| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/LinearGradientPaint.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/LayoutManager2.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/List.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/LinearGradientPaint.html)    [**NO FRAMES**](http://docs.google.com/LinearGradientPaint.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | FIELD | [CONSTR](#1t3h5sf) | [METHOD](#4d34og8) | DETAIL: FIELD | [CONSTR](#3rdcrjn) | [METHOD](#2jxsxqh) |

## **java.awt**

Class LinearGradientPaint

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

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

public final class **LinearGradientPaint**extends [MultipleGradientPaint](http://docs.google.com/java/awt/MultipleGradientPaint.html)

The LinearGradientPaint class provides a way to fill a [Shape](http://docs.google.com/java/awt/Shape.html) with a linear color gradient pattern. The user may specify two or more gradient colors, and this paint will provide an interpolation between each color. The user also specifies start and end points which define where in user space the color gradient should begin and end.

The user must provide an array of floats specifying how to distribute the colors along the gradient. These values should range from 0.0 to 1.0 and act like keyframes along the gradient (they mark where the gradient should be exactly a particular color).

In the event that the user does not set the first keyframe value equal to 0 and/or the last keyframe value equal to 1, keyframes will be created at these positions and the first and last colors will be replicated there. So, if a user specifies the following arrays to construct a gradient:

{Color.BLUE, Color.RED}, {.3f, .7f}

this will be converted to a gradient with the following keyframes:

{Color.BLUE, Color.BLUE, Color.RED, Color.RED}, {0f, .3f, .7f, 1f}

The user may also select what action the LinearGradientPaint should take when filling color outside the start and end points. If no cycle method is specified, NO\_CYCLE will be chosen by default, which means the endpoint colors will be used to fill the remaining area.

The colorSpace parameter allows the user to specify in which colorspace the interpolation should be performed, default sRGB or linearized RGB.

The following code demonstrates typical usage of LinearGradientPaint:

Point2D start = new Point2D.Float(0, 0);  
 Point2D end = new Point2D.Float(50, 50);  
 float[] dist = {0.0f, 0.2f, 1.0f};  
 Color[] colors = {Color.RED, Color.WHITE, Color.BLUE};  
 LinearGradientPaint p =  
 new LinearGradientPaint(start, end, dist, colors);

This code will create a LinearGradientPaint which interpolates between red and white for the first 20% of the gradient and between white and blue for the remaining 80%.

This image demonstrates the example code above for each of the three cycle methods:



**Since:** 1.6 **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))

| **Nested Class Summary** | |
| --- | --- |

| **Nested classes/interfaces inherited from class java.awt.**[**MultipleGradientPaint**](http://docs.google.com/java/awt/MultipleGradientPaint.html) |
| --- |
| [MultipleGradientPaint.ColorSpaceType](http://docs.google.com/java/awt/MultipleGradientPaint.ColorSpaceType.html), [MultipleGradientPaint.CycleMethod](http://docs.google.com/java/awt/MultipleGradientPaint.CycleMethod.html) |

| **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** | |
| --- | --- |
| [**LinearGradientPaint**](http://docs.google.com/java/awt/LinearGradientPaint.html#LinearGradientPaint(float,%20float,%20float,%20float,%20float%5B%5D,%20java.awt.Color%5B%5D))(float startX, float startY, float endX, float endY, float[] fractions, [Color](http://docs.google.com/java/awt/Color.html)[] colors)            Constructs a LinearGradientPaint with a default NO\_CYCLE repeating method and SRGB color space. |
| [**LinearGradientPaint**](http://docs.google.com/java/awt/LinearGradientPaint.html#LinearGradientPaint(float,%20float,%20float,%20float,%20float%5B%5D,%20java.awt.Color%5B%5D,%20java.awt.MultipleGradientPaint.CycleMethod))(float startX, float startY, float endX, float endY, float[] fractions, [Color](http://docs.google.com/java/awt/Color.html)[] colors, [MultipleGradientPaint.CycleMethod](http://docs.google.com/java/awt/MultipleGradientPaint.CycleMethod.html) cycleMethod)            Constructs a LinearGradientPaint with a default SRGB color space. |
| [**LinearGradientPaint**](http://docs.google.com/java/awt/LinearGradientPaint.html#LinearGradientPaint(java.awt.geom.Point2D,%20java.awt.geom.Point2D,%20float%5B%5D,%20java.awt.Color%5B%5D))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) start, [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) end, float[] fractions, [Color](http://docs.google.com/java/awt/Color.html)[] colors)            Constructs a LinearGradientPaint with a default NO\_CYCLE repeating method and SRGB color space. |
| [**LinearGradientPaint**](http://docs.google.com/java/awt/LinearGradientPaint.html#LinearGradientPaint(java.awt.geom.Point2D,%20java.awt.geom.Point2D,%20float%5B%5D,%20java.awt.Color%5B%5D,%20java.awt.MultipleGradientPaint.CycleMethod))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) start, [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) end, float[] fractions, [Color](http://docs.google.com/java/awt/Color.html)[] colors, [MultipleGradientPaint.CycleMethod](http://docs.google.com/java/awt/MultipleGradientPaint.CycleMethod.html) cycleMethod)            Constructs a LinearGradientPaint with a default SRGB color space. |
| [**LinearGradientPaint**](http://docs.google.com/java/awt/LinearGradientPaint.html#LinearGradientPaint(java.awt.geom.Point2D,%20java.awt.geom.Point2D,%20float%5B%5D,%20java.awt.Color%5B%5D,%20java.awt.MultipleGradientPaint.CycleMethod,%20java.awt.MultipleGradientPaint.ColorSpaceType,%20java.awt.geom.AffineTransform))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) start, [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) end, float[] fractions, [Color](http://docs.google.com/java/awt/Color.html)[] colors, [MultipleGradientPaint.CycleMethod](http://docs.google.com/java/awt/MultipleGradientPaint.CycleMethod.html) cycleMethod, [MultipleGradientPaint.ColorSpaceType](http://docs.google.com/java/awt/MultipleGradientPaint.ColorSpaceType.html) colorSpace, [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) gradientTransform)            Constructs a LinearGradientPaint. |

| **Method Summary** | |
| --- | --- |
| [PaintContext](http://docs.google.com/java/awt/PaintContext.html) | [**createContext**](http://docs.google.com/java/awt/LinearGradientPaint.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) transform, [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) hints)            Creates and returns a [PaintContext](http://docs.google.com/java/awt/PaintContext.html) used to generate the color pattern. |
| [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) | [**getEndPoint**](http://docs.google.com/java/awt/LinearGradientPaint.html#getEndPoint())()            Returns a copy of the end point of the gradient axis. |
| [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) | [**getStartPoint**](http://docs.google.com/java/awt/LinearGradientPaint.html#getStartPoint())()            Returns a copy of the start point of the gradient axis. |

| **Methods inherited from class java.awt.**[**MultipleGradientPaint**](http://docs.google.com/java/awt/MultipleGradientPaint.html) |
| --- |
| [getColors](http://docs.google.com/java/awt/MultipleGradientPaint.html#getColors()), [getColorSpace](http://docs.google.com/java/awt/MultipleGradientPaint.html#getColorSpace()), [getCycleMethod](http://docs.google.com/java/awt/MultipleGradientPaint.html#getCycleMethod()), [getFractions](http://docs.google.com/java/awt/MultipleGradientPaint.html#getFractions()), [getTransform](http://docs.google.com/java/awt/MultipleGradientPaint.html#getTransform()), [getTransparency](http://docs.google.com/java/awt/MultipleGradientPaint.html#getTransparency()) |

| **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** |
| --- |

### LinearGradientPaint

public **LinearGradientPaint**(float startX,  
 float startY,  
 float endX,  
 float endY,  
 float[] fractions,  
 [Color](http://docs.google.com/java/awt/Color.html)[] colors)

Constructs a LinearGradientPaint with a default NO\_CYCLE repeating method and SRGB color space.

**Parameters:**startX - the X coordinate of the gradient axis start point in user spacestartY - the Y coordinate of the gradient axis start point in user spaceendX - the X coordinate of the gradient axis end point in user spaceendY - the Y coordinate of the gradient axis end point in user spacefractions - numbers ranging from 0.0 to 1.0 specifying the distribution of colors along the gradientcolors - array of colors corresponding to each fractional value **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if fractions array is null, or colors array is null, [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if start and end points are the same points, or fractions.length != colors.length, or colors is less than 2 in size, or a fractions value is less than 0.0 or greater than 1.0, or the fractions are not provided in strictly increasing order

### LinearGradientPaint

public **LinearGradientPaint**(float startX,  
 float startY,  
 float endX,  
 float endY,  
 float[] fractions,  
 [Color](http://docs.google.com/java/awt/Color.html)[] colors,  
 [MultipleGradientPaint.CycleMethod](http://docs.google.com/java/awt/MultipleGradientPaint.CycleMethod.html) cycleMethod)

Constructs a LinearGradientPaint with a default SRGB color space.

**Parameters:**startX - the X coordinate of the gradient axis start point in user spacestartY - the Y coordinate of the gradient axis start point in user spaceendX - the X coordinate of the gradient axis end point in user spaceendY - the Y coordinate of the gradient axis end point in user spacefractions - numbers ranging from 0.0 to 1.0 specifying the distribution of colors along the gradientcolors - array of colors corresponding to each fractional valuecycleMethod - either NO\_CYCLE, REFLECT, or REPEAT **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if fractions array is null, or colors array is null, or cycleMethod is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if start and end points are the same points, or fractions.length != colors.length, or colors is less than 2 in size, or a fractions value is less than 0.0 or greater than 1.0, or the fractions are not provided in strictly increasing order

### LinearGradientPaint

public **LinearGradientPaint**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) start,  
 [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) end,  
 float[] fractions,  
 [Color](http://docs.google.com/java/awt/Color.html)[] colors)

Constructs a LinearGradientPaint with a default NO\_CYCLE repeating method and SRGB color space.

**Parameters:**start - the gradient axis start Point2D in user spaceend - the gradient axis end Point2D in user spacefractions - numbers ranging from 0.0 to 1.0 specifying the distribution of colors along the gradientcolors - array of colors corresponding to each fractional value **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if one of the points is null, or fractions array is null, or colors array is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if start and end points are the same points, or fractions.length != colors.length, or colors is less than 2 in size, or a fractions value is less than 0.0 or greater than 1.0, or the fractions are not provided in strictly increasing order

### LinearGradientPaint

public **LinearGradientPaint**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) start,  
 [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) end,  
 float[] fractions,  
 [Color](http://docs.google.com/java/awt/Color.html)[] colors,  
 [MultipleGradientPaint.CycleMethod](http://docs.google.com/java/awt/MultipleGradientPaint.CycleMethod.html) cycleMethod)

Constructs a LinearGradientPaint with a default SRGB color space.

**Parameters:**start - the gradient axis start Point2D in user spaceend - the gradient axis end Point2D in user spacefractions - numbers ranging from 0.0 to 1.0 specifying the distribution of colors along the gradientcolors - array of colors corresponding to each fractional valuecycleMethod - either NO\_CYCLE, REFLECT, or REPEAT **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if one of the points is null, or fractions array is null, or colors array is null, or cycleMethod is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if start and end points are the same points, or fractions.length != colors.length, or colors is less than 2 in size, or a fractions value is less than 0.0 or greater than 1.0, or the fractions are not provided in strictly increasing order

### LinearGradientPaint

public **LinearGradientPaint**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) start,  
 [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) end,  
 float[] fractions,  
 [Color](http://docs.google.com/java/awt/Color.html)[] colors,  
 [MultipleGradientPaint.CycleMethod](http://docs.google.com/java/awt/MultipleGradientPaint.CycleMethod.html) cycleMethod,  
 [MultipleGradientPaint.ColorSpaceType](http://docs.google.com/java/awt/MultipleGradientPaint.ColorSpaceType.html) colorSpace,  
 [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) gradientTransform)

Constructs a LinearGradientPaint.

**Parameters:**start - the gradient axis start Point2D in user spaceend - the gradient axis end Point2D in user spacefractions - numbers ranging from 0.0 to 1.0 specifying the distribution of colors along the gradientcolors - array of colors corresponding to each fractional valuecycleMethod - either NO\_CYCLE, REFLECT, or REPEATcolorSpace - which color space to use for interpolation, either SRGB or LINEAR\_RGBgradientTransform - transform to apply to the gradient **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if one of the points is null, or fractions array is null, or colors array is null, or cycleMethod is null, or colorSpace is null, or gradientTransform is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if start and end points are the same points, or fractions.length != colors.length, or colors is less than 2 in size, or a fractions value is less than 0.0 or greater than 1.0, or the fractions are not provided in strictly increasing order

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

### 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) transform,  
 [RenderingHints](http://docs.google.com/java/awt/RenderingHints.html) hints)

Creates and returns a [PaintContext](http://docs.google.com/java/awt/PaintContext.html) used to generate the color pattern. Since the ColorModel argument to createContext is only a hint, implementations of Paint should accept a null argument for ColorModel. Note that if the application does not prefer a specific ColorModel, the null ColorModel argument will give the Paint implementation full leeway in using the most efficient ColorModel it prefers for its raster processing.

Since the API documentation was not specific about this in releases before 1.4, there may be implementations of Paint that do not accept a null ColorModel argument. If a developer is writing code which passes a null ColorModel argument to the createContext method of Paint objects from arbitrary sources it would be wise to code defensively by manufacturing a non-null ColorModel for those objects which throw a NullPointerException.

**Parameters:**cm - the [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 renderedtransform - the [AffineTransform](http://docs.google.com/java/awt/geom/AffineTransform.html) from user space into device spacehints - the hint that the context object uses to choose between rendering alternatives **Returns:**the PaintContext for generating color patterns**See Also:**[PaintContext](http://docs.google.com/java/awt/PaintContext.html)

### getStartPoint

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

Returns a copy of the start point of the gradient axis.

**Returns:**a Point2D object that is a copy of the point that anchors the first color of this LinearGradientPaint

### getEndPoint

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

Returns a copy of the end point of the gradient axis.

**Returns:**a Point2D object that is a copy of the point that anchors the last color of this LinearGradientPaint

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/LinearGradientPaint.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/LayoutManager2.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/List.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/LinearGradientPaint.html)    [**NO FRAMES**](http://docs.google.com/LinearGradientPaint.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | FIELD | [CONSTR](#1t3h5sf) | [METHOD](#4d34og8) | DETAIL: FIELD | [CONSTR](#3rdcrjn) | [METHOD](#2jxsxqh) |

[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).