

Drawable

```
public abstract class Drawable  
extends Object  
(https://developer.android.com/reference/java/lang/Object.html)
```

```
java.lang.Object (https://developer.android.com/reference/java/lang/Object.html)  
↳ android.graphics.drawable.Drawable
```

added in API level 1
(<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)
Nested Classes (#nestedclasses) | Ctors (#ctors) | Protected Methods (#protectedmethods) | Inherited Methods (#inheritedmethods) | [Expand All] (#)

▼ (#)Known Direct Subclasses

AdaptiveIconDrawable

(<https://developer.android.com/reference/android/graphics/drawable/AdaptiveIconDrawable.html>), AnimatedVectorDrawable
(<https://developer.android.com/reference/android/graphics/drawable/AnimatedVectorDrawable.html>), BitmapDrawable
(<https://developer.android.com/reference/android/graphics/drawable/BitmapDrawable.html>), ColorDrawable
(<https://developer.android.com/reference/android/graphics/drawable/ColorDrawable.html>), DrawableContainer
(<https://developer.android.com/reference/android/graphics/drawable/DrawableContainer.html>), DrawableWrapper
(<https://developer.android.com/reference/android/graphics/drawable/DrawableWrapper.html>), GradientDrawable
(<https://developer.android.com/reference/android/graphics/drawable/GradientDrawable.html>), LayerDrawable
(<https://developer.android.com/reference/android/graphics/drawable/LayerDrawable.html>), NinePatchDrawable
(<https://developer.android.com/reference/android/graphics/drawable/NinePatchDrawable.html>), PictureDrawable
(<https://developer.android.com/reference/android/graphics/drawable/PictureDrawable.html>), ShapeDrawable
(<https://developer.android.com/reference/android/graphics/drawable/ShapeDrawable.html>), VectorDrawable
(<https://developer.android.com/reference/android/graphics/drawable/VectorDrawable.html>)

▼ (#)Known Indirect Subclasses

AnimatedStateListDrawable

(<https://developer.android.com/reference/android/graphics/drawable/AnimatedStateListDrawable.html>), AnimationDrawable
(<https://developer.android.com/reference/android/graphics/drawable/AnimationDrawable.html>), ClipDrawable
(<https://developer.android.com/reference/android/graphics/drawable/ClipDrawable.html>), InsetDrawable
(<https://developer.android.com/reference/android/graphics/drawable/InsetDrawable.html>), LevelListDrawable
(<https://developer.android.com/reference/android/graphics/drawable/LevelListDrawable.html>), PaintDrawable
(<https://developer.android.com/reference/android/graphics/drawable/PaintDrawable.html>), RippleDrawable
(<https://developer.android.com/reference/android/graphics/drawable/RippleDrawable.html>), RotateDrawable
(<https://developer.android.com/reference/android/graphics/drawable/RotateDrawable.html>), ScaleDrawable
(<https://developer.android.com/reference/android/graphics/drawable/ScaleDrawable.html>), StateListDrawable
(<https://developer.android.com/reference/android/graphics/drawable/StateListDrawable.html>), TransitionDrawable

This site uses cookies to store your preferences for site-specific language and display options.

OK

A Drawable is a general abstraction for "something that can be drawn." Most often you will deal with Drawable as the type of resource retrieved for drawing things to the screen; the Drawable class provides a generic API for dealing with an underlying visual resource that may take a variety of forms. Unlike a [View](https://developer.android.com/reference/android/view/View.html) (<https://developer.android.com/reference/android/view/View.html>), a Drawable does not have any facility to receive events or otherwise interact with the user.

In addition to simple drawing, Drawable provides a number of generic mechanisms for its client to interact with what is being drawn:

- The [setBounds\(Rect\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setBounds(android.graphics.Rect))

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setBounds\(android.graphics.Rect\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setBounds(android.graphics.Rect))) method **must** be called to tell the Drawable where it is drawn and how large it should be. All Drawables should respect the requested size, often simply by scaling their imagery. A client can find the preferred size for some Drawables with the [`getIntrinsicHeight\(\)`](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getIntrinsicHeight())

([`getIntrinsicHeight\(\)`](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getIntrinsicHeight())) and [`getIntrinsicWidth\(\)`](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getIntrinsicWidth())

([`getIntrinsicWidth\(\)`](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getIntrinsicWidth())) methods.

- The [getPadding\(Rect\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getPadding(android.graphics.Rect))

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getPadding\(android.graphics.Rect\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getPadding(android.graphics.Rect))) method can return from some Drawables information about how to frame content that is placed inside of them. For example, a Drawable that is intended to be the frame for a button widget would need to return padding that correctly places the label inside of itself.

- The [`setState\(int\[\]\)`](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setState(int[])) ([`https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setState\(int\[\]\)`](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setState(int[])))

method allows the client to tell the Drawable in which state it is to be drawn, such as "focused", "selected", etc. Some drawables may modify their imagery based on the selected state.

- The [`setLevel\(int\)`](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setLevel(int)) ([`https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setLevel\(int\)`](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setLevel(int)))

method allows the client to supply a single continuous controller that can modify the Drawable is displayed, such as a battery level or progress level. Some drawables may modify their imagery based on the current level.

- A Drawable can perform animations by calling back to its client through the [`Drawable.Callback`](https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback)

([`https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html`](https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html)) interface. All clients should support this interface (via [`setCallback\(Drawable.Callback\)`](https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback))

([`https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setCallback\(android.graphics.drawable.Drawable.Callback\)`](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setCallback(android.graphics.drawable.Drawable.Callback))) so that animations will work. A simple way to do this is through the system facilities such as [`setBackground\(Drawable\)`](https://developer.android.com/reference/android/view/View.html#setBackground(android.graphics.drawable.Drawable))

([`https://developer.android.com/reference/android/view/View.html#setBackground\(android.graphics.drawable.Drawable\)`](https://developer.android.com/reference/android/view/View.html#setBackground(android.graphics.drawable.Drawable))) and [`ImageView`](https://developer.android.com/reference/android/widget/ImageView.html) ([`https://developer.android.com/reference/android/widget/ImageView.html`](https://developer.android.com/reference/android/widget/ImageView.html)).

This site uses cookies to store your preferences for site-specific language and display options.

OK

- **Nine Patch**: an extension to the PNG format allows it to specify information about how to stretch it and place things inside of it.
- **Vector**: a drawable defined in an XML file as a set of points, lines, and curves along with its associated color information. This type of drawable can be scaled without loss of display quality.
- **Shape**: contains simple drawing commands instead of a raw bitmap, allowing it to resize better in some cases.
- **Layers**: a compound drawable, which draws multiple underlying drawables on top of each other.
- **States**: a compound drawable that selects one of a set of drawables based on its state.
- **Levels**: a compound drawable that selects one of a set of drawables based on its level.
- **Scale**: a compound drawable with a single child drawable, whose overall size is modified based on the current level.

Custom drawables

All versions of Android allow the `Drawable` class to be extended and used at run time in place of framework-provided drawable classes. Starting in [API 24](https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#N) (https://developer.android.com/reference/android/os/Build.VERSION_CODES.html#N), custom drawables classes may also be used in XML.

Note: Custom drawable classes are only accessible from within your application package. Other applications will not be able to load them.

At a minimum, custom drawable classes must implement the abstract methods on `Drawable` and should override the `draw(Canvas)` ([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#draw\(android.graphics.Canvas\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#draw(android.graphics.Canvas))) method to draw content.

Custom drawables classes may be used in XML in multiple ways:

- Using the fully-qualified class name as the XML element name. For this method, the custom drawable class must be a public top-level class.

```
<com.myapp.MyCustomDrawable xmlns:android="http://schemas.android.com/apk/res/android"  
    android:color="#ffff0000" />
```

- Using `drawable` as the XML element name and specifying the fully-qualified class name from the `class` attribute. This method may be used for both public top-level classes and public static inner classes.

```
<drawable xmlns:android="http://schemas.android.com/apk/res/android"  
    class="com.myapp.MyTopLevelClass$InnerCustomDrawable"  
    android:color="#ffff0000" />
```

This site uses cookies to store your preferences for site-specific language and display options.

OK

For more information about how to use drawables, read the [Canvas and Drawables](#) (<https://developer.android.com/guide/topics/graphics/2d-graphics.html>) developer guide. For information and examples of creating drawable resources (XML or bitmap files that can be loaded in code), read the [Drawable Resources](#) (<https://developer.android.com/guide/topics/resources/drawable-resource.html>) document.

Summary

Nested classes	
interface	Drawable.Callback (https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html) Implement this interface if you want to create an animated drawable that extends Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html).
class	Drawable.ConstantState (https://developer.android.com/reference/android/graphics/drawable/Drawable.ConstantState.html) This abstract class is used by Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html)s to store shared constant state and data between Drawables.

Public constructors
Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#Drawable())()

Public methods	
void	applyTheme (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#applyTheme(android.content.res.Resources.Theme)) Applies the specified theme.
boolean	canApplyTheme (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#canApplyTheme())
void	clearColorFilter (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#clearColorFilter()) Removes the color filter from this drawable.
final Rect (https://developer.android.com/reference/android/graphics/Rect.html)	copyBounds (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#copyBounds(android.graphics.Rect)) Return a copy of the draw

This site uses cookies to store your preferences for site-specific language and display options.

OK

<pre>static Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html)</pre>	<code>createFromPath</code> (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#method_detail) Create a drawable from fi
<pre>static Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html)</pre>	<code>createFromResourceStre</code> (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#method_detail) <code>android.util.TypedValue</code> , java (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#method_detail) <code>BitmapFactory.Options</code> Create a drawable from a
<pre>static Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html)</pre>	<code>createFromResourceStre</code> (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#method_detail) <code>android.util.TypedValue</code> , java (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#method_detail) <code>TypedValue</code> (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#method_detail) Create a drawable from a
<pre>static Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html)</pre>	<code>createFromStream</code> (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#method_detail) <code>InputStream</code> (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#method_detail) <code>srcName</code> Create a drawable from a
<pre>static Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html)</pre>	<code>createFromXml</code> (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#method_detail) <code>org.xmlpull.v1.XmlPullParser</code> (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#method_detail) Create a drawable from a
<pre>static Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html)</pre>	<code>createFromXml</code> (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#method_detail) <code>org.xmlpull.v1.XmlPullParser</code> <code>XmlPullParser</code> (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#method_detail) Create a drawable from a (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#method_detail)
<pre>static Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html)</pre>	<code>createFromXmlInner</code> (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#method_detail) <code>org.xmlpull.v1.XmlPullParser</code> (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#method_detail) (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#method_detail)

This site uses cookies to store your preferences for site-specific language and display options.

OK

This site uses cookies to store your preferences for site-specific language and display options.

OK

		Returns the resolved layout.
final int	getLevel (https://developer.android.com/reference/android/graphics/Drawable.html#method_detail)	Retrieve the current level.
int	getMinimumHeight (https://developer.android.com/reference/android/graphics/Drawable.html#method_detail)	Returns the minimum height.
int	getMinimumWidth (https://developer.android.com/reference/android/graphics/Drawable.html#method_detail)	Returns the minimum width.
abstract int	getOpacity (https://developer.android.com/reference/android/graphics/Drawable.html#method_detail)	Return the opacity/transparency level.
void	getOutline (https://developer.android.com/reference/android/graphics/Drawable.html#method_detail)	(https://developer.android.com/reference/android/graphics/Outline.html) Called to get the drawable's outline.
boolean	getPadding (https://developer.android.com/reference/android/graphics/Drawable.html#method_detail)	(https://developer.android.com/reference/android/graphics/Region.html) Return in padding the insets.
int[]	getState (https://developer.android.com/reference/android/graphics/Drawable.html#method_detail)	Describes the current state. (https://developer.android.com/reference/android/graphics/Drawable.State.html) (https://developer.android.com/reference/android/graphics/Drawable.StateSet.html)
Region (https://developer.android.com/reference/android/graphics/Region.html)	getTransparentRegion (https://developer.android.com/reference/android/graphics/Drawable.html#method_detail)	Returns a Region representing the transparent area.
void	inflate (https://developer.android.com/reference/android/graphics/Drawable.html#method_detail)	android.util.AttributeSet, java.lang.String) XmlPullParser (https://developer.android.com/reference/java/lang/String.html) (https://developer.android.com/reference/java/lang/String.html) (https://developer.android.com/reference/java/lang/String.html) Inflate this Drawable from XML.
void	inflate (https://developer.android.com/reference/android/graphics/Drawable.html#method_detail)	android.util.AttributeSet)) (https://developer.android.com/reference/java/lang/String.html) (https://developer.android.com/reference/java/lang/String.html) Inflate this Drawable from XML.
void	invalidateSelf (https://developer.android.com/reference/android/graphics/Drawable.html#method_detail)	Use the current Drawable .

This site uses cookies to store your preferences for site-specific language and display options.

OK

		Tells if this Drawable will
boolean		isFilterBitmap (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#isFilterBitmap())
boolean		isStateful (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#isStateful())
final boolean		isVisible (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#isVisible())
void		jumpToCurrentState (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#jumpToCurrentState()) If this Drawable does tran
Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html)		mutate (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#mutate()) Make this drawable muta
boolean		onLayoutDirectionChange (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#onLayoutDirectionChange(int)) Called when the drawab
static int		resolveOpacity (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#resolveOpacity(int)) Return the appropriate op
void		scheduleSelf (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#scheduleSelf(Runnable, long)) Use the current Drawable Drawable scheduled.
abstract void		setAlpha (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setAlpha(int)) Specify an alpha value for
void		setAutoMirrored (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setAutoMirrored(boolean)) Set whether this Drawab
void		setBounds (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setBounds(int left, int top, int right, int bottom)) Specify a bounding recta
void		setBounds (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setBounds(int left, int top, int right, int bottom)) Specify a bounding recta
final void		setCallback (https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#setCallback(android.graphics.drawable.Drawable.Callback)) (Drawable.Callback (htt
void		setChangingConfiguration (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setChangingConfiguration(int)) Bind a Drawable.Callback Bind a Drawable.Callback

This site uses cookies to store your preferences for site-specific language and display options.

OK

	Specify a color and Porter-Duff mode for this Drawable.
abstract void setColorFilter (https://developer.android.com/reference/android/graphics/Drawable.html#setColorFilter(int, PorterDuff.Mode))	Specify an optional color filter for this Drawable.
void setDither (https://developer.android.com/reference/android/graphics/Drawable.html#setDither(boolean))	<i>This method was deprecated in API level 11.</i>
void setFilterBitmap (https://developer.android.com/reference/android/graphics/Drawable.html#setFilterBitmap(Bitmap))	Set to true to have the drawables be filtered.
void setHotspot (https://developer.android.com/reference/android/graphics/Drawable.html#setHotspot(float, float, float, float))	Specifies the hotspot's location.
void setHotspotBounds (https://developer.android.com/reference/android/graphics/Drawable.html#setHotspotBounds(int, int, int, int))	Sets the bounds to which the hotspot is limited.
final boolean setLayoutDirection (https://developer.android.com/reference/android/graphics/Drawable.html#setLayoutDirection(int))	Set the layout direction for this Drawable.
final boolean setLevel (https://developer.android.com/reference/android/graphics/Drawable.html#setLevel(int))	Specify the level for the drawables.
boolean setState (<a data-kind="parent" data-rs="2" href="https://developer.android.com/reference/android/graphics/Drawable.html#setState(int...)">https://developer.android.com/reference/android/graphics/Drawable.html#setState(int...))	Specify a set of states for this Drawable.
void setTint (<a data-kind="ghost" href="https://developer.android.com/reference/android/graphics/Drawable.html#setTint(int)">)	Specifies tint color for this Drawable.
void setTintList (<a data-kind="parent" data-rs="2" href="https://developer.android.com/reference/android/graphics/Drawable.html#setTintList(ColorStateList)">https://developer.android.com/reference/android/graphics/Drawable.html#setTintList(ColorStateList))	ColorStateList (<a data-kind="parent" data-rs="2" href="https://developer.android.com/reference/android/color/ColorStateList.html">https://developer.android.com/reference/android/color/ColorStateList.html)
	Specifies tint color for this Drawable.
void setTintMode (<a data-kind="parent" data-rs="2" href="https://developer.android.com/reference/android/graphics/Drawable.html#setTintMode(PorterDuff.Mode)">https://developer.android.com/reference/android/graphics/Drawable.html#setTintMode(PorterDuff.Mode))	PorterDuff.Mode (<a data-kind="parent" data-rs="2" href="https://developer.android.com/reference/android/graphics/PorterDuff.Mode.html">https://developer.android.com/reference/android/graphics/PorterDuff.Mode.html)
	Specifies a tint blending mode.
boolean setVisible (<a data-kind="parent" data-rs="2" href="https://developer.android.com/reference/android/graphics/Drawable.html#setVisible(boolean, boolean)">https://developer.android.com/reference/android/graphics/Drawable.html#setVisible(boolean, boolean))	restart)
	Set whether this Drawable is visible.
void unscheduleSelf (<a data-kind="parent" data-rs="2" href="https://developer.android.com/reference/android/graphics/Drawable.html#unscheduleSelf(Runnable)">https://developer.android.com/reference/android/graphics/Drawable.html#unscheduleSelf(Runnable))	(https://developer.android.com/reference/android/graphics/Drawable.html#unscheduleSelf(Runnable))
	Use the current Drawable .

This site uses cookies to store your preferences for site-specific language and display options.

OK

Protected methods

void	onBoundsChange (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#onBoundsChange(android.graphics.Rect)) Override this in your subclass to change appearance if you vary based on the bounds.
boolean	onLevelChange (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#onLevelChange(int)) Override this in your subclass to change appearance if you vary based on level.
boolean	onStateChange (https://developer.android.com/reference/android/graphics/drawable/Drawable.html#onStateChange(int[])) Override this in your subclass to change appearance if you recognize the specified state.

Inherited methods

▼ (#)From class `java.lang.Object` (<https://developer.android.com/reference/java/lang/Object.html>)

Public constructors

Drawable

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Drawable ()`

Public methods

applyTheme

added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void applyTheme (Resources.Theme t)` (<https://developer.android.com/reference/android/content/res/Resources.Theme.html>)

Applies the specified theme to this Drawable and its children.

This site uses cookies to store your preferences for site-specific language and display options.

OK

This value must never be **null**.

canApplyTheme

added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`boolean canApplyTheme ()`

Returns

boolean

clearColorFilter

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void clearColorFilter ()`

Removes the color filter for this drawable.

copyBounds

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Rect (https://developer.android.com/reference/android/graphics/Rect.html) copyBounds ()`

Return a copy of the drawable's bounds in a new Rect. This returns the same values as `getBounds()`, but the returned object is guaranteed to not be changed later by the drawable (i.e. it retains no reference to this rect). If the caller already has a Rect allocated, call `copyBounds(rect)`.

Returns

Rect

[\(<https://developer.android.com/reference/android/graphics/Rect.html>\)](https://developer.android.com/reference/android/graphics/Rect.html)

A copy of the drawable's bounds

This value will never be **null**.

copyBounds

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void copyBounds (Rect (https://developer.android.com/reference/android/graphics/Rect.html) bounds)`

This site uses cookies to store your preferences for site-specific language and display options.

OK

Parameters

bounds	Rect: Rect to receive the drawable's bounds (allocated by the caller). This value must never be null.
--------	--

createFromPath

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)Drawable (<https://developer.android.com/reference/android/graphics/Drawable.html>) createFromPath (String (<https://developer.android.com/reference/java/lang/String.html>))

Create a drawable from file path name.

Parameters

pathName	String
----------	--------

Returns

Drawable	This value may be null. (https://developer.android.com/reference/android/graphics/Drawable.html)
----------	--

createFromResourceStream

added in API level 5 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)Drawable (<https://developer.android.com/reference/android/graphics/Drawable.html>) createFromResourceStream (Resources res, TypedValue value, InputStream is, String srcName, BitmapFactory.Options options)

Create a drawable from an inputstream, using the given resources and value to determine density information.

Parameters

res	Resources
value	TypedValue
is	InputStream

This site uses cookies to store your preferences for site-specific language and display options.

OK

Returns

Drawable

(<https://developer.android.com/reference/android/graphics/drawable/Drawable.html>)

createFromResourceStream

added in API level 4 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

Drawable (<https://developer.android.com/reference/android/graphics/drawable/Drawable.html>) `createFromResourceStream (Resources res, TypedValue value, InputStream is, String srcName)`

Create a drawable from an inputstream, using the given resources and value to determine density information.

Parameters

<code>res</code>	Resources
<code>value</code>	TypedValue
<code>is</code>	InputStream
<code>srcName</code>	String

Returns

Drawable

(<https://developer.android.com/reference/android/graphics/drawable/Drawable.html>)

createFromStream

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

Drawable (<https://developer.android.com/reference/android/graphics/drawable/Drawable.html>) `createFromStream (InputStream is, String srcName)`

Create a drawable from an inputstream

Parameters

<code>is</code>	InputStream
-----------------	-------------

This site uses cookies to store your preferences for site-specific language and display options.

OK

Returns

Drawable

(<https://developer.android.com/reference/android/graphics/Drawable.html>)

createFromXml

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

Drawable (<https://developer.android.com/reference/android/graphics/Drawable.html>) createFromXml (Resources (<https://developer.android.com/reference/org/xmlpull/v1/XmlPullParser.html>) parser)

Create a drawable from an XML document. For more information on how to create resources in XML, see Drawable Resources (<https://developer.android.com/guide/topics/resources/drawable-resource.html>).

Parameters

r

Resources

This value must never be **null**.

parser

XmlPullParser

This value must never be **null**.

Returns

Drawable

This value will never be **null**.

(<https://developer.android.com/reference/android/graphics/Drawable.html>)

Throws

XmlPullParserException

(<https://developer.android.com/reference/org/xmlpull/v1/XmlPullParserException.html>)

IOException (<https://developer.android.com/reference/java/io/IOException.html>)

createFromXml

added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

Drawable (<https://developer.android.com/reference/android/graphics/Drawable.html>) createFromXml (Resources (<https://developer.android.com/reference/org/xmlpull/v1/XmlPullParser.html>) parser, Resources.Theme (<https://developer.android.com/reference/android/content/res/Resources.Theme.html>) theme)

This site uses cookies to store your preferences for site-specific language and display options.

OK

Create a drawable from an XML document using an optional [Resources.Theme](#) (<https://developer.android.com/reference/android/content/res/Resources.Theme.html>). For more information on how to create resources in XML, see [Drawable Resources](#) (<https://developer.android.com/guide/topics/resources/drawable-resource.html>).

Parameters

<code>r</code>	Resources
	This value must never be <code>null</code> .
<code>parser</code>	XmlPullParser
	This value must never be <code>null</code> .
<code>theme</code>	Resources.Theme
	This value may be <code>null</code> .

Returns

Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html)	This value will never be <code>null</code> .
---	--

Throws

XmlPullParserException (https://developer.android.com/reference/org/xmlpull/v1/XmlPullParserException.html)	
IOException (https://developer.android.com/reference/java/io/IOException.html)	

createFromXmlInner

added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

[Drawable](#) (<https://developer.android.com/reference/android/graphics/drawable/Drawable.html>) `createFromXmlInner` ([Resources](#) [XmlPullParser](#) (<https://developer.android.com/reference/org/xmlpull/v1/XmlPullParser.html>) `parser`, [AttributeSet](#) (<https://developer.android.com/reference/android/util/AttributeSet.html>) `attrs`, [Resources.Theme](#) (<https://developer.android.com/reference/android/content/res/Resources.Theme.html>) `theme`

Create a drawable from inside an XML document using an optional [Resources.Theme](#) (<https://developer.android.com/reference/android/content/res/Resources.Theme.html>). Called on a parser positioned at a tag in an XML document, tries to create a Drawable from that tag. Returns `null` if the tag is not a valid drawable.

Parameters

<code>r</code>	Resources
----------------	---------------------------

This site uses cookies to store your preferences for site-specific language and display options.

OK

	This value must never be <code>null</code> .
<code>attrs</code>	<code>AttributeSet</code> This value must never be <code>null</code> .
<code>theme</code>	<code>Resources.Theme</code> This value may be <code>null</code> .

Returns	
<code>Drawable</code> (https://developer.android.com/reference/android/graphics/Drawable.html)	This value will never be <code>null</code> .

Throws	
<code>XmlPullParserException</code> (https://developer.android.com/reference/org/xmlpull/v1/XmlPullParserException.html)	
<code>IOException</code> (https://developer.android.com/reference/java/io/IOException.html)	

createFromXmlInner

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Drawable` (<https://developer.android.com/reference/android/graphics/Drawable.html>) `createFromXmlInner` (`Resources`
..... `XmlPullParser` (<https://developer.android.com/reference/org/xmlpull/v1/XmlPullParser.html>) `parser`,
..... `AttributeSet` (<https://developer.android.com/reference/android/util/AttributeSet.html>) `attrs`)

Create from inside an XML document. Called on a parser positioned at a tag in an XML document, tries to create a Drawable from that tag. Returns `null` if the tag is not a valid drawable.

Parameters	
<code>r</code>	<code>Resources</code> This value must never be <code>null</code> .
<code>parser</code>	<code>XmlPullParser</code> This value must never be <code>null</code> .
<code>attrs</code>	<code>AttributeSet</code> This value must never be <code>null</code> .

Returns	
---------	--

This site uses cookies to store your preferences for site-specific language and display options.

OK

Throws	
XmlPullParserException (https://developer.android.com/reference/org/xmlpull/v1/XmlPullParserException.html)	
IOException (https://developer.android.com/reference/java/io/IOException.html)	

draw

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void draw (Canvas (https://developer.android.com/reference/android/graphics/Canvas.html) canvas)`

Draw in its bounds (set via `setBounds`) respecting optional effects such as alpha (set via `setAlpha`) and color filter (set via `setColorFilter`).

Parameters	
canvas	Canvas : The canvas to draw into This value must never be <code>null</code> .

getAlpha

added in API level 19 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`int getAlpha ()`

Gets the current alpha value for the drawable. 0 means fully transparent, 255 means fully opaque. This method is implemented by `Drawable` subclasses and the value returned is specific to how that class treats alpha. The default return value is 255 if the class does not override this method to return a value specific to its use of alpha.

Returns	
int	Value is 255 or less.

getBounds

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Rect (https://developer.android.com/reference/android/graphics/Rect.html) getBounds ()`

This site uses cookies to store your preferences for site-specific language and display options.

OK

instead. You should also not change the object returned by this method as it may be the same object stored in the drawable.

Returns

Rect

(<https://developer.android.com/reference/android/graphics/Rect.html>)

The bounds of the drawable (which may change later, so caller beware). DO NOT ALTER the returned object as it may change the stored bounds of this drawable.

This value will never be `null`.

See also:

`copyBounds()` ([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#copyBounds\(\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#copyBounds()))

`copyBounds(android.graphics.Rect)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#copyBounds\(android.graphics.Rect\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#copyBounds(android.graphics.Rect)))

getCallback

added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Drawable.Callback` (<https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html>) `getCallback()`

Return the current `Drawable.Callback`

(<https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html>) implementation attached to this `Drawable`.

Returns

Drawable.Callback

(<https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html>)

A `Drawable.Callback`

(<https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html>) instance or `null` if no callback is attached.

See also:

`setCallback(android.graphics.drawable.Drawable.Callback)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setCallback\(android.graphics.drawable.Drawable.Callback\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setCallback(android.graphics.drawable.Drawable.Callback)))

This site uses cookies to store your preferences for site-specific language and display options.

OK

```
int getChangingConfigurations ()
```

Return a mask of the configuration parameters for which this drawable may change, requiring that it be re-created. The default implementation returns whatever was provided through `setChangingConfigurations(int)` ([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setChangingConfigurations\(int\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setChangingConfigurations(int))) or 0 by default. Subclasses may extend this to or in the changing configurations of any other drawables they hold.

Returns

<code>int</code>	Returns a mask of the changing configuration parameters, as defined by <code>ActivityInfo</code> (https://developer.android.com/reference/android/content/pm/ActivityInfo.html).
------------------	---

See also:

`ActivityInfo` (<https://developer.android.com/reference/android/content/pm/ActivityInfo.html>)

getColorFilter

added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`ColorFilter` (<https://developer.android.com/reference/android/graphics/ColorFilter.html>) `getColorFilter ()`

Returns the current color filter, or `null` if none set.

Returns

<code>ColorFilter</code> (https://developer.android.com/reference/android/graphics/ColorFilter.html)	the current color filter, or <code>null</code> if none set
---	--

getConstantState

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Drawable.ConstantState` (<https://developer.android.com/reference/android/graphics/drawable/Drawable.ConstantState.html>) `getConstantState ()`

Return a `Drawable.ConstantState`

(<https://developer.android.com/reference/android/graphics/drawable/Drawable.ConstantState.html>) instance that holds the shared state of this Drawable.

Returns

<code>Drawable.ConstantState</code>	The ConstantState
-------------------------------------	-------------------

This site uses cookies to store your preferences for site-specific language and display options.

OK

This value may be
null.

See also:

`Drawable.ConstantState` (<https://developer.android.com/reference/android/graphics/drawable/Drawable.ConstantState.html>)

`mutate()` ([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#mutate\(\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#mutate()))

getCurrent

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Drawable` (<https://developer.android.com/reference/android/graphics/drawable/Drawable.html>) `getCurrent ()`

Returns

Drawable

(<https://developer.android.com/reference/android/graphics/drawable/Drawable.html>)

The current drawable that will be used is just the drawable itself. For drawables (<https://developer.android.com/reference/android/graphics/drawable/InsetDrawable.html>) and `LevelListDrawable` (<https://developer.android.com/reference/android/graphics/drawable/LevelListDrawable.html>) this will be the child drawable currently. This value will never be **null**.

getDirtyBounds

added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Rect` (<https://developer.android.com/reference/android/graphics/Rect.html>) `getDirtyBounds ()`

Return the drawable's dirty bounds `Rect`. Note: for efficiency, the returned object may be the same object stored in the drawable (though this is not guaranteed).

By default, this returns the full drawable bounds. Custom drawables may override this method to perform more precise invalidation.

Returns

Rect

(<https://developer.android.com/reference/android/graphics/Rect.html>)

The dirty bounds of this drawable
This value will never be **null**.

This site uses cookies to store your preferences for site-specific language and display options.

OK

getHotspotBounds

added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void getHotspotBounds (Rect (https://developer.android.com/reference/android/graphics/Rect.html) outRect)
```

Populates `outRect` with the hotspot bounds.

Parameters

<code>outRect</code>	<code>Rect</code> : the rect to populate with the hotspot bounds This value must never be <code>null</code> .
----------------------	--

See also:

```
setHotspotBounds(int, int, int, int)
```

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setHotspotBounds\(int, int, int, int\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setHotspotBounds(int, int, int, int)))

getIntrinsicHeight

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
int getIntrinsicHeight ()
```

Returns the drawable's intrinsic height.

Intrinsic height is the height at which the drawable would like to be laid out, including any inherent padding. If the drawable has no intrinsic height, such as a solid color, this method returns -1.

Returns

<code>int</code>	the intrinsic height, or -1 if no intrinsic height
------------------	--

getIntrinsicWidth

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
int getIntrinsicWidth ()
```

Returns the drawable's intrinsic width.

Intrinsic width is the width at which the drawable would like to be laid out, including any inherent padding. If the drawable has no intrinsic width, such as a solid color, this method returns -1.

Returns

This site uses cookies to store your preferences for site-specific language and display options.

OK

getLayoutDirection

added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
int getLayoutDirection ()
```

Returns the resolved layout direction for this Drawable.

Returns

int

One of `LAYOUT_DIRECTION_LTR`

(https://developer.android.com/reference/android/view/View.html#LAYOUT_DIRECTION_LTR), `LAYOUT_DIRECTION_RTL`

(https://developer.android.com/reference/android/view/View.html#LAYOUT_DIRECTION_RTL)

See also:

`setLayoutDirection(int)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setLayoutDirection\(int\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setLayoutDirection(int)))

getLevel

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
int getLevel ()
```

Retrieve the current level.

Returns

int

int Current level, from 0 (minimum) to 10000 (maximum).

Value is 10000 or less.

getMinimumHeight

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
int getMinimumHeight ()
```

Returns the minimum height suggested by this Drawable. If a View uses this Drawable as a background, it is suggested that the View use at least this value for its height. (There will be some scenarios where this will not be possible.) This value should INCLUDE any padding.

Returns

int

The minimum height suggested by this Drawable. If this Drawable doesn't have a suggested minimum

This site uses cookies to store your preferences for site-specific language and display options.

OK

getMinimumWidth

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
int getMinimumWidth ()
```

Returns the minimum width suggested by this Drawable. If a View uses this Drawable as a background, it is suggested that the View use at least this value for its width. (There will be some scenarios where this will not be possible.) This value should INCLUDE any padding.

Returns	
int	The minimum width suggested by this Drawable. If this Drawable doesn't have a suggested minimum width, 0 is returned.

getOpacity

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
int getOpacity ()
```

Return the opacity/transparency of this Drawable. The returned value is one of the abstract format constants in [PixelFormat](https://developer.android.com/reference/android/graphics/PixelFormat.html) (<https://developer.android.com/reference/android/graphics/PixelFormat.html>): UNKNOWN (<https://developer.android.com/reference/android/graphics/PixelFormat.html#UNKNOWN>), TRANSLUCENT (<https://developer.android.com/reference/android/graphics/PixelFormat.html#TRANSLUCENT>), TRANSPARENT (<https://developer.android.com/reference/android/graphics/PixelFormat.html#TRANSPARENT>), or OPAQUE (<https://developer.android.com/reference/android/graphics/PixelFormat.html#OPAQUE>).

An OPAQUE drawable is one that draws all content within its bounds, completely covering anything behind the drawable. A TRANSPARENT drawable is one that draws nothing within its bounds, allowing everything behind it to show through. A TRANSLUCENT drawable is a drawable in any other state, where the drawable will draw some, but not all, of the content within its bounds and at least some content behind the drawable will be visible. If the visibility of the drawable's contents cannot be determined, the safest/best return value is TRANSLUCENT.

Generally a Drawable should be as conservative as possible with the value it returns. For example, if it contains multiple child drawables and only shows one of them at a time, if only one of the children is TRANSLUCENT and the others are OPAQUE then TRANSLUCENT should be returned. You can use the method [resolveOpacity\(int, int\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#resolveOpacity(int, int)) ([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#resolveOpacity\(int, int\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#resolveOpacity(int, int))) to perform a standard reduction of two opacities to the appropriate single output.

Note that the returned value does not necessarily take into account a custom alpha or color filter that has been applied by the client through the [setAlpha\(int\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setAlpha(int))

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setAlpha\(int\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setAlpha(int))) OR

This site uses cookies to store your preferences for site-specific language and display options.

OK

(<https://developer.android.com/reference/android/graphics/drawable/BitmapDrawable.html>), **ColorDrawable** (<https://developer.android.com/reference/android/graphics/drawable/ColorDrawable.html>), and **GradientDrawable** (<https://developer.android.com/reference/android/graphics/drawable/GradientDrawable.html>), do account for the value of **setAlpha(int)** ([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setAlpha\(int\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setAlpha(int))), but the general behavior is dependent upon the implementation of the subclass.

Returns	
int	int The opacity class of the Drawable.

See also:

PixelFormat (<https://developer.android.com/reference/android/graphics/PixelFormat.html>)

getOutline

added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void getOutline (Outline (https://developer.android.com/reference/android/graphics/Outline.html) outline)`

Called to get the drawable to populate the Outline that defines its drawing area.

This method is called by the default **ViewOutlineProvider**

(<https://developer.android.com/reference/android/view/ViewOutlineProvider.html>) to define the outline of the View.

The default behavior defines the outline to be the bounding rectangle of 0 alpha. Subclasses that wish to convey a different shape or alpha value must override this method.

Parameters	
outline	Outline This value must never be null .

See also:

setOutlineProvider(android.view.ViewOutlineProvider)

([https://developer.android.com/reference/android/view/View.html#setOutlineProvider\(android.view.ViewOutlineProvider\)](https://developer.android.com/reference/android/view/View.html#setOutlineProvider(android.view.ViewOutlineProvider)))

getPadding

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

This site uses cookies to store your preferences for site-specific language and display options.

OK

Return in padding the insets suggested by this Drawable for placing content inside the drawable's bounds. Positive values move toward the center of the Drawable (set Rect.inset).

Parameters

padding

Rect

This value must never be `null`.

Returns

boolean

true if this drawable actually has a padding, else false. When false is returned, the padding is always set to 0.

getState

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`int[] getState ()`

Describes the current state, as a union of primitive states, such as `state_focused`

(https://developer.android.com/reference/android/R.attr.html#state_focused), `state_selected`

(https://developer.android.com/reference/android/R.attr.html#state_selected), etc. Some drawables may modify their imagery based on the selected state.

Returns

int[]

An array of resource Ids describing the current state.

This value will never be `null`.

getTransparentRegion

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`Region (https://developer.android.com/reference/android/graphics/Region.html) getTransparentRegion ()`

Returns a Region representing the part of the Drawable that is completely transparent. This can be used to perform drawing operations, identifying which parts of the target will not change when rendering the Drawable. The default implementation returns null, indicating no transparent region; subclasses can optionally override this to return an actual Region if they want to supply this optimization information, but it is not required that they do so.

Returns

This site uses cookies to store your preferences for site-specific language and display options.

OK

holding the parts of the Drawable's bounds that are transparent.

inflate

added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void inflate (Resources (https://developer.android.com/reference/android/content/res/Resources.html) r,  
             XmlPullParser (https://developer.android.com/reference/org/xmlpull/v1/XmlPullParser.html) parser,  
             AttributeSet (https://developer.android.com/reference/android/util/AttributeSet.html) attrs,  
             Resources.Theme (https://developer.android.com/reference/android/content/res/Resources.Theme.html) theme
```

Inflate this Drawable from an XML resource optionally styled by a theme. This can't be called more than once for each Drawable. Note that framework may have called this once to create the Drawable instance from XML resource.

Parameters	
r	Resources : Resources used to resolve attribute values This value must never be null .
parser	XmlPullParser : XML parser from which to inflate this Drawable This value must never be null .
attrs	AttributeSet : Base set of attribute values This value must never be null .
theme	Resources.Theme : Theme to apply, may be null

Throws	
	XmlPullParserException
IOException (https://developer.android.com/reference/java/io/IOException.html) XmlPullParserException (https://developer.android.com/reference/org/xmlpull/v1/XmlPullParserException.html)	

inflate

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void inflate (Resources (https://developer.android.com/reference/android/content/res/Resources.html) r,  
             XmlPullParser (https://developer.android.com/reference/org/xmlpull/v1/XmlPullParser.html) parser,  
             AttributeSet (https://developer.android.com/reference/android/util/AttributeSet.html) attrs)
```

This site uses cookies to store your preferences for site-specific language and display options.

OK

Parameters

r	Resources This value must never be <code>null</code> .
parser	XmlPullParser This value must never be <code>null</code> .
attrs	AttributeSet This value must never be <code>null</code> .

Throws

XmlPullParserException (https://developer.android.com/reference/org/xmlpull/v1/XmlPullParserException.html)	
IOException (https://developer.android.com/reference/java/io/IOException.html)	

See also:

`inflate(Resources, XmlPullParser, AttributeSet, Theme)`
 ([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#inflate\(android.content.res.Resources, org.xmlpull.v1.XmlPullParser, android.util.AttributeSet, android.content.res.Theme\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#inflate(android.content.res.Resources, org.xmlpull.v1.XmlPullParser, android.util.AttributeSet, android.content.res.Theme)))

invalidateSelf

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void invalidateSelf ()`

Use the current `Drawable.Callback`

(<https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html>) implementation to have this `Drawable` redrawn. Does nothing if there is no `Callback` attached to the `Drawable`.

See also:

`invalidateDrawable(Drawable)`
 ([https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#invalidateDrawable\(android.graphics.drawable.Drawable\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#invalidateDrawable(android.graphics.drawable.Drawable)))

`getCallback()` ([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getCallback\(\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getCallback()))

`setCallback(android.graphics.drawable.Drawable.Callback)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setCallback\(android.graphics.drawable.Drawable.Callback\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setCallback(android.graphics.drawable.Drawable.Callback)))

This site uses cookies to store your preferences for site-specific language and display options.

OK

isAutoMirrored

added in API level 19 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`boolean isAutoMirrored ()`

Tells if this Drawable will be automatically mirrored when its layout direction is RTL right-to-left. See [LayoutDirection](#) (<https://developer.android.com/reference/android/util/LayoutDirection.html>).

Returns

<code>boolean</code>	boolean Returns true if this Drawable will be automatically mirrored.
----------------------	---

isFilterBitmap

added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`boolean isFilterBitmap ()`

Returns

<code>boolean</code>	whether this drawable filters its bitmaps
----------------------	---

See also:

[setFilterBitmap\(boolean\)](#)

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setFilterBitmap\(boolean\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setFilterBitmap(boolean)))

isStateful

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`boolean isStateful ()`

Indicates whether this drawable will change its appearance based on state. Clients can use this to determine whether it is necessary to calculate their state and call `setState`.

Returns

<code>boolean</code>	True if this drawable changes its appearance based on state, false otherwise.
----------------------	---

See also:

[setState\(int\[\]\)](#) ([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setState\(int\[\]\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setState(int[])))

This site uses cookies to store your preferences for site-specific language and display options.

OK

isVisible

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean isVisible ()
```

Returns

boolean	
---------	--

jumpToCurrentState

added in API level 11 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void jumpToCurrentState ()
```

If this Drawable does transition animations between states, ask that it immediately jump to the current state and skip any active animations.

mutate

added in API level 3 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
Drawable (https://developer.android.com/reference/android/graphics/drawable/Drawable.html) mutate ()
```

Make this drawable mutable. This operation cannot be reversed. A mutable drawable is guaranteed to not share its state with any other drawable. This is especially useful when you need to modify properties of drawables loaded from resources. By default, all drawables instances loaded from the same resource share a common state; if you modify the state of one instance, all the other instances will receive the same modification. Calling this method on a mutable Drawable will have no effect.

Returns

Drawable	This drawable.
----------	----------------

[\(https://developer.android.com/reference/android/graphics/drawable/Drawable.html\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html)

This value will never be **null**.

See also:

[Drawable.ConstantState](https://developer.android.com/reference/android/graphics/drawable/Drawable.ConstantState.html) (<https://developer.android.com/reference/android/graphics/drawable/Drawable.ConstantState.html>)

[getConstantState\(\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getConstantState()) ([\(https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getConstantState\(\)\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getConstantState()))

This site uses cookies to store your preferences for site-specific language and display options.

OK

```
boolean onLayoutDirectionChanged (int layoutDirection)
```

Called when the drawable's resolved layout direction changes.

Parameters

layoutDirection	int: the new resolved layout direction
-----------------	--

Returns

boolean	true if the layout direction change has caused the appearance of the drawable to change such that it needs to be re-drawn, false otherwise
---------	--

See also:

[setLayoutDirection\(int\)](#)

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setLayoutDirection\(int\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setLayoutDirection(int)))

resolveOpacity

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
int resolveOpacity (int op1,  
                   int op2)
```

Return the appropriate opacity value for two source opacities. If either is UNKNOWN, that is returned; else, if either is TRANSLUCENT, that is returned; else, if either is TRANSPARENT, that is returned; else, OPAQUE is returned.

This is to help in implementing [getOpacity\(\)](#)

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getOpacity\(\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getOpacity())).

Parameters

op1	int: One opacity value.
-----	-------------------------

op2	int: Another opacity value.
-----	-----------------------------

Returns

int	int The combined opacity value.
-----	---------------------------------

See also:

[getOpacity\(\)](#) ([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getOpacity\(\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getOpacity()))

This site uses cookies to store your preferences for site-specific language and display options.

OK

scheduleSelf

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void scheduleSelf (Runnable (https://developer.android.com/reference/java/lang/Runnable.html) what,
                  long when)
```

Use the current `Drawable.Callback`

(<https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html>) implementation to have this Drawable scheduled. Does nothing if there is no Callback attached to the Drawable.

Parameters	
<code>what</code>	<code>Runnable</code> : The action being scheduled. This value must never be <code>null</code> .
<code>when</code>	<code>long</code> : The time (in milliseconds) to run.

See also:

`scheduleDrawable(Drawable, Runnable, long)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#scheduleDrawable\(android.graphics.drawable.Drawable, java.lang.Runnable, long\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#scheduleDrawable(android.graphics.drawable.Drawable, java.lang.Runnable, long)))

setAlpha

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setAlpha (int alpha)
```

Specify an alpha value for the drawable. 0 means fully transparent, and 255 means fully opaque.

Parameters	
<code>alpha</code>	<code>int</code> Value is 255 or less.

setAutoMirrored

added in API level 19 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setAutoMirrored (boolean mirrored)
```

Set whether this Drawable is automatically mirrored when its layout direction is RTL (right-to left). See `LayoutDirection` (<https://developer.android.com/reference/android/util/LayoutDirection.html>).

This site uses cookies to store your preferences for site-specific language and display options.

OK

mirrored**boolean**: Set to true if the Drawable should be mirrored, false if not.

setBounds

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setBounds (int left,  
                int top,  
                int right,  
                int bottom)
```

Specify a bounding rectangle for the Drawable. This is where the drawable will draw when its draw() method is called.

Parameters	
left	int
top	int
right	int
bottom	int

setBounds

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setBounds (Rect (https://developer.android.com/reference/android/graphics/Rect.html) bounds)
```

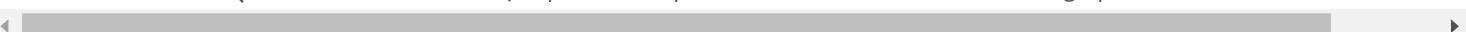
Specify a bounding rectangle for the Drawable. This is where the drawable will draw when its draw() method is called.

Parameters	
bounds	Rect This value must never be null .

setCallback

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setCallback (Drawable.Callback (https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html)
```



This site uses cookies to store your preferences for site-specific language and display options.

OK

Parameters

cb	<code>Drawable.Callback</code> : The client's Callback implementation. This value may be <code>null</code> .
----	---

See also:

`getCallback()` ([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getCallback\(\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getCallback()))

setChangingConfigurations

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void setChangingConfigurations (int configs)`

Set a mask of the configuration parameters for which this drawable may change, requiring that it be re-created.

Parameters

configs	<code>int</code> : A mask of the changing configuration parameters, as defined by <code>ActivityInfo</code> (https://developer.android.com/reference/android/content/pm/ActivityInfo.html).
---------	--

See also:

`ActivityInfo` (<https://developer.android.com/reference/android/content/pm/ActivityInfo.html>)

setColorFilter

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void setColorFilter (int color, PorterDuff.Mode mode)`

Specify a color and Porter-Duff mode to be the color filter for this drawable.

Convenience for `setColorFilter(ColorFilter)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setColorFilter\(android.graphics.ColorFilter\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setColorFilter(android.graphics.ColorFilter)))

which constructs a `PorterDuffColorFilter`

(<https://developer.android.com/reference/android/graphics/PorterDuffColorFilter.html>).

Note: Setting a color filter disables tint

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setTintList\(android.content.res.ColorStateList\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setTintList(android.content.res.ColorStateList)))

This site uses cookies to store your preferences for site-specific language and display options.

OK

color	int
mode	<p>PorterDuff.Mode</p> <p>This value must never be <code>null</code>.</p>

setColorFilter

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void setColorFilter (ColorFilter (https://developer.android.com/reference/android/graphics/ColorFilter.html) colorFilter)`

Specify an optional color filter for the drawable.

If a Drawable has a ColorFilter, each output pixel of the Drawable's drawing contents will be modified by the color filter before it is blended onto the render target of a Canvas.

Pass `null` to remove any existing color filter.

Note: Setting a non-`null` color filter disables `tint`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setTintList\(android.content.res.ColorStateList\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setTintList(android.content.res.ColorStateList))).

Parameters

`colorFilter` `ColorFilter`: The color filter to apply, or `null` to remove the existing color filter

setDither

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void setDither (boolean dither)`

This method was deprecated in API level 23.

This property is ignored.

Set to true to have the drawable dither its colors when drawn to a device with fewer than 8-bits per color component.

Parameters

`dither` `boolean`

This site uses cookies to store your preferences for site-specific language and display options.

OK

setFilterBitmap

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setFilterBitmap (boolean filter)
```

Set to true to have the drawable filter its bitmaps with bilinear sampling when they are scaled or rotated.

This can improve appearance when bitmaps are rotated. If the drawable does not use bitmaps, this call is ignored.

Parameters

filter	boolean
--------	---------

See also:

[isFilterBitmap\(\)](#) ([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#isFilterBitmap\(\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#isFilterBitmap()))

[setFilterBitmap\(boolean\)](#) ([https://developer.android.com/reference/android/graphics/Paint.html#setFilterBitmap\(boolean\)](https://developer.android.com/reference/android/graphics/Paint.html#setFilterBitmap(boolean)))

setHotspot

added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setHotspot (float x,  
                 float y)
```

Specifies the hotspot's location within the drawable.

Parameters

x	float: The X coordinate of the center of the hotspot
y	float: The Y coordinate of the center of the hotspot

setHotspotBounds

added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setHotspotBounds (int left,  
                      int top,  
                      int right,  
                      int bottom)
```

Sets the bounds to which the hotspot is constrained, if they should be different from the drawable bounds.

Parameters

This site uses cookies to store your preferences for site-specific language and display options.

OK

right	<code>int</code> : position in pixels of the right bound
bottom	<code>int</code> : position in pixels of the bottom bound

See also:

`getHotspotBounds(android.graphics.Rect)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getHotspotBounds\(android.graphics.Rect\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getHotspotBounds(android.graphics.Rect)))

setLayoutDirection

added in API level 23 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`boolean setLayoutDirection (int layoutDirection)`

Set the layout direction for this drawable. Should be a resolved layout direction, as the Drawable has no capacity to do the resolution on its own.

Parameters	
<code>layoutDirection</code>	<code>int</code> : the resolved layout direction for the drawable, either <code>LAYOUT_DIRECTION_LTR</code> (https://developer.android.com/reference/android/view/View.html#LAYOUT_DIRECTION_LTR) or <code>LAYOUT_DIRECTION_RTL</code> (https://developer.android.com/reference/android/view/View.html#LAYOUT_DIRECTION_RTL)

Returns	
<code>boolean</code>	<code>true</code> if the layout direction change has caused the appearance of the drawable to change such that it needs to be re-drawn, <code>false</code> otherwise

See also:

`getLayoutDirection()` ([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getLayoutDirection\(\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#getLayoutDirection()))

setLevel

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`boolean setLevel (int level)`

Specify the level for the drawable. This allows a drawable to vary its imagery based on a continuous controller, for example to show progress or volume level.

This site uses cookies to store your preferences for site-specific language and display options.

OK

to have itself redrawn, and true will be returned from this function.

Parameters

level	int : The new level, from 0 (minimum) to 10000 (maximum). Value is 10000 or less.
--------------	---

Returns

boolean	Returns true if this change in level has caused the appearance of the Drawable to change (hence requiring an invalidate), otherwise returns false.
----------------	--

setState

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`boolean setState (int[] stateSet)`

Specify a set of states for the drawable. These are use-case specific, so see the relevant documentation. As an example, the background for widgets like Button understand the following states: [**state_focused**

(https://developer.android.com/reference/android/R.attr.html#state_focused), **state_pressed**

(https://developer.android.com/reference/android/R.attr.html#state_pressed)].

If the new state you are supplying causes the appearance of the Drawable to change, then it is responsible for calling **invalidateSelf()** ([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#invalidateSelf\(\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#invalidateSelf())) in order to have itself redrawn, and true will be returned from this function.

Note: The Drawable holds a reference on to **stateSet** until a new state array is given to it, so you must not modify this array during that time.

Parameters

stateSet	int : The new set of states to be displayed. This value must never be null .
-----------------	---

Returns

boolean	Returns true if this change in state has caused the appearance of the Drawable to change (hence requiring an invalidate), otherwise returns false.
----------------	--

This site uses cookies to store your preferences for site-specific language and display options.

OK

```
void setTint (int tintColor)
```

Specifies tint color for this drawable.

A Drawable's drawing content will be blended together with its tint before it is drawn to the screen. This functions similarly to `setColorFilter(int, PorterDuff.Mode)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setColorFilter\(int, android.graphics.PorterDuff.Mode\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setColorFilter(int, android.graphics.PorterDuff.Mode))).

To clear the tint, pass `null` to `setTintList(ColorStateList)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setTintList\(android.content.res.ColorStateList\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setTintList(android.content.res.ColorStateList)))

Note: Setting a color filter via `setColorFilter(ColorFilter)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setColorFilter\(android.graphics.ColorFilter\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setColorFilter(android.graphics.ColorFilter))) or `setColorFilter(int, PorterDuff.Mode)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setColorFilter\(int, android.graphics.PorterDuff.Mode\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setColorFilter(int, android.graphics.PorterDuff.Mode))) overrides tint.

Parameters

<code>tintColor</code>	<code>int</code> : Color to use for tinting this drawable
------------------------	---

See also:

`setTintList(ColorStateList)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setTintList\(android.content.res.ColorStateList\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setTintList(android.content.res.ColorStateList)))

`setTintMode(PorterDuff.Mode)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setTintMode\(android.graphics.PoterDuff.Mode\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setTintMode(android.graphics.PoterDuff.Mode)))

setTintList

added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void setTintList (ColorStateList (https://developer.android.com/reference/android/content/res/ColorStateList.html) tint)
```

Specifies tint color for this drawable as a color state list.

A Drawable's drawing content will be blended together with its tint before it is drawn to the screen. This functions similarly to `setColorFilter(int, PorterDuff.Mode)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setColorFilter\(int, android.graphics.PorterDuff.Mode\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setColorFilter(int, android.graphics.PorterDuff.Mode))).

This site uses cookies to store your preferences for site-specific language and display options.

OK

) or `setColorFilter(int, PorterDuff.Mode)`
([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setColorFilter\(int, android.graphics.PorterDuff.Mode\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setColorFilter(int, android.graphics.PorterDuff.Mode))) overrides tint.

Parameters

<code>tint</code>	<code>ColorStateList</code> : Color state list to use for tinting this drawable, or <code>null</code> to clear the tint
-------------------	---

See also:

`setTint(int)` ([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setTint\(int\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setTint(int)))

`setTintMode(PorterDuff.Mode)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setTintMode\(android.graphics.PorterDuff.Mode\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setTintMode(android.graphics.PorterDuff.Mode)))

setTintMode

added in API level 21 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

`void setTintMode (PorterDuff.Mode` (<https://developer.android.com/reference/android/graphics/PorterDuff.Mode.html>) `tintMode`

Specifies a tint blending mode for this drawable.

Defines how this drawable's tint color should be blended into the drawable before it is drawn to screen. Default tint mode is `SRC_IN` (https://developer.android.com/reference/android/graphics/PorterDuff.Mode.html#SRC_IN).

Note: Setting a color filter via `setColorFilter(ColorFilter)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setColorFilter\(android.graphics.ColorFilter\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setColorFilter(android.graphics.ColorFilter)))

) or `setColorFilter(int, PorterDuff.Mode)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setColorFilter\(int, android.graphics.PorterDuff.Mode\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setColorFilter(int, android.graphics.PorterDuff.Mode))) overrides tint.

Parameters

<code>tintMode</code>	<code>PorterDuff.Mode</code> : A Porter-Duff blending mode
-----------------------	--

This value must never be `null`.

See also:

`setTint(int)` ([https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setTint\(int\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.html#setTint(int)))

`setTintList(ColorStateList)`

This site uses cookies to store your preferences for site-specific language and display options.

OK

setVisible

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
boolean setVisible (boolean visible,
                  boolean restart)
```

Set whether this Drawable is visible. This generally does not impact the Drawable's behavior, but is a hint that can be used by some Drawables, for example, to decide whether run animations.

Parameters	
visible	boolean: Set to true if visible, false if not.
restart	boolean: You can supply true here to force the drawable to behave as if it has just become visible, even if it had last been set visible. Used for example to force animations to restart.

Returns	
boolean	boolean Returns true if the new visibility is different than its previous state.

unscheduleSelf

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

```
void unscheduleSelf (Runnable (https://developer.android.com/reference/java/lang/Runnable.html) what)
```

Use the current `Drawable.Callback`

(<https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html>) implementation to have this Drawable unscheduled. Does nothing if there is no Callback attached to the Drawable.

Parameters	
what	Runnable: The runnable that you no longer want called. This value must never be null.

See also:

`unscheduleDrawable(Drawable, Runnable)`

([https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#unscheduleDrawable\(android.graphics.drawable.Drawable, java.lang.Runnable\)](https://developer.android.com/reference/android/graphics/drawable/Drawable.Callback.html#unscheduleDrawable(android.graphics.drawable.Drawable, java.lang.Runnable)))

This site uses cookies to store your preferences for site-specific language and display options.

OK

onBoundsChange

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

void onBoundsChange (Rect (<https://developer.android.com/reference/android/graphics/Rect.html>) bounds)

Override this in your subclass to change appearance if you vary based on the bounds.

Parameters

bounds	Rect
--------	------

onLevelChange

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

boolean onLevelChange (int level)

Override this in your subclass to change appearance if you vary based on level.

Parameters

level	int
-------	-----

Returns

boolean	Returns true if the level change has caused the appearance of the Drawable to change (that is, it needs to be drawn), else false if it looks the same and there is no need to redraw it since its last level.
---------	---

onStateChange

added in API level 1 (<https://developer.android.com/guide/topics/manifest/uses-sdk-element.html#ApiLevels>)

boolean onStateChange (int[] state)

Override this in your subclass to change appearance if you recognize the specified state.

Parameters

state	int
-------	-----

Returns

boolean	Returns true if the state change has caused the appearance of the Drawable to change (that is, it needs to be drawn), else false if it looks the same and there is no need to redraw it since its last state.
---------	---

This site uses cookies to store your preferences for site-specific language and display options.

OK



This site uses cookies to store your preferences for site-specific language and display options.

OK

