| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/GradientPaint.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/Frame.AccessibleAWTFrame.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Graphics.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/GradientPaint.html)    [**NO FRAMES**](http://docs.google.com/GradientPaint.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#lnxbz9) |

## **java.awt**

Class GradientPaint

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **java.awt.GradientPaint**

**All Implemented Interfaces:** [Paint](http://docs.google.com/java/awt/Paint.html), [Transparency](http://docs.google.com/java/awt/Transparency.html)

public class **GradientPaint**extends [Object](http://docs.google.com/java/lang/Object.html)implements [Paint](http://docs.google.com/java/awt/Paint.html)

The GradientPaint class provides a way to fill a [Shape](http://docs.google.com/java/awt/Shape.html) with a linear color gradient pattern. If [Point](http://docs.google.com/java/awt/Point.html) P1 with [Color](http://docs.google.com/java/awt/Color.html) C1 and Point P2 with Color C2 are specified in user space, the Color on the P1, P2 connecting line is proportionally changed from C1 to C2. Any point P not on the extended P1, P2 connecting line has the color of the point P' that is the perpendicular projection of P on the extended P1, P2 connecting line. Points on the extended line outside of the P1, P2 segment can be colored in one of two ways.

* If the gradient is cyclic then the points on the extended P1, P2 connecting line cycle back and forth between the colors C1 and C2.
* If the gradient is acyclic then points on the P1 side of the segment have the constant Color C1 while points on the P2 side have the constant Color C2.

**See Also:**[Paint](http://docs.google.com/java/awt/Paint.html), [Graphics2D.setPaint(java.awt.Paint)](http://docs.google.com/java/awt/Graphics2D.html#setPaint(java.awt.Paint))

| **Field Summary** | |
| --- | --- |

| **Fields inherited from interface java.awt.**[**Transparency**](http://docs.google.com/java/awt/Transparency.html) |
| --- |
| [BITMASK](http://docs.google.com/java/awt/Transparency.html#BITMASK), [OPAQUE](http://docs.google.com/java/awt/Transparency.html#OPAQUE), [TRANSLUCENT](http://docs.google.com/java/awt/Transparency.html#TRANSLUCENT) |

| **Constructor Summary** | |
| --- | --- |
| [**GradientPaint**](http://docs.google.com/java/awt/GradientPaint.html#GradientPaint(float,%20float,%20java.awt.Color,%20float,%20float,%20java.awt.Color))(float x1, float y1, [Color](http://docs.google.com/java/awt/Color.html) color1, float x2, float y2, [Color](http://docs.google.com/java/awt/Color.html) color2)            Constructs a simple acyclic GradientPaint object. |
| [**GradientPaint**](http://docs.google.com/java/awt/GradientPaint.html#GradientPaint(float,%20float,%20java.awt.Color,%20float,%20float,%20java.awt.Color,%20boolean))(float x1, float y1, [Color](http://docs.google.com/java/awt/Color.html) color1, float x2, float y2, [Color](http://docs.google.com/java/awt/Color.html) color2, boolean cyclic)            Constructs either a cyclic or acyclic GradientPaint object depending on the boolean parameter. |
| [**GradientPaint**](http://docs.google.com/java/awt/GradientPaint.html#GradientPaint(java.awt.geom.Point2D,%20java.awt.Color,%20java.awt.geom.Point2D,%20java.awt.Color))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) pt1, [Color](http://docs.google.com/java/awt/Color.html) color1, [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) pt2, [Color](http://docs.google.com/java/awt/Color.html) color2)            Constructs a simple acyclic GradientPaint object. |
| [**GradientPaint**](http://docs.google.com/java/awt/GradientPaint.html#GradientPaint(java.awt.geom.Point2D,%20java.awt.Color,%20java.awt.geom.Point2D,%20java.awt.Color,%20boolean))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) pt1, [Color](http://docs.google.com/java/awt/Color.html) color1, [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) pt2, [Color](http://docs.google.com/java/awt/Color.html) color2, boolean cyclic)            Constructs either a cyclic or acyclic GradientPaint object depending on the boolean parameter. |

| **Method Summary** | |
| --- | --- |
| [PaintContext](http://docs.google.com/java/awt/PaintContext.html) | [**createContext**](http://docs.google.com/java/awt/GradientPaint.html#createContext(java.awt.image.ColorModel,%20java.awt.Rectangle,%20java.awt.geom.Rectangle2D,%20java.awt.geom.AffineTransform,%20java.awt.RenderingHints))([ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) cm, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) deviceBounds, [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) userBounds, [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) xform, [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) hints)            Creates and returns a context used to generate the color pattern. |
| [Color](http://docs.google.com/java/awt/Color.html) | [**getColor1**](http://docs.google.com/java/awt/GradientPaint.html#getColor1())()            Returns the color C1 anchored by the point P1. |
| [Color](http://docs.google.com/java/awt/Color.html) | [**getColor2**](http://docs.google.com/java/awt/GradientPaint.html#getColor2())()            Returns the color C2 anchored by the point P2. |
| [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) | [**getPoint1**](http://docs.google.com/java/awt/GradientPaint.html#getPoint1())()            Returns a copy of the point P1 that anchors the first color. |
| [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) | [**getPoint2**](http://docs.google.com/java/awt/GradientPaint.html#getPoint2())()            Returns a copy of the point P2 which anchors the second color. |
| int | [**getTransparency**](http://docs.google.com/java/awt/GradientPaint.html#getTransparency())()            Returns the transparency mode for this GradientPaint. |
| boolean | [**isCyclic**](http://docs.google.com/java/awt/GradientPaint.html#isCyclic())()            Returns true if the gradient cycles repeatedly between the two colors C1 and C2. |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

| **Constructor Detail** |
| --- |

### GradientPaint

public **GradientPaint**(float x1,  
 float y1,  
 [Color](http://docs.google.com/java/awt/Color.html) color1,  
 float x2,  
 float y2,  
 [Color](http://docs.google.com/java/awt/Color.html) color2)

Constructs a simple acyclic GradientPaint object.

**Parameters:**x1 - x coordinate of the first specified Point in user spacey1 - y coordinate of the first specified Point in user spacecolor1 - Color at the first specified Pointx2 - x coordinate of the second specified Point in user spacey2 - y coordinate of the second specified Point in user spacecolor2 - Color at the second specified Point **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if either one of colors is null

### GradientPaint

public **GradientPaint**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) pt1,  
 [Color](http://docs.google.com/java/awt/Color.html) color1,  
 [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) pt2,  
 [Color](http://docs.google.com/java/awt/Color.html) color2)

Constructs a simple acyclic GradientPaint object.

**Parameters:**pt1 - the first specified Point in user spacecolor1 - Color at the first specified Pointpt2 - the second specified Point in user spacecolor2 - Color at the second specified Point **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if either one of colors or points is null

### GradientPaint

public **GradientPaint**(float x1,  
 float y1,  
 [Color](http://docs.google.com/java/awt/Color.html) color1,  
 float x2,  
 float y2,  
 [Color](http://docs.google.com/java/awt/Color.html) color2,  
 boolean cyclic)

Constructs either a cyclic or acyclic GradientPaint object depending on the boolean parameter.

**Parameters:**x1 - x coordinate of the first specified Point in user spacey1 - y coordinate of the first specified Point in user spacecolor1 - Color at the first specified Pointx2 - x coordinate of the second specified Point in user spacey2 - y coordinate of the second specified Point in user spacecolor2 - Color at the second specified Pointcyclic - true if the gradient pattern should cycle repeatedly between the two colors; false otherwise

### GradientPaint

public **GradientPaint**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) pt1,  
 [Color](http://docs.google.com/java/awt/Color.html) color1,  
 [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) pt2,  
 [Color](http://docs.google.com/java/awt/Color.html) color2,  
 boolean cyclic)

Constructs either a cyclic or acyclic GradientPaint object depending on the boolean parameter.

**Parameters:**pt1 - the first specified Point in user spacecolor1 - Color at the first specified Pointpt2 - the second specified Point in user spacecolor2 - Color at the second specified Pointcyclic - true if the gradient pattern should cycle repeatedly between the two colors; false otherwise **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if either one of colors or points is null

| **Method Detail** |
| --- |

### getPoint1

public [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) **getPoint1**()

Returns a copy of the point P1 that anchors the first color.

**Returns:**a [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) object that is a copy of the point that anchors the first color of this GradientPaint.

### getColor1

public [Color](http://docs.google.com/java/awt/Color.html) **getColor1**()

Returns the color C1 anchored by the point P1.

**Returns:**a Color object that is the color anchored by P1.

### getPoint2

public [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) **getPoint2**()

Returns a copy of the point P2 which anchors the second color.

**Returns:**a [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) object that is a copy of the point that anchors the second color of this GradientPaint.

### getColor2

public [Color](http://docs.google.com/java/awt/Color.html) **getColor2**()

Returns the color C2 anchored by the point P2.

**Returns:**a Color object that is the color anchored by P2.

### isCyclic

public boolean **isCyclic**()

Returns true if the gradient cycles repeatedly between the two colors C1 and C2.

**Returns:**true if the gradient cycles repeatedly between the two colors; false otherwise.

### createContext

public [PaintContext](http://docs.google.com/java/awt/PaintContext.html) **createContext**([ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) cm,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) deviceBounds,  
 [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) userBounds,  
 [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) xform,  
 [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) hints)

Creates and returns a context used to generate the color pattern.

**Specified by:**[createContext](http://docs.google.com/java/awt/Paint.html#createContext(java.awt.image.ColorModel,%20java.awt.Rectangle,%20java.awt.geom.Rectangle2D,%20java.awt.geom.AffineTransform,%20java.awt.RenderingHints)) in interface [Paint](http://docs.google.com/java/awt/Paint.html) **Parameters:**cm - [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) that receives the Paint data. This is used only as a hint.deviceBounds - the device space bounding box of the graphics primitive being rendereduserBounds - the user space bounding box of the graphics primitive being renderedxform - the [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) from user space into device spacehints - the hints that the context object uses to choose between rendering alternatives **Returns:**the [PaintContext](http://docs.google.com/java/awt/PaintContext.html) that generates color patterns.**See Also:**[PaintContext](http://docs.google.com/java/awt/PaintContext.html)

### getTransparency

public int **getTransparency**()

Returns the transparency mode for this GradientPaint.

**Specified by:**[getTransparency](http://docs.google.com/java/awt/Transparency.html#getTransparency()) in interface [Transparency](http://docs.google.com/java/awt/Transparency.html) **Returns:**an integer value representing this GradientPaint object's transparency mode.**See Also:**[Transparency](http://docs.google.com/java/awt/Transparency.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/GradientPaint.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/Frame.AccessibleAWTFrame.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Graphics.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/GradientPaint.html)    [**NO FRAMES**](http://docs.google.com/GradientPaint.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#4d34og8) | [METHOD](#lnxbz9) |

[Submit a bug or feature](http://bugs.sun.com/services/bugreport/index.jsp)

For further API reference and developer documentation, see [Java SE Developer Documentation](http://docs.google.com/webnotes/devdocs-vs-specs.html). That documentation contains more detailed, developer-targeted descriptions, with conceptual overviews, definitions of terms, workarounds, and working code examples.

Copyright 2006 Sun Microsystems, Inc. All rights reserved. Use is subject to [license terms](http://docs.google.com/legal/license.html). Also see the [documentation redistribution policy](http://java.sun.com/docs/redist.html).