| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RadialGradientPaint.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/PrintJob.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Rectangle.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/RadialGradientPaint.html)    [**NO FRAMES**](http://docs.google.com/RadialGradientPaint.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | FIELD | [CONSTR](#1t3h5sf) | [METHOD](#4d34og8) | DETAIL: FIELD | [CONSTR](#3rdcrjn) | [METHOD](#1y810tw) |

## **java.awt**

Class RadialGradientPaint

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

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

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

The RadialGradientPaint class provides a way to fill a shape with a circular radial color gradient pattern. The user may specify 2 or more gradient colors, and this paint will provide an interpolation between each color.

The user must specify the circle controlling the gradient pattern, which is described by a center point and a radius. The user can also specify a separate focus point within that circle, which controls the location of the first color of the gradient. By default the focus is set to be the center of the circle.

This paint will map the first color of the gradient to the focus point, and the last color to the perimeter of the circle, interpolating smoothly for any in-between colors specified by the user. Any line drawn from the focus point to the circumference will thus span all the gradient colors.

Specifying a focus point outside of the circle's radius will result in the focus being set to the intersection point of the focus-center line and the perimeter of the circle.

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 RadialGradientPaint should take when filling color outside the bounds of the circle's radius. If no cycle method is specified, NO\_CYCLE will be chosen by default, which means the the last keyframe color 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 RadialGradientPaint, where the center and focus points are the same:

Point2D center = new Point2D.Float(50, 50);  
 float radius = 25;  
 float[] dist = {0.0f, 0.2f, 1.0f};  
 Color[] colors = {Color.RED, Color.WHITE, Color.BLUE};  
 RadialGradientPaint p =  
 new RadialGradientPaint(center, radius, dist, colors);

This image demonstrates the example code above, with default (centered) focus for each of the three cycle methods:



It is also possible to specify a non-centered focus point, as in the following code:

Point2D center = new Point2D.Float(50, 50);  
 float radius = 25;  
 Point2D focus = new Point2D.Float(40, 40);  
 float[] dist = {0.0f, 0.2f, 1.0f};  
 Color[] colors = {Color.RED, Color.WHITE, Color.BLUE};  
 RadialGradientPaint p =  
 new RadialGradientPaint(center, radius, focus,  
 dist, colors,  
 CycleMethod.NO\_CYCLE);

This image demonstrates the previous example code, with non-centered focus 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** | |
| --- | --- |
| [**RadialGradientPaint**](http://docs.google.com/java/awt/RadialGradientPaint.html#RadialGradientPaint(float,%20float,%20float,%20float%5B%5D,%20java.awt.Color%5B%5D))(float cx, float cy, float radius, float[] fractions, [Color](http://docs.google.com/java/awt/Color.html)[] colors)            Constructs a RadialGradientPaint with a default NO\_CYCLE repeating method and SRGB color space, using the center as the focus point. |
| [**RadialGradientPaint**](http://docs.google.com/java/awt/RadialGradientPaint.html#RadialGradientPaint(float,%20float,%20float,%20float%5B%5D,%20java.awt.Color%5B%5D,%20java.awt.MultipleGradientPaint.CycleMethod))(float cx, float cy, float radius, 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 RadialGradientPaint with a default SRGB color space, using the center as the focus point. |
| [**RadialGradientPaint**](http://docs.google.com/java/awt/RadialGradientPaint.html#RadialGradientPaint(float,%20float,%20float,%20float,%20float,%20float%5B%5D,%20java.awt.Color%5B%5D,%20java.awt.MultipleGradientPaint.CycleMethod))(float cx, float cy, float radius, float fx, float fy, 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 RadialGradientPaint with a default SRGB color space. |
| [**RadialGradientPaint**](http://docs.google.com/java/awt/RadialGradientPaint.html#RadialGradientPaint(java.awt.geom.Point2D,%20float,%20float%5B%5D,%20java.awt.Color%5B%5D))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) center, float radius, float[] fractions, [Color](http://docs.google.com/java/awt/Color.html)[] colors)            Constructs a RadialGradientPaint with a default NO\_CYCLE repeating method and SRGB color space, using the center as the focus point. |
| [**RadialGradientPaint**](http://docs.google.com/java/awt/RadialGradientPaint.html#RadialGradientPaint(java.awt.geom.Point2D,%20float,%20float%5B%5D,%20java.awt.Color%5B%5D,%20java.awt.MultipleGradientPaint.CycleMethod))([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) center, float radius, 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 RadialGradientPaint with a default SRGB color space, using the center as the focus point. |
| [**RadialGradientPaint**](http://docs.google.com/java/awt/RadialGradientPaint.html#RadialGradientPaint(java.awt.geom.Point2D,%20float,%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) center, float radius, [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) focus, 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 RadialGradientPaint with a default SRGB color space. |
| [**RadialGradientPaint**](http://docs.google.com/java/awt/RadialGradientPaint.html#RadialGradientPaint(java.awt.geom.Point2D,%20float,%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) center, float radius, [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) focus, 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 RadialGradientPaint. |
| [**RadialGradientPaint**](http://docs.google.com/java/awt/RadialGradientPaint.html#RadialGradientPaint(java.awt.geom.Rectangle2D,%20float%5B%5D,%20java.awt.Color%5B%5D,%20java.awt.MultipleGradientPaint.CycleMethod))([Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) gradientBounds, 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 RadialGradientPaint with a default SRGB color space. |

| **Method Summary** | |
| --- | --- |
| [PaintContext](http://docs.google.com/java/awt/PaintContext.html) | [**createContext**](http://docs.google.com/java/awt/RadialGradientPaint.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) | [**getCenterPoint**](http://docs.google.com/java/awt/RadialGradientPaint.html#getCenterPoint())()            Returns a copy of the center point of the radial gradient. |
| [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) | [**getFocusPoint**](http://docs.google.com/java/awt/RadialGradientPaint.html#getFocusPoint())()            Returns a copy of the end point of the gradient axis. |
| float | [**getRadius**](http://docs.google.com/java/awt/RadialGradientPaint.html#getRadius())()            Returns the radius of the circle defining the radial gradient. |

| **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** |
| --- |

### RadialGradientPaint

public **RadialGradientPaint**(float cx,  
 float cy,  
 float radius,  
 float[] fractions,  
 [Color](http://docs.google.com/java/awt/Color.html)[] colors)

Constructs a RadialGradientPaint with a default NO\_CYCLE repeating method and SRGB color space, using the center as the focus point.

**Parameters:**cx - the X coordinate in user space of the center point of the circle defining the gradient. The last color of the gradient is mapped to the perimeter of this circle.cy - the Y coordinate in user space of the center point of the circle defining the gradient. The last color of the gradient is mapped to the perimeter of this circle.radius - the radius of the circle defining the extents of the color gradientfractions - numbers ranging from 0.0 to 1.0 specifying the distribution of colors along the gradientcolors - array of colors to use in the gradient. The first color is used at the focus point, the last color around the perimeter of the circle. **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 radius is non-positive, 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

### RadialGradientPaint

public **RadialGradientPaint**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) center,  
 float radius,  
 float[] fractions,  
 [Color](http://docs.google.com/java/awt/Color.html)[] colors)

Constructs a RadialGradientPaint with a default NO\_CYCLE repeating method and SRGB color space, using the center as the focus point.

**Parameters:**center - the center point, in user space, of the circle defining the gradientradius - the radius of the circle defining the extents of the color gradientfractions - numbers ranging from 0.0 to 1.0 specifying the distribution of colors along the gradientcolors - array of colors to use in the gradient. The first color is used at the focus point, the last color around the perimeter of the circle. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if center point is null, or fractions array is null, or colors array is null [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if radius is non-positive, 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

### RadialGradientPaint

public **RadialGradientPaint**(float cx,  
 float cy,  
 float radius,  
 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 RadialGradientPaint with a default SRGB color space, using the center as the focus point.

**Parameters:**cx - the X coordinate in user space of the center point of the circle defining the gradient. The last color of the gradient is mapped to the perimeter of this circle.cy - the Y coordinate in user space of the center point of the circle defining the gradient. The last color of the gradient is mapped to the perimeter of this circle.radius - the radius of the circle defining the extents of the color gradientfractions - numbers ranging from 0.0 to 1.0 specifying the distribution of colors along the gradientcolors - array of colors to use in the gradient. The first color is used at the focus point, the last color around the perimeter of the circle.cycleMethod - 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 radius is non-positive, 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

### RadialGradientPaint

public **RadialGradientPaint**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) center,  
 float radius,  
 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 RadialGradientPaint with a default SRGB color space, using the center as the focus point.

**Parameters:**center - the center point, in user space, of the circle defining the gradientradius - the radius of the circle defining the extents of the color gradientfractions - numbers ranging from 0.0 to 1.0 specifying the distribution of colors along the gradientcolors - array of colors to use in the gradient. The first color is used at the focus point, the last color around the perimeter of the circle.cycleMethod - either NO\_CYCLE, REFLECT, or REPEAT **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if center point 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 radius is non-positive, 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

### RadialGradientPaint

public **RadialGradientPaint**(float cx,  
 float cy,  
 float radius,  
 float fx,  
 float fy,  
 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 RadialGradientPaint with a default SRGB color space.

**Parameters:**cx - the X coordinate in user space of the center point of the circle defining the gradient. The last color of the gradient is mapped to the perimeter of this circle.cy - the Y coordinate in user space of the center point of the circle defining the gradient. The last color of the gradient is mapped to the perimeter of this circle.radius - the radius of the circle defining the extents of the color gradientfx - the X coordinate of the point in user space to which the first color is mappedfy - the Y coordinate of the point in user space to which the first color is mappedfractions - numbers ranging from 0.0 to 1.0 specifying the distribution of colors along the gradientcolors - array of colors to use in the gradient. The first color is used at the focus point, the last color around the perimeter of the circle.cycleMethod - 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 radius is non-positive, 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

### RadialGradientPaint

public **RadialGradientPaint**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) center,  
 float radius,  
 [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) focus,  
 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 RadialGradientPaint with a default SRGB color space.

**Parameters:**center - the center point, in user space, of the circle defining the gradient. The last color of the gradient is mapped to the perimeter of this circle.radius - the radius of the circle defining the extents of the color gradientfocus - the point in user space to which the first color is mappedfractions - numbers ranging from 0.0 to 1.0 specifying the distribution of colors along the gradientcolors - array of colors to use in the gradient. The first color is used at the focus point, the last color around the perimeter of the circle.cycleMethod - 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 radius is non-positive, 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

### RadialGradientPaint

public **RadialGradientPaint**([Point2D](http://docs.google.com/java/awt/geom/Point2D.html) center,  
 float radius,  
 [Point2D](http://docs.google.com/java/awt/geom/Point2D.html) focus,  
 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 RadialGradientPaint.

**Parameters:**center - the center point in user space of the circle defining the gradient. The last color of the gradient is mapped to the perimeter of this circle.radius - the radius of the circle defining the extents of the color gradientfocus - the point in user space to which the first color is mappedfractions - numbers ranging from 0.0 to 1.0 specifying the distribution of colors along the gradientcolors - array of colors to use in the gradient. The first color is used at the focus point, the last color around the perimeter of the circle.cycleMethod - 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 radius is non-positive, 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

### RadialGradientPaint

public **RadialGradientPaint**([Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) gradientBounds,  
 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 RadialGradientPaint with a default SRGB color space. The gradient circle of the RadialGradientPaint is defined by the given bounding box.

This constructor is a more convenient way to express the following (equivalent) code:

double gw = gradientBounds.getWidth();  
 double gh = gradientBounds.getHeight();  
 double cx = gradientBounds.getCenterX();  
 double cy = gradientBounds.getCenterY();  
 Point2D center = new Point2D.Double(cx, cy);  
  
 AffineTransform gradientTransform = new AffineTransform();  
 gradientTransform.translate(cx, cy);  
 gradientTransform.scale(gw / 2, gh / 2);  
 gradientTransform.translate(-cx, -cy);  
  
 RadialGradientPaint gp =  
 new RadialGradientPaint(center, 1.0f, center,  
 fractions, colors,  
 cycleMethod,  
 ColorSpaceType.SRGB,  
 gradientTransform);

**Parameters:**gradientBounds - the bounding box, in user space, of the circle defining the outermost extent of the gradientfractions - numbers ranging from 0.0 to 1.0 specifying the distribution of colors along the gradientcolors - array of colors to use in the gradient. The first color is used at the focus point, the last color around the perimeter of the circle.cycleMethod - either NO\_CYCLE, REFLECT, or REPEAT **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if gradientBounds 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 gradientBounds is empty, 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)

### getCenterPoint

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

Returns a copy of the center point of the radial gradient.

**Returns:**a Point2D object that is a copy of the center point

### getFocusPoint

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

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

**Returns:**a Point2D object that is a copy of the focus point

### getRadius

public float **getRadius**()

Returns the radius of the circle defining the radial gradient.

**Returns:**the radius of the circle defining the radial gradient

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/RadialGradientPaint.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/PrintJob.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Rectangle.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/RadialGradientPaint.html)    [**NO FRAMES**](http://docs.google.com/RadialGradientPaint.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | FIELD | [CONSTR](#1t3h5sf) | [METHOD](#4d34og8) | DETAIL: FIELD | [CONSTR](#3rdcrjn) | [METHOD](#1y810tw) |

[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).