up the context menu for this view.</p>

boolean	showContextMenuForChild(View originalView) Bring up a context menu for the specified view or its ancestors.
Protected Methods	
boolean	checkLayoutParams(ViewGroup.LayoutParams p)
int	computeHorizontalScrollExtent() Compute the horizontal extent of the horizontal scrollbar's thumb within the horizontal range.
int	computeHorizontalScrollOffset() Compute the horizontal offset of the horizontal scrollbar's thumb within the horizontal range.
int	computeHorizontalScrollRange() Compute the horizontal range that the horizontal scrollbar represents.
void	dispatchSetPressed(boolean pressed) Dispatch setPressed to all of this View's children.
ViewGroup.LayoutParams	generateDefaultLayoutParams() Returns a set of default layout parameters.
ViewGroup.LayoutParams	generateLayoutParams(ViewGroup.LayoutParams p) Returns a safe set of layout parameters based on the supplied layout params.
int	getChildDrawingOrder(int childCount, int i) Returns the index of the child to draw for this iteration.
boolean	getChildStaticTransformation(View child, Transformation t)
ContextMenu.ContextMenuInfo	getContextMenuInfo()

	Views should implement this if they have extra information to associate with the c
void	onFocusChanged (boolean gainFocus, int direction, Rect previouslyFocusedRect) Called by the view system when the focus state of this view changes.
void	onLayout (boolean changed, int l, int t, int r, int b) Called from layout when this view should assign a size and position to each of its c

[Expand]

Inherited Methods

- ▶ From class [android.widget.AbsSpinner](#)
- ▶ From class [android.widget.AdapterView](#)
- ▶ From class [android.view.ViewGroup](#)
- ▶ From class [android.view.View](#)
- ▶ From class [java.lang.Object](#)
- ▶ From interface [android.graphics.drawable.Drawable.Callback](#)
- ▶ From interface [android.view.GestureDetector.OnGestureListener](#)
- ▶ From interface [android.view.KeyEvent.Callback](#)
- ▶ From interface [android.view.ViewManager](#)
- ▶ From interface [android.view.ViewParent](#)
- ▶ From interface [android.view.accessibility.AccessibilityEventSource](#)

XML Attributes

android: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](#).

Related Methods

- [setAnimationDuration\(int\)](#)

android:gravity

Specifies how to place the content of an object, both on the x- and y-axis, within the object itself.

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.
<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 [gravity](#).

Related Methods

- [setGravity\(int\)](#)

android:spacing

Related Methods

- [setSpacing\(int\)](#)

android: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](#).

Related Methods

- [setUnselectedAlpha\(float\)](#)

Public Constructors

public Gallery ([Context](#) context)

Since: API Level 1

public Gallery ([Context](#) context, [AttributeSet](#) attrs)

Since: API Level 1

public Gallery ([Context](#) context, [AttributeSet](#) attrs, int defStyle)

Since: API Level 1

Public Methods

public boolean dispatchKeyEvent ([KeyEvent](#) event)

Since: API Level 1

Dispatch a key event to the next view on the focus path. This path runs from the top of the view tree down to the currently focused view. If this view has focus, it will dispatch to itself. Otherwise it will dispatch the next node down the focus path. This method also fires any key listeners.

Parameters

event The key event to be dispatched.

Returns

- True if the event was handled, false otherwise.

```
public void dispatchSetSelected (boolean selected)
```

Since: API Level 1

Dispatch setSelected to all of this View's children.

Parameters

selected The new selected state

```
public ViewGroup.LayoutParams generateLayoutParams (AttributeSet attrs)
```

Since: API Level 1

Returns a new set of layout parameters based on the supplied attributes set.

Parameters

attrs the attributes to build the layout parameters from

Returns

- an instance of [ViewGroup.LayoutParams](#) or one of its descendants

```
public boolean onDown (MotionEvent e)
```

Since: API Level 1

Notified when a tap occurs with the down [MotionEvent](#) that triggered it. This will be triggered immediately for every down event. All other events should be preceded by this.

Parameters

e The down motion event.

```
public boolean onFling (MotionEvent e1, MotionEvent e2, float velocityX, float velocityY)
```

Since: API Level 1

Notified of a fling event when it occurs with the initial on down [MotionEvent](#) and the matching up [MotionEvent](#). The calculated velocity is supplied along the x and y axis in pixels per second.

Parameters

- e1** The first down motion event that started the fling.
- e2** The move motion event that triggered the current onFling.
- velocityX** The velocity of this fling measured in pixels per second along the x axis.
- velocityY** The velocity of this fling measured in pixels per second along the y axis.

Returns

- true if the event is consumed, else false

```
public boolean onKeyDown (int keyCode, KeyEvent event)
```

Since: API Level 1

Handles left, right, and clicking

Parameters

- keyCode** A key code that represents the button pressed, from [KeyEvent](#).
- event** The KeyEvent object that defines the button action.

Returns

- If you handled the event, return true. If you want to allow the event to be handled by the next receiver, return false.

See Also

- [onKeyDown\(int, KeyEvent\)](#)

```
public boolean onKeyUp (int keyCode, KeyEvent event)
```

Since: API Level 1

Default implementation of [KeyEvent.Callback.onKeyUp\(\)](#): perform clicking of the view when [KEYCODE_DPAD_CENTER](#) or [KEYCODE_ENTER](#) is released.

Parameters

keyCode A key code that represents the button pressed, from [KeyEvent](#).

event The KeyEvent object that defines the button action.

Returns

- If you handled the event, return true. If you want to allow the event to be handled by the next receiver, return false.

```
public void onLongPress (MotionEvent e)
```

Since: API Level 1

Notified when a long press occurs with the initial on down [MotionEvent](#) that triggered it.

Parameters

e The initial on down motion event that started the longpress.

```
public boolean onScroll (MotionEvent e1, MotionEvent e2, float distanceX, float distanceY)
```

Since: API Level 1

Notified when a scroll occurs with the initial on down [MotionEvent](#) and the current move [MotionEvent](#). The distance in x and y is also supplied for convenience.

Parameters

e1 The first down motion event that started the scrolling.

e2 The move motion event that triggered the current onScroll.

distanceX The distance along the X axis that has been scrolled since the last call to onScroll. This is NOT the distance between **e1** and **e2**.

distanceY The distance along the Y axis that has been scrolled since the last call to onScroll. This is NOT the distance between **e1** and **e2**.

Returns

- true if the event is consumed, else false

```
public void onShowPress (MotionEvent e)
```

Since: API Level 1

The user has performed a down [MotionEvent](#) and not performed a move or up yet. This event is commonly used to provide visual feedback to the user to let them know that their action has been recognized i.e. highlight an element.

Parameters

- e* The down motion event

```
public boolean onSingleTapUp (MotionEvent e)
```

Since: API Level 1

Notified when a tap occurs with the up [MotionEvent](#) that triggered it.

Parameters

- e* The up motion event that completed the first tap

Returns

- true if the event is consumed, else false

```
public boolean onTouchEvent (MotionEvent event)
```

Since: API Level 1

Implement this method to handle touch screen motion events.

Parameters

- event* The motion event.

Returns

- True if the event was handled, false otherwise.

```
public void setAnimationDuration (int animationDurationMillis)
```

Since: API Level 1

Sets how long the transition animation should run when a child view changes position. Only relevant if animation is turned on.

Related XML Attributes

- [android:animationDuration](#)

Parameters

animationDurationMillis The duration of the transition, in milliseconds.

```
public void setCallbackDuringFling (boolean shouldCallback)
```

Since: API Level 1

Whether or not to callback on any [getOnItemSelectedListener\(\)](#) while the items are being flinged. If false, only the final selected item will cause the callback. If true, all items between the first and the final will cause callbacks.

Parameters

shouldCallback Whether or not to callback on the listener while the items are being flinged.

```
public void setGravity (int gravity)
```

Since: API Level 1

Describes how the child views are aligned.

Related XML Attributes

- [android:gravity](#)

```
public void setSpacing (int spacing)
```

Since: API Level 1

Sets the spacing between items in a Gallery

Related XML Attributes

- [android:spacing](#)

Parameters

spacing The spacing in pixels between items in the Gallery

```
public void setUnselectedAlpha (float unselectedAlpha)
```

Since: API Level 1

Sets the alpha of items that are not selected in the Gallery.

Related XML Attributes

- [android:unselectedAlpha](#)

Parameters

unselectedAlpha the alpha for the items that are not selected.

```
public boolean showContextMenu ()
```

Since: API Level 1

Bring up the context menu for this view.

Returns

- Whether a context menu was displayed.

```
public boolean showContextMenuForChild (View originalView)
```

Since: API Level 1

Bring up a context menu for the specified view or its ancestors.

In most cases, a subclass does not need to override this. However, if the subclass is added directly to the window manager (for example, [addView\(View, android.view.ViewGroup.LayoutParams\)](#)) then it should override this and show the context menu.

Parameters

originalView The source view where the context menu was first invoked

Returns

- true if a context menu was displayed

Protected Methods

```
protected boolean checkLayoutParams (ViewGroup.LayoutParams p)
```

Since: API Level 1

```
protected int computeHorizontalScrollExtent ()
```

Since: API Level 1

Compute the horizontal extent of the horizontal scrollbar's thumb within the horizontal range. This value is used to compute the length of the thumb within the scrollbar's track.

The range is expressed in arbitrary units that must be the same as the units used by [computeHorizontalScrollRange\(\)](#) and [computeHorizontalScrollOffset\(\)](#).

The default extent is the drawing width of this view.

Returns

- the horizontal extent of the scrollbar's thumb

```
protected int computeHorizontalScrollOffset ()
```

Since: API Level 1

Compute the horizontal offset of the horizontal scrollbar's thumb within the horizontal range. This value is used to compute the position of the thumb within the scrollbar's track.

The range is expressed in arbitrary units that must be the same as the units used by [computeHorizontalScrollRange\(\)](#) and [computeHorizontalScrollExtent\(\)](#).

The default offset is the scroll offset of this view.

Returns

- the horizontal offset of the scrollbar's thumb

```
protected int computeHorizontalScrollRange ()
```

Since: API Level 1

Compute the horizontal range that the horizontal scrollbar represents.

The range is expressed in arbitrary units that must be the same as the units used by [computeHorizontalScrollExtent\(\)](#) and [computeHorizontalScrollOffset\(\)](#).

The default range is the drawing width of this view.

Returns

- the total horizontal range represented by the horizontal scrollbar

```
protected void dispatchSetPressed (boolean pressed)
```

Since: API Level 1

Dispatch setPressed to all of this View's children.

Parameters

pressed The new pressed state

protected [ViewGroup.LayoutParams](#) **generateDefaultLayoutParams** ()

Since: API Level 1

Returns a set of default layout parameters. These parameters are requested when the View passed to [addView\(View\)](#) has no layout parameters already set. If null is returned, an exception is thrown from [addView](#).

Returns

- a set of default layout parameters or null

protected [ViewGroup.LayoutParams](#) **generateLayoutParams** ([ViewGroup.LayoutParams](#) p)

Since: API Level 1

Returns a safe set of layout parameters based on the supplied layout params. When a ViewGroup is passed a View whose layout params do not pass the test of [checkLayoutParams\(android.view.ViewGroup.LayoutParams\)](#), this method is invoked. This method should return a new set of layout params suitable for this ViewGroup, possibly by copying the appropriate attributes from the specified set of layout params.

Parameters

- p* The layout parameters to convert into a suitable set of layout parameters for this ViewGroup.

Returns

- an instance of [ViewGroup.LayoutParams](#) or one of its descendants

protected int **getChildDrawingOrder** (int childCount, int i)

Since: API Level 1

Returns the index of the child to draw for this iteration. Override this if you want to change the drawing order of children. By default, it returns i.

NOTE: In order for this method to be called, you must enable child ordering first by calling [setChildrenDrawingOrderEnabled\(boolean\)](#).

Parameters

- i The current iteration.

Returns

- The index of the child to draw this iteration.

protected boolean **getChildStaticTransformation** ([View](#) child, [Transformation](#) t)

Since: API Level 1

protected [ContextMenu.ContextMenuInfo](#) **getContextMenuInfo** ()

Since: API Level 1

Views should implement this if they have extra information to associate with the context menu. The return result is supplied as a parameter to the [onCreateContextMenu \(ContextMenu, View, ContextMenuInfo\)](#) callback.

Returns

- Extra information about the item for which the context menu should be shown. This information will vary across different subclasses of View.

protected void **onFocusChanged** (*boolean* gainFocus, *int* direction, [Rect](#) previouslyFocusedRect)

Since: API Level 1

Called by the view system when the focus state of this view changes. When the focus change event is caused by directional navigation, direction and previouslyFocusedRect provide insight into where the focus is coming from. When overriding, be sure to call up through to the super class so that the standard focus handling will occur.

Parameters

<i>gainFocus</i>	True if the View has focus; false otherwise.
<i>direction</i>	The direction focus has moved when requestFocus() is called to give this view focus. Values are FOCUS_UP , FOCUS_DOWN , FOCUS_LEFT , FOCUS_RIGHT , FOCUS_FORWARD , or FOCUS_BACKWARD . It may not always apply, in which case use the default.
<i>previouslyFocusedRect</i>	The rectangle, in this view's coordinate system, of the previously focused view. If applicable, this will be passed in as finer grained information about where the focus is coming from (in addition to direction). Will

be `null` otherwise.

*protected void **onLayout** (boolean changed, int l, int t, int r, int b)*

Since: API Level 1

Called from layout when this view should assign a size and position to each of its children. Derived classes with children should override this method and call layout on each of their children.

Parameters

changed This is a new size or position for this view

l Left position, relative to parent

t Top position, relative to parent

r Right position, relative to parent

b Bottom position, relative to parent