| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Rectangle.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/RadialGradientPaint.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/RenderingHints.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Rectangle.html)    [**NO FRAMES**](http://docs.google.com/Rectangle.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3dy6vkm) | [FIELD](#1t3h5sf) | [CONSTR](#2s8eyo1) | [METHOD](#17dp8vu) | DETAIL: [FIELD](#1ksv4uv) | [CONSTR](#1y810tw) | [METHOD](#1pxezwc) |

## **java.awt**

Class Rectangle

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.awt.geom.RectangularShape](http://docs.google.com/java/awt/geom/RectangularShape.html)  
 [java.awt.geom.Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html)  
 **java.awt.Rectangle**

**All Implemented Interfaces:** [Shape](http://docs.google.com/java/awt/Shape.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [Cloneable](http://docs.google.com/java/lang/Cloneable.html) **Direct Known Subclasses:** [DefaultCaret](http://docs.google.com/javax/swing/text/DefaultCaret.html)

public class **Rectangle**extends [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html)implements [Shape](http://docs.google.com/java/awt/Shape.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

A Rectangle specifies an area in a coordinate space that is enclosed by the Rectangle object's upper-left point (x,y) in the coordinate space, its width, and its height.

A Rectangle object's width and height are public fields. The constructors that create a Rectangle, and the methods that can modify one, do not prevent setting a negative value for width or height.

A Rectangle whose width or height is exactly zero has location along those axes with zero dimension, but is otherwise considered empty. The [isEmpty()](http://docs.google.com/java/awt/Rectangle.html#isEmpty()) method will return true for such a Rectangle. Methods which test if an empty Rectangle contains or intersects a point or rectangle will always return false if either dimension is zero. Methods which combine such a Rectangle with a point or rectangle will include the location of the Rectangle on that axis in the result as if the [add(Point)](http://docs.google.com/java/awt/Rectangle.html#add(java.awt.Point)) method were being called.

A Rectangle whose width or height is negative has neither location nor dimension along those axes with negative dimensions. Such a Rectangle is treated as non-existant along those axes. Such a Rectangle is also empty with respect to containment calculations and methods which test if it contains or intersects a point or rectangle will always return false. Methods which combine such a Rectangle with a point or rectangle will ignore the Rectangle entirely in generating the result. If two Rectangle objects are combined and each has a negative dimension, the result will have at least one negative dimension.

Methods which affect only the location of a Rectangle will operate on its location regardless of whether or not it has a negative or zero dimension along either axis.

Note that a Rectangle constructed with the default no-argument constructor will have dimensions of 0x0 and therefore be empty. That Rectangle will still have a location of (0,0) and will contribute that location to the union and add operations. Code attempting to accumulate the bounds of a set of points should therefore initially construct the Rectangle with a specifically negative width and height or it should use the first point in the set to construct the Rectangle. For example:

Rectangle bounds = new Rectangle(0, 0, -1, -1);  
 for (int i = 0; i < points.length; i++) {  
 bounds.add(points[i]);  
 }

or if we know that the points array contains at least one point:

Rectangle bounds = new Rectangle(points[0]);  
 for (int i = 1; i < points.length; i++) {  
 bounds.add(points[i]);  
 }

This class uses 32-bit integers to store its location and dimensions. Frequently operations may produce a result that exceeds the range of a 32-bit integer. The methods will calculate their results in a way that avoids any 32-bit overflow for intermediate results and then choose the best representation to store the final results back into the 32-bit fields which hold the location and dimensions. The location of the result will be stored into the [x](http://docs.google.com/java/awt/Rectangle.html#x) and [y](http://docs.google.com/java/awt/Rectangle.html#y) fields by clipping the true result to the nearest 32-bit value. The values stored into the [width](http://docs.google.com/java/awt/Rectangle.html#width) and [height](http://docs.google.com/java/awt/Rectangle.html#height) dimension fields will be chosen as the 32-bit values that encompass the largest part of the true result as possible. Generally this means that the dimension will be clipped independently to the range of 32-bit integers except that if the location had to be moved to store it into its pair of 32-bit fields then the dimensions will be adjusted relative to the "best representation" of the location. If the true result had a negative dimension and was therefore non-existant along one or both axes, the stored dimensions will be negative numbers in those axes. If the true result had a location that could be represented within the range of 32-bit integers, but zero dimension along one or both axes, then the stored dimensions will be zero in those axes.

**Since:** 1.0 **See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#java.awt.Rectangle)

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

| **Nested classes/interfaces inherited from class java.awt.geom.**[**Rectangle2D**](http://docs.google.com/java/awt/geom/Rectangle2D.html) |
| --- |
| [Rectangle2D.Double](http://docs.google.com/java/awt/geom/Rectangle2D.Double.html), [Rectangle2D.Float](http://docs.google.com/java/awt/geom/Rectangle2D.Float.html) |

| **Field Summary** | |
| --- | --- |
| int | [**height**](http://docs.google.com/java/awt/Rectangle.html#height)            The height of the Rectangle. |
| int | [**width**](http://docs.google.com/java/awt/Rectangle.html#width)            The width of the Rectangle. |
| int | [**x**](http://docs.google.com/java/awt/Rectangle.html#x)            The X coordinate of the upper-left corner of the Rectangle. |
| int | [**y**](http://docs.google.com/java/awt/Rectangle.html#y)            The Y coordinate of the upper-left corner of the Rectangle. |

| **Fields inherited from class java.awt.geom.**[**Rectangle2D**](http://docs.google.com/java/awt/geom/Rectangle2D.html) |
| --- |
| [OUT\_BOTTOM](http://docs.google.com/java/awt/geom/Rectangle2D.html#OUT_BOTTOM), [OUT\_LEFT](http://docs.google.com/java/awt/geom/Rectangle2D.html#OUT_LEFT), [OUT\_RIGHT](http://docs.google.com/java/awt/geom/Rectangle2D.html#OUT_RIGHT), [OUT\_TOP](http://docs.google.com/java/awt/geom/Rectangle2D.html#OUT_TOP) |

| **Constructor Summary** | |
| --- | --- |
| [**Rectangle**](http://docs.google.com/java/awt/Rectangle.html#Rectangle())()            Constructs a new Rectangle whose upper-left corner is at (0, 0) in the coordinate space, and whose width and height are both zero. |
| [**Rectangle**](http://docs.google.com/java/awt/Rectangle.html#Rectangle(java.awt.Dimension))([Dimension](http://docs.google.com/java/awt/Dimension.html) d)            Constructs a new Rectangle whose top left corner is (0, 0) and whose width and height are specified by the Dimension argument. |
| [**Rectangle**](http://docs.google.com/java/awt/Rectangle.html#Rectangle(int,%20int))(int width, int height)            Constructs a new Rectangle whose upper-left corner is at (0, 0) in the coordinate space, and whose width and height are specified by the arguments of the same name. |
| [**Rectangle**](http://docs.google.com/java/awt/Rectangle.html#Rectangle(int,%20int,%20int,%20int))(int x, int y, int width, int height)            Constructs a new Rectangle whose upper-left corner is specified as (x,y) and whose width and height are specified by the arguments of the same name. |
| [**Rectangle**](http://docs.google.com/java/awt/Rectangle.html#Rectangle(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) p)            Constructs a new Rectangle whose upper-left corner is the specified Point, and whose width and height are both zero. |
| [**Rectangle**](http://docs.google.com/java/awt/Rectangle.html#Rectangle(java.awt.Point,%20java.awt.Dimension))([Point](http://docs.google.com/java/awt/Point.html) p, [Dimension](http://docs.google.com/java/awt/Dimension.html) d)            Constructs a new Rectangle whose upper-left corner is specified by the [Point](http://docs.google.com/java/awt/Point.html) argument, and whose width and height are specified by the [Dimension](http://docs.google.com/java/awt/Dimension.html) argument. |
| [**Rectangle**](http://docs.google.com/java/awt/Rectangle.html#Rectangle(java.awt.Rectangle))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)            Constructs a new Rectangle, initialized to match the values of the specified Rectangle. |

| **Method Summary** | |
| --- | --- |
| void | [**add**](http://docs.google.com/java/awt/Rectangle.html#add(int,%20int))(int newx, int newy)            Adds a point, specified by the integer arguments newx,newy to the bounds of this Rectangle. |
| void | [**add**](http://docs.google.com/java/awt/Rectangle.html#add(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) pt)            Adds the specified Point to the bounds of this Rectangle. |
| void | [**add**](http://docs.google.com/java/awt/Rectangle.html#add(java.awt.Rectangle))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)            Adds a Rectangle to this Rectangle. |
| boolean | [**contains**](http://docs.google.com/java/awt/Rectangle.html#contains(int,%20int))(int x, int y)            Checks whether or not this Rectangle contains the point at the specified location (x,y). |
| boolean | [**contains**](http://docs.google.com/java/awt/Rectangle.html#contains(int,%20int,%20int,%20int))(int X, int Y, int W, int H)            Checks whether this Rectangle entirely contains the Rectangle at the specified location (X,Y) with the specified dimensions (W,H). |
| boolean | [**contains**](http://docs.google.com/java/awt/Rectangle.html#contains(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) p)            Checks whether or not this Rectangle contains the specified Point. |
| boolean | [**contains**](http://docs.google.com/java/awt/Rectangle.html#contains(java.awt.Rectangle))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)            Checks whether or not this Rectangle entirely contains the specified Rectangle. |
| [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) | [**createIntersection**](http://docs.google.com/java/awt/Rectangle.html#createIntersection(java.awt.geom.Rectangle2D))([Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) r)            Returns a new Rectangle2D object representing the intersection of this Rectangle2D with the specified Rectangle2D. |
| [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) | [**createUnion**](http://docs.google.com/java/awt/Rectangle.html#createUnion(java.awt.geom.Rectangle2D))([Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) r)            Returns a new Rectangle2D object representing the union of this Rectangle2D with the specified Rectangle2D. |
| boolean | [**equals**](http://docs.google.com/java/awt/Rectangle.html#equals(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) obj)            Checks whether two rectangles are equal. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getBounds**](http://docs.google.com/java/awt/Rectangle.html#getBounds())()            Gets the bounding Rectangle of this Rectangle. |
| [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) | [**getBounds2D**](http://docs.google.com/java/awt/Rectangle.html#getBounds2D())()            Returns a high precision and more accurate bounding box of the Shape than the getBounds method. |
| double | [**getHeight**](http://docs.google.com/java/awt/Rectangle.html#getHeight())()            Returns the height of the bounding Rectangle in double precision. |
| [Point](http://docs.google.com/java/awt/Point.html) | [**getLocation**](http://docs.google.com/java/awt/Rectangle.html#getLocation())()            Returns the location of this Rectangle. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getSize**](http://docs.google.com/java/awt/Rectangle.html#getSize())()            Gets the size of this Rectangle, represented by the returned Dimension. |
| double | [**getWidth**](http://docs.google.com/java/awt/Rectangle.html#getWidth())()            Returns the width of the bounding Rectangle in double precision. |
| double | [**getX**](http://docs.google.com/java/awt/Rectangle.html#getX())()            Returns the X coordinate of the bounding Rectangle in double precision. |
| double | [**getY**](http://docs.google.com/java/awt/Rectangle.html#getY())()            Returns the Y coordinate of the bounding Rectangle in double precision. |
| void | [**grow**](http://docs.google.com/java/awt/Rectangle.html#grow(int,%20int))(int h, int v)            Resizes the Rectangle both horizontally and vertically. |
| boolean | [**inside**](http://docs.google.com/java/awt/Rectangle.html#inside(int,%20int))(int X, int Y)  **Deprecated.** *As of JDK version 1.1, replaced by contains(int, int).* |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**intersection**](http://docs.google.com/java/awt/Rectangle.html#intersection(java.awt.Rectangle))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)            Computes the intersection of this Rectangle with the specified Rectangle. |
| boolean | [**intersects**](http://docs.google.com/java/awt/Rectangle.html#intersects(java.awt.Rectangle))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)            Determines whether or not this Rectangle and the specified Rectangle intersect. |
| boolean | [**isEmpty**](http://docs.google.com/java/awt/Rectangle.html#isEmpty())()            Determines whether the RectangularShape is empty. |
| void | [**move**](http://docs.google.com/java/awt/Rectangle.html#move(int,%20int))(int x, int y)  **Deprecated.** *As of JDK version 1.1, replaced by setLocation(int, int).* |
| int | [**outcode**](http://docs.google.com/java/awt/Rectangle.html#outcode(double,%20double))(double x, double y)            Determines where the specified coordinates lie with respect to this Rectangle2D. |
| void | [**reshape**](http://docs.google.com/java/awt/Rectangle.html#reshape(int,%20int,%20int,%20int))(int x, int y, int width, int height)  **Deprecated.** *As of JDK version 1.1, replaced by setBounds(int, int, int, int).* |
| void | [**resize**](http://docs.google.com/java/awt/Rectangle.html#resize(int,%20int))(int width, int height)  **Deprecated.** *As of JDK version 1.1, replaced by setSize(int, int).* |
| void | [**setBounds**](http://docs.google.com/java/awt/Rectangle.html#setBounds(int,%20int,%20int,%20int))(int x, int y, int width, int height)            Sets the bounding Rectangle of this Rectangle to the specified x, y, width, and height. |
| void | [**setBounds**](http://docs.google.com/java/awt/Rectangle.html#setBounds(java.awt.Rectangle))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)            Sets the bounding Rectangle of this Rectangle to match the specified Rectangle. |
| void | [**setLocation**](http://docs.google.com/java/awt/Rectangle.html#setLocation(int,%20int))(int x, int y)            Moves this Rectangle to the specified location. |
| void | [**setLocation**](http://docs.google.com/java/awt/Rectangle.html#setLocation(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) p)            Moves this Rectangle to the specified location. |
| void | [**setRect**](http://docs.google.com/java/awt/Rectangle.html#setRect(double,%20double,%20double,%20double))(double x, double y, double width, double height)            Sets the bounds of this Rectangle to the integer bounds which encompass the specified x, y, width, and height. |
| void | [**setSize**](http://docs.google.com/java/awt/Rectangle.html#setSize(java.awt.Dimension))([Dimension](http://docs.google.com/java/awt/Dimension.html) d)            Sets the size of this Rectangle to match the specified Dimension. |
| void | [**setSize**](http://docs.google.com/java/awt/Rectangle.html#setSize(int,%20int))(int width, int height)            Sets the size of this Rectangle to the specified width and height. |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/awt/Rectangle.html#toString())()            Returns a String representing this Rectangle and its values. |
| void | [**translate**](http://docs.google.com/java/awt/Rectangle.html#translate(int,%20int))(int dx, int dy)            Translates this Rectangle the indicated distance, to the right along the X coordinate axis, and downward along the Y coordinate axis. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**union**](http://docs.google.com/java/awt/Rectangle.html#union(java.awt.Rectangle))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)            Computes the union of this Rectangle with the specified Rectangle. |

| **Methods inherited from class java.awt.geom.**[**Rectangle2D**](http://docs.google.com/java/awt/geom/Rectangle2D.html) |
| --- |
| [add](http://docs.google.com/java/awt/geom/Rectangle2D.html#add(double,%20double)), [add](http://docs.google.com/java/awt/geom/Rectangle2D.html#add(java.awt.geom.Point2D)), [add](http://docs.google.com/java/awt/geom/Rectangle2D.html#add(java.awt.geom.Rectangle2D)), [contains](http://docs.google.com/java/awt/geom/Rectangle2D.html#contains(double,%20double)), [contains](http://docs.google.com/java/awt/geom/Rectangle2D.html#contains(double,%20double,%20double,%20double)), [getPathIterator](http://docs.google.com/java/awt/geom/Rectangle2D.html#getPathIterator(java.awt.geom.AffineTransform)), [getPathIterator](http://docs.google.com/java/awt/geom/Rectangle2D.html#getPathIterator(java.awt.geom.AffineTransform,%20double)), [hashCode](http://docs.google.com/java/awt/geom/Rectangle2D.html#hashCode()), [intersect](http://docs.google.com/java/awt/geom/Rectangle2D.html#intersect(java.awt.geom.Rectangle2D,%20java.awt.geom.Rectangle2D,%20java.awt.geom.Rectangle2D)), [intersects](http://docs.google.com/java/awt/geom/Rectangle2D.html#intersects(double,%20double,%20double,%20double)), [intersectsLine](http://docs.google.com/java/awt/geom/Rectangle2D.html#intersectsLine(double,%20double,%20double,%20double)), [intersectsLine](http://docs.google.com/java/awt/geom/Rectangle2D.html#intersectsLine(java.awt.geom.Line2D)), [outcode](http://docs.google.com/java/awt/geom/Rectangle2D.html#outcode(java.awt.geom.Point2D)), [setFrame](http://docs.google.com/java/awt/geom/Rectangle2D.html#setFrame(double,%20double,%20double,%20double)), [setRect](http://docs.google.com/java/awt/geom/Rectangle2D.html#setRect(java.awt.geom.Rectangle2D)), [union](http://docs.google.com/java/awt/geom/Rectangle2D.html#union(java.awt.geom.Rectangle2D,%20java.awt.geom.Rectangle2D,%20java.awt.geom.Rectangle2D)) |

| **Methods inherited from class java.awt.geom.**[**RectangularShape**](http://docs.google.com/java/awt/geom/RectangularShape.html) |
| --- |
| [clone](http://docs.google.com/java/awt/geom/RectangularShape.html#clone()), [contains](http://docs.google.com/java/awt/geom/RectangularShape.html#contains(java.awt.geom.Point2D)), [contains](http://docs.google.com/java/awt/geom/RectangularShape.html#contains(java.awt.geom.Rectangle2D)), [getCenterX](http://docs.google.com/java/awt/geom/RectangularShape.html#getCenterX()), [getCenterY](http://docs.google.com/java/awt/geom/RectangularShape.html#getCenterY()), [getFrame](http://docs.google.com/java/awt/geom/RectangularShape.html#getFrame()), [getMaxX](http://docs.google.com/java/awt/geom/RectangularShape.html#getMaxX()), [getMaxY](http://docs.google.com/java/awt/geom/RectangularShape.html#getMaxY()), [getMinX](http://docs.google.com/java/awt/geom/RectangularShape.html#getMinX()), [getMinY](http://docs.google.com/java/awt/geom/RectangularShape.html#getMinY()), [intersects](http://docs.google.com/java/awt/geom/RectangularShape.html#intersects(java.awt.geom.Rectangle2D)), [setFrame](http://docs.google.com/java/awt/geom/RectangularShape.html#setFrame(java.awt.geom.Point2D,%20java.awt.geom.Dimension2D)), [setFrame](http://docs.google.com/java/awt/geom/RectangularShape.html#setFrame(java.awt.geom.Rectangle2D)), [setFrameFromCenter](http://docs.google.com/java/awt/geom/RectangularShape.html#setFrameFromCenter(double,%20double,%20double,%20double)), [setFrameFromCenter](http://docs.google.com/java/awt/geom/RectangularShape.html#setFrameFromCenter(java.awt.geom.Point2D,%20java.awt.geom.Point2D)), [setFrameFromDiagonal](http://docs.google.com/java/awt/geom/RectangularShape.html#setFrameFromDiagonal(double,%20double,%20double,%20double)), [setFrameFromDiagonal](http://docs.google.com/java/awt/geom/RectangularShape.html#setFrameFromDiagonal(java.awt.geom.Point2D,%20java.awt.geom.Point2D)) |

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [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)) |

| **Methods inherited from interface java.awt.**[**Shape**](http://docs.google.com/java/awt/Shape.html) |
| --- |
| [contains](http://docs.google.com/java/awt/Shape.html#contains(double,%20double)), [contains](http://docs.google.com/java/awt/Shape.html#contains(double,%20double,%20double,%20double)), [contains](http://docs.google.com/java/awt/Shape.html#contains(java.awt.geom.Point2D)), [contains](http://docs.google.com/java/awt/Shape.html#contains(java.awt.geom.Rectangle2D)), [getPathIterator](http://docs.google.com/java/awt/Shape.html#getPathIterator(java.awt.geom.AffineTransform)), [getPathIterator](http://docs.google.com/java/awt/Shape.html#getPathIterator(java.awt.geom.AffineTransform,%20double)), [intersects](http://docs.google.com/java/awt/Shape.html#intersects(double,%20double,%20double,%20double)), [intersects](http://docs.google.com/java/awt/Shape.html#intersects(java.awt.geom.Rectangle2D)) |

| **Field Detail** |
| --- |

### x

public int **x**

The X coordinate of the upper-left corner of the Rectangle.

**Since:** 1.0 **See Also:**[setLocation(int, int)](http://docs.google.com/java/awt/Rectangle.html#setLocation(int,%20int)), [getLocation()](http://docs.google.com/java/awt/Rectangle.html#getLocation())

### y

public int **y**

The Y coordinate of the upper-left corner of the Rectangle.

**Since:** 1.0 **See Also:**[setLocation(int, int)](http://docs.google.com/java/awt/Rectangle.html#setLocation(int,%20int)), [getLocation()](http://docs.google.com/java/awt/Rectangle.html#getLocation())

### width

public int **width**

The width of the Rectangle.

**Since:** 1.0 **See Also:**[setSize(int, int)](http://docs.google.com/java/awt/Rectangle.html#setSize(int,%20int)), [getSize()](http://docs.google.com/java/awt/Rectangle.html#getSize())

### height

public int **height**

The height of the Rectangle.

**Since:** 1.0 **See Also:**[setSize(int, int)](http://docs.google.com/java/awt/Rectangle.html#setSize(int,%20int)), [getSize()](http://docs.google.com/java/awt/Rectangle.html#getSize())

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

### Rectangle

public **Rectangle**()

Constructs a new Rectangle whose upper-left corner is at (0, 0) in the coordinate space, and whose width and height are both zero.

### Rectangle

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

Constructs a new Rectangle, initialized to match the values of the specified Rectangle.

**Parameters:**r - the Rectangle from which to copy initial values to a newly constructed Rectangle**Since:** 1.1

### Rectangle

public **Rectangle**(int x,  
 int y,  
 int width,  
 int height)

Constructs a new Rectangle whose upper-left corner is specified as (x,y) and whose width and height are specified by the arguments of the same name.

**Parameters:**x - the specified X coordinatey - the specified Y coordinatewidth - the width of the Rectangleheight - the height of the Rectangle**Since:** 1.0

### Rectangle

public **Rectangle**(int width,  
 int height)

Constructs a new Rectangle whose upper-left corner is at (0, 0) in the coordinate space, and whose width and height are specified by the arguments of the same name.

**Parameters:**width - the width of the Rectangleheight - the height of the Rectangle

### Rectangle

public **Rectangle**([Point](http://docs.google.com/java/awt/Point.html) p,  
 [Dimension](http://docs.google.com/java/awt/Dimension.html) d)

Constructs a new Rectangle whose upper-left corner is specified by the [Point](http://docs.google.com/java/awt/Point.html) argument, and whose width and height are specified by the [Dimension](http://docs.google.com/java/awt/Dimension.html) argument.

**Parameters:**p - a Point that is the upper-left corner of the Rectangled - a Dimension, representing the width and height of the Rectangle

### Rectangle

public **Rectangle**([Point](http://docs.google.com/java/awt/Point.html) p)

Constructs a new Rectangle whose upper-left corner is the specified Point, and whose width and height are both zero.

**Parameters:**p - a Point that is the top left corner of the Rectangle

### Rectangle

public **Rectangle**([Dimension](http://docs.google.com/java/awt/Dimension.html) d)

Constructs a new Rectangle whose top left corner is (0, 0) and whose width and height are specified by the Dimension argument.

**Parameters:**d - a Dimension, specifying width and height

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

### getX

public double **getX**()

Returns the X coordinate of the bounding Rectangle in double precision.

**Specified by:**[getX](http://docs.google.com/java/awt/geom/RectangularShape.html#getX()) in class [RectangularShape](http://docs.google.com/java/awt/geom/RectangularShape.html) **Returns:**the X coordinate of the bounding Rectangle.

### getY

public double **getY**()

Returns the Y coordinate of the bounding Rectangle in double precision.

**Specified by:**[getY](http://docs.google.com/java/awt/geom/RectangularShape.html#getY()) in class [RectangularShape](http://docs.google.com/java/awt/geom/RectangularShape.html) **Returns:**the Y coordinate of the bounding Rectangle.

### getWidth

public double **getWidth**()

Returns the width of the bounding Rectangle in double precision.

**Specified by:**[getWidth](http://docs.google.com/java/awt/geom/RectangularShape.html#getWidth()) in class [RectangularShape](http://docs.google.com/java/awt/geom/RectangularShape.html) **Returns:**the width of the bounding Rectangle.

### getHeight

public double **getHeight**()

Returns the height of the bounding Rectangle in double precision.

**Specified by:**[getHeight](http://docs.google.com/java/awt/geom/RectangularShape.html#getHeight()) in class [RectangularShape](http://docs.google.com/java/awt/geom/RectangularShape.html) **Returns:**the height of the bounding Rectangle.

### getBounds

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

Gets the bounding Rectangle of this Rectangle.

This method is included for completeness, to parallel the getBounds method of [Component](http://docs.google.com/java/awt/Component.html).

**Specified by:**[getBounds](http://docs.google.com/java/awt/Shape.html#getBounds()) in interface [Shape](http://docs.google.com/java/awt/Shape.html)**Overrides:**[getBounds](http://docs.google.com/java/awt/geom/RectangularShape.html#getBounds()) in class [RectangularShape](http://docs.google.com/java/awt/geom/RectangularShape.html) **Returns:**a new Rectangle, equal to the bounding Rectangle for this Rectangle.**Since:** 1.1 **See Also:**[Component.getBounds()](http://docs.google.com/java/awt/Component.html#getBounds()), [setBounds(Rectangle)](http://docs.google.com/java/awt/Rectangle.html#setBounds(java.awt.Rectangle)), [setBounds(int, int, int, int)](http://docs.google.com/java/awt/Rectangle.html#setBounds(int,%20int,%20int,%20int))

### getBounds2D

public [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) **getBounds2D**()

Returns a high precision and more accurate bounding box of the Shape than the getBounds method. Note that there is no guarantee that the returned [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) is the smallest bounding box that encloses the Shape, only that the Shape lies entirely within the indicated Rectangle2D. The bounding box returned by this method is usually tighter than that returned by the getBounds method and never fails due to overflow problems since the return value can be an instance of the Rectangle2D that uses double precision values to store the dimensions.

**Specified by:**[getBounds2D](http://docs.google.com/java/awt/Shape.html#getBounds2D()) in interface [Shape](http://docs.google.com/java/awt/Shape.html)**Overrides:**[getBounds2D](http://docs.google.com/java/awt/geom/Rectangle2D.html#getBounds2D()) in class [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) **Returns:**an instance of Rectangle2D that is a high-precision bounding box of the Shape.**Since:** 1.2 **See Also:**[Shape.getBounds()](http://docs.google.com/java/awt/Shape.html#getBounds())

### setBounds

public void **setBounds**([Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)

Sets the bounding Rectangle of this Rectangle to match the specified Rectangle.

This method is included for completeness, to parallel the setBounds method of Component.

**Parameters:**r - the specified Rectangle**Since:** 1.1 **See Also:**[getBounds()](http://docs.google.com/java/awt/Rectangle.html#getBounds()), [Component.setBounds(java.awt.Rectangle)](http://docs.google.com/java/awt/Component.html#setBounds(java.awt.Rectangle))

### setBounds

public void **setBounds**(int x,  
 int y,  
 int width,  
 int height)

Sets the bounding Rectangle of this Rectangle to the specified x, y, width, and height.

This method is included for completeness, to parallel the setBounds method of Component.

**Parameters:**x - the new X coordinate for the upper-left corner of this Rectangley - the new Y coordinate for the upper-left corner of this Rectanglewidth - the new width for this Rectangleheight - the new height for this Rectangle**Since:** 1.1 **See Also:**[getBounds()](http://docs.google.com/java/awt/Rectangle.html#getBounds()), [Component.setBounds(int, int, int, int)](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%20int))

### setRect

public void **setRect**(double x,  
 double y,  
 double width,  
 double height)

Sets the bounds of this Rectangle to the integer bounds which encompass the specified x, y, width, and height. If the parameters specify a Rectangle that exceeds the maximum range of integers, the result will be the best representation of the specified Rectangle intersected with the maximum integer bounds.

**Specified by:**[setRect](http://docs.google.com/java/awt/geom/Rectangle2D.html#setRect(double,%20double,%20double,%20double)) in class [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) **Parameters:**x - the X coordinate of the upper-left corner of the specified rectangley - the Y coordinate of the upper-left corner of the specified rectanglewidth - the width of the specified rectangleheight - the new height of the specified rectangle

### reshape

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **reshape**(int x,  
 int y,  
 int width,  
 int height)

**Deprecated.** *As of JDK version 1.1, replaced by setBounds(int, int, int, int).*

Sets the bounding Rectangle of this Rectangle to the specified x, y, width, and height.

**Parameters:**x - the new X coordinate for the upper-left corner of this Rectangley - the new Y coordinate for the upper-left corner of this Rectanglewidth - the new width for this Rectangleheight - the new height for this Rectangle

### getLocation

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

Returns the location of this Rectangle.

This method is included for completeness, to parallel the getLocation method of Component.

**Returns:**the Point that is the upper-left corner of this Rectangle.**Since:** 1.1 **See Also:**[Component.getLocation()](http://docs.google.com/java/awt/Component.html#getLocation()), [setLocation(Point)](http://docs.google.com/java/awt/Rectangle.html#setLocation(java.awt.Point)), [setLocation(int, int)](http://docs.google.com/java/awt/Rectangle.html#setLocation(int,%20int))

### setLocation

public void **setLocation**([Point](http://docs.google.com/java/awt/Point.html) p)

Moves this Rectangle to the specified location.

This method is included for completeness, to parallel the setLocation method of Component.

**Parameters:**p - the Point specifying the new location for this Rectangle**Since:** 1.1 **See Also:**[Component.setLocation(java.awt.Point)](http://docs.google.com/java/awt/Component.html#setLocation(java.awt.Point)), [getLocation()](http://docs.google.com/java/awt/Rectangle.html#getLocation())

### setLocation

public void **setLocation**(int x,  
 int y)

Moves this Rectangle to the specified location.

This method is included for completeness, to parallel the setLocation method of Component.

**Parameters:**x - the X coordinate of the new locationy - the Y coordinate of the new location**Since:** 1.1 **See Also:**[getLocation()](http://docs.google.com/java/awt/Rectangle.html#getLocation()), [Component.setLocation(int, int)](http://docs.google.com/java/awt/Component.html#setLocation(int,%20int))

### move

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **move**(int x,  
 int y)

**Deprecated.** *As of JDK version 1.1, replaced by setLocation(int, int).*

Moves this Rectangle to the specified location.

**Parameters:**x - the X coordinate of the new locationy - the Y coordinate of the new location

### translate

public void **translate**(int dx,  
 int dy)

Translates this Rectangle the indicated distance, to the right along the X coordinate axis, and downward along the Y coordinate axis.

**Parameters:**dx - the distance to move this Rectangle along the X axisdy - the distance to move this Rectangle along the Y axis**See Also:**[setLocation(int, int)](http://docs.google.com/java/awt/Rectangle.html#setLocation(int,%20int)), [setLocation(java.awt.Point)](http://docs.google.com/java/awt/Rectangle.html#setLocation(java.awt.Point))

### getSize

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

Gets the size of this Rectangle, represented by the returned Dimension.

This method is included for completeness, to parallel the getSize method of Component.

**Returns:**a Dimension, representing the size of this Rectangle.**Since:** 1.1 **See Also:**[Component.getSize()](http://docs.google.com/java/awt/Component.html#getSize()), [setSize(Dimension)](http://docs.google.com/java/awt/Rectangle.html#setSize(java.awt.Dimension)), [setSize(int, int)](http://docs.google.com/java/awt/Rectangle.html#setSize(int,%20int))

### setSize

public void **setSize**([Dimension](http://docs.google.com/java/awt/Dimension.html) d)

Sets the size of this Rectangle to match the specified Dimension.

This method is included for completeness, to parallel the setSize method of Component.

**Parameters:**d - the new size for the Dimension object**Since:** 1.1 **See Also:**[Component.setSize(java.awt.Dimension)](http://docs.google.com/java/awt/Component.html#setSize(java.awt.Dimension)), [getSize()](http://docs.google.com/java/awt/Rectangle.html#getSize())

### setSize

public void **setSize**(int width,  
 int height)

Sets the size of this Rectangle to the specified width and height.

This method is included for completeness, to parallel the setSize method of Component.

**Parameters:**width - the new width for this Rectangleheight - the new height for this Rectangle**Since:** 1.1 **See Also:**[Component.setSize(int, int)](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [getSize()](http://docs.google.com/java/awt/Rectangle.html#getSize())

### resize

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **resize**(int width,  
 int height)

**Deprecated.** *As of JDK version 1.1, replaced by setSize(int, int).*

Sets the size of this Rectangle to the specified width and height.

**Parameters:**width - the new width for this Rectangleheight - the new height for this Rectangle

### contains

public boolean **contains**([Point](http://docs.google.com/java/awt/Point.html) p)

Checks whether or not this Rectangle contains the specified Point.

**Parameters:**p - the Point to test **Returns:**true if the specified Point is inside this Rectangle; false otherwise.**Since:** 1.1

### contains

public boolean **contains**(int x,  
 int y)

Checks whether or not this Rectangle contains the point at the specified location (x,y).

**Parameters:**x - the specified X coordinatey - the specified Y coordinate **Returns:**true if the point (x,y) is inside this Rectangle; false otherwise.**Since:** 1.1

### contains

public boolean **contains**([Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)

Checks whether or not this Rectangle entirely contains the specified Rectangle.

**Parameters:**r - the specified Rectangle **Returns:**true if the Rectangle is contained entirely inside this Rectangle; false otherwise**Since:** 1.2

### contains

public boolean **contains**(int X,  
 int Y,  
 int W,  
 int H)

Checks whether this Rectangle entirely contains the Rectangle at the specified location (X,Y) with the specified dimensions (W,H).

**Parameters:**X - the specified X coordinateY - the specified Y coordinateW - the width of the RectangleH - the height of the Rectangle **Returns:**true if the Rectangle specified by (X, Y, W, H) is entirely enclosed inside this Rectangle; false otherwise.**Since:** 1.1

### inside

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public boolean **inside**(int X,  
 int Y)

**Deprecated.** *As of JDK version 1.1, replaced by contains(int, int).*

Checks whether or not this Rectangle contains the point at the specified location (X,Y).

**Parameters:**X - the specified X coordinateY - the specified Y coordinate **Returns:**true if the point (X,Y) is inside this Rectangle; false otherwise.

### intersects

public boolean **intersects**([Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)

Determines whether or not this Rectangle and the specified Rectangle intersect. Two rectangles intersect if their intersection is nonempty.

**Parameters:**r - the specified Rectangle **Returns:**true if the specified Rectangle and this Rectangle intersect; false otherwise.

### intersection

public [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **intersection**([Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)

Computes the intersection of this Rectangle with the specified Rectangle. Returns a new Rectangle that represents the intersection of the two rectangles. If the two rectangles do not intersect, the result will be an empty rectangle.

**Parameters:**r - the specified Rectangle **Returns:**the largest Rectangle contained in both the specified Rectangle and in this Rectangle; or if the rectangles do not intersect, an empty rectangle.

### union

public [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **union**([Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)

Computes the union of this Rectangle with the specified Rectangle. Returns a new Rectangle that represents the union of the two rectangles.

If either Rectangle has any dimension less than zero the rules for [non-existant](#2et92p0) rectangles apply. If only one has a dimension less than zero, then the result will be a copy of the other Rectangle. If both have dimension less than zero, then the result will have at least one dimension less than zero.

If the resulting Rectangle would have a dimension too large to be expressed as an int, the result will have a dimension of Integer.MAX\_VALUE along that dimension.

**Parameters:**r - the specified Rectangle **Returns:**the smallest Rectangle containing both the specified Rectangle and this Rectangle.

### add

public void **add**(int newx,  
 int newy)

Adds a point, specified by the integer arguments newx,newy to the bounds of this Rectangle.

If this Rectangle has any dimension less than zero, the rules for [non-existant](#2et92p0) rectangles apply. In that case, the new bounds of this Rectangle will have a location equal to the specified coordinates and width and height equal to zero.

After adding a point, a call to contains with the added point as an argument does not necessarily return true. The contains method does not return true for points on the right or bottom edges of a Rectangle. Therefore, if the added point falls on the right or bottom edge of the enlarged Rectangle, contains returns false for that point. If the specified point must be contained within the new Rectangle, a 1x1 rectangle should be added instead:

r.add(newx, newy, 1, 1);

**Parameters:**newx - the X coordinate of the new pointnewy - the Y coordinate of the new point

### add

public void **add**([Point](http://docs.google.com/java/awt/Point.html) pt)

Adds the specified Point to the bounds of this Rectangle.

If this Rectangle has any dimension less than zero, the rules for [non-existant](#2et92p0) rectangles apply. In that case, the new bounds of this Rectangle will have a location equal to the coordinates of the specified Point and width and height equal to zero.

After adding a Point, a call to contains with the added Point as an argument does not necessarily return true. The contains method does not return true for points on the right or bottom edges of a Rectangle. Therefore if the added Point falls on the right or bottom edge of the enlarged Rectangle, contains returns false for that Point. If the specified point must be contained within the new Rectangle, a 1x1 rectangle should be added instead:

r.add(pt.x, pt.y, 1, 1);

**Parameters:**pt - the new Point to add to this Rectangle

### add

public void **add**([Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)

Adds a Rectangle to this Rectangle. The resulting Rectangle is the union of the two rectangles.

If either Rectangle has any dimension less than 0, the result will have the dimensions of the other Rectangle. If both Rectangles have at least one dimension less than 0, the result will have at least one dimension less than 0.

If either Rectangle has one or both dimensions equal to 0, the result along those axes with 0 dimensions will be equivalent to the results obtained by adding the corresponding origin coordinate to the result rectangle along that axis, similar to the operation of the [add(Point)](http://docs.google.com/java/awt/Rectangle.html#add(java.awt.Point)) method, but contribute no further dimension beyond that.

If the resulting Rectangle would have a dimension too large to be expressed as an int, the result will have a dimension of Integer.MAX\_VALUE along that dimension.

**Parameters:**r - the specified Rectangle

### grow

public void **grow**(int h,  
 int v)

Resizes the Rectangle both horizontally and vertically.

This method modifies the Rectangle so that it is h units larger on both the left and right side, and v units larger at both the top and bottom.

The new Rectangle has (x - h, y - v) as its upper-left corner, width of (width + 2h), and a height of (height + 2v).

If negative values are supplied for h and v, the size of the Rectangle decreases accordingly. The grow method will check for integer overflow and underflow, but does not check whether the resulting values of width and height grow from negative to non-negative or shrink from non-negative to negative.

**Parameters:**h - the horizontal expansionv - the vertical expansion

### isEmpty

public boolean **isEmpty**()

Determines whether the RectangularShape is empty. When the RectangularShape is empty, it encloses no area.

**Specified by:**[isEmpty](http://docs.google.com/java/awt/geom/RectangularShape.html#isEmpty()) in class [RectangularShape](http://docs.google.com/java/awt/geom/RectangularShape.html) **Returns:**true if the RectangularShape is empty; false otherwise.**Since:** 1.2

### outcode

public int **outcode**(double x,  
 double y)

Determines where the specified coordinates lie with respect to this Rectangle2D. This method computes a binary OR of the appropriate mask values indicating, for each side of this Rectangle2D, whether or not the specified coordinates are on the same side of the edge as the rest of this Rectangle2D.

**Specified by:**[outcode](http://docs.google.com/java/awt/geom/Rectangle2D.html#outcode(double,%20double)) in class [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) **Parameters:**x - the specified X coordinatey - the specified Y coordinate **Returns:**the logical OR of all appropriate out codes.**Since:** 1.2 **See Also:**[Rectangle2D.OUT\_LEFT](http://docs.google.com/java/awt/geom/Rectangle2D.html#OUT_LEFT), [Rectangle2D.OUT\_TOP](http://docs.google.com/java/awt/geom/Rectangle2D.html#OUT_TOP), [Rectangle2D.OUT\_RIGHT](http://docs.google.com/java/awt/geom/Rectangle2D.html#OUT_RIGHT), [Rectangle2D.OUT\_BOTTOM](http://docs.google.com/java/awt/geom/Rectangle2D.html#OUT_BOTTOM)

### createIntersection

public [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) **createIntersection**([Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) r)

Returns a new Rectangle2D object representing the intersection of this Rectangle2D with the specified Rectangle2D.

**Specified by:**[createIntersection](http://docs.google.com/java/awt/geom/Rectangle2D.html#createIntersection(java.awt.geom.Rectangle2D)) in class [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) **Parameters:**r - the Rectangle2D to be intersected with this Rectangle2D **Returns:**the largest Rectangle2D contained in both the specified Rectangle2D and in this Rectangle2D.**Since:** 1.2

### createUnion

public [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) **createUnion**([Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) r)

Returns a new Rectangle2D object representing the union of this Rectangle2D with the specified Rectangle2D.

**Specified by:**[createUnion](http://docs.google.com/java/awt/geom/Rectangle2D.html#createUnion(java.awt.geom.Rectangle2D)) in class [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) **Parameters:**r - the Rectangle2D to be combined with this Rectangle2D **Returns:**the smallest Rectangle2D containing both the specified Rectangle2D and this Rectangle2D.**Since:** 1.2

### equals

public boolean **equals**([Object](http://docs.google.com/java/lang/Object.html) obj)

Checks whether two rectangles are equal.

The result is true if and only if the argument is not null and is a Rectangle object that has the same upper-left corner, width, and height as this Rectangle.

**Overrides:**[equals](http://docs.google.com/java/awt/geom/Rectangle2D.html#equals(java.lang.Object)) in class [Rectangle2D](http://docs.google.com/java/awt/geom/Rectangle2D.html) **Parameters:**obj - the Object to compare with this Rectangle **Returns:**true if the objects are equal; false otherwise.**See Also:**[Object.hashCode()](http://docs.google.com/java/lang/Object.html#hashCode()), [Hashtable](http://docs.google.com/java/util/Hashtable.html)

### toString

public [String](http://docs.google.com/java/lang/String.html) **toString**()

Returns a String representing this Rectangle and its values.

**Overrides:**[toString](http://docs.google.com/java/lang/Object.html#toString()) in class [Object](http://docs.google.com/java/lang/Object.html) **Returns:**a String representing this Rectangle object's coordinate and size values.

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Rectangle.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/RadialGradientPaint.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/RenderingHints.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Rectangle.html)    [**NO FRAMES**](http://docs.google.com/Rectangle.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3dy6vkm) | [FIELD](#1t3h5sf) | [CONSTR](#2s8eyo1) | [METHOD](#17dp8vu) | DETAIL: [FIELD](#1ksv4uv) | [CONSTR](#1y810tw) | [METHOD](#1pxezwc) |

[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).