| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SwingUtilities.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/javax/swing/SwingConstants.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/SwingWorker.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/SwingUtilities.html)    [**NO FRAMES**](http://docs.google.com/SwingUtilities.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#1t3h5sf) |

## **javax.swing**

Class SwingUtilities

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.swing.SwingUtilities**

**All Implemented Interfaces:** [SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html)

public class **SwingUtilities**extends [Object](http://docs.google.com/java/lang/Object.html)implements [SwingConstants](http://docs.google.com/javax/swing/SwingConstants.html)

A collection of utility methods for Swing.

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

| **Fields inherited from interface javax.swing.**[**SwingConstants**](http://docs.google.com/javax/swing/SwingConstants.html) |
| --- |
| [BOTTOM](http://docs.google.com/javax/swing/SwingConstants.html#BOTTOM), [CENTER](http://docs.google.com/javax/swing/SwingConstants.html#CENTER), [EAST](http://docs.google.com/javax/swing/SwingConstants.html#EAST), [HORIZONTAL](http://docs.google.com/javax/swing/SwingConstants.html#HORIZONTAL), [LEADING](http://docs.google.com/javax/swing/SwingConstants.html#LEADING), [LEFT](http://docs.google.com/javax/swing/SwingConstants.html#LEFT), [NEXT](http://docs.google.com/javax/swing/SwingConstants.html#NEXT), [NORTH](http://docs.google.com/javax/swing/SwingConstants.html#NORTH), [NORTH\_EAST](http://docs.google.com/javax/swing/SwingConstants.html#NORTH_EAST), [NORTH\_WEST](http://docs.google.com/javax/swing/SwingConstants.html#NORTH_WEST), [PREVIOUS](http://docs.google.com/javax/swing/SwingConstants.html#PREVIOUS), [RIGHT](http://docs.google.com/javax/swing/SwingConstants.html#RIGHT), [SOUTH](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH), [SOUTH\_EAST](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH_EAST), [SOUTH\_WEST](http://docs.google.com/javax/swing/SwingConstants.html#SOUTH_WEST), [TOP](http://docs.google.com/javax/swing/SwingConstants.html#TOP), [TRAILING](http://docs.google.com/javax/swing/SwingConstants.html#TRAILING), [VERTICAL](http://docs.google.com/javax/swing/SwingConstants.html#VERTICAL), [WEST](http://docs.google.com/javax/swing/SwingConstants.html#WEST) |

| **Method Summary** | |
| --- | --- |
| static [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**calculateInnerArea**](http://docs.google.com/javax/swing/SwingUtilities.html#calculateInnerArea(javax.swing.JComponent,%20java.awt.Rectangle))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)            Stores the position and size of the inner painting area of the specified component in r and returns r. |
| static [Rectangle](http://docs.google.com/java/awt/Rectangle.html)[] | [**computeDifference**](http://docs.google.com/javax/swing/SwingUtilities.html#computeDifference(java.awt.Rectangle,%20java.awt.Rectangle))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) rectA, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) rectB)            Convenience returning an array of rect representing the regions within rectA that do not overlap with rectB. |
| static [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**computeIntersection**](http://docs.google.com/javax/swing/SwingUtilities.html#computeIntersection(int,%20int,%20int,%20int,%20java.awt.Rectangle))(int x, int y, int width, int height, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) dest)            Convenience to calculate the intersection of two rectangles without allocating a new rectangle. |
| static int | [**computeStringWidth**](http://docs.google.com/javax/swing/SwingUtilities.html#computeStringWidth(java.awt.FontMetrics,%20java.lang.String))([FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) fm, [String](http://docs.google.com/java/lang/String.html) str)            Compute the width of the string using a font with the specified "metrics" (sizes). |
| static [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**computeUnion**](http://docs.google.com/javax/swing/SwingUtilities.html#computeUnion(int,%20int,%20int,%20int,%20java.awt.Rectangle))(int x, int y, int width, int height, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) dest)            Convenience method that calculates the union of two rectangles without allocating a new rectangle. |
| static [MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) | [**convertMouseEvent**](http://docs.google.com/javax/swing/SwingUtilities.html#convertMouseEvent(java.awt.Component,%20java.awt.event.MouseEvent,%20java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) source, [MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) sourceEvent, [Component](http://docs.google.com/java/awt/Component.html) destination)            Returns a MouseEvent similar to sourceEvent except that its x and y members have been converted to destination's coordinate system. |
| static [Point](http://docs.google.com/java/awt/Point.html) | [**convertPoint**](http://docs.google.com/javax/swing/SwingUtilities.html#convertPoint(java.awt.Component,%20int,%20int,%20java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) source, int x, int y, [Component](http://docs.google.com/java/awt/Component.html) destination)            Convert the point (x,y) in source coordinate system to destination coordinate system. |
| static [Point](http://docs.google.com/java/awt/Point.html) | [**convertPoint**](http://docs.google.com/javax/swing/SwingUtilities.html#convertPoint(java.awt.Component,%20java.awt.Point,%20java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) source, [Point](http://docs.google.com/java/awt/Point.html) aPoint, [Component](http://docs.google.com/java/awt/Component.html) destination)            Convert a aPoint in source coordinate system to destination coordinate system. |
| static void | [**convertPointFromScreen**](http://docs.google.com/javax/swing/SwingUtilities.html#convertPointFromScreen(java.awt.Point,%20java.awt.Component))([Point](http://docs.google.com/java/awt/Point.html) p, [Component](http://docs.google.com/java/awt/Component.html) c)            Convert a point from a screen coordinates to a component's coordinate system |
| static void | [**convertPointToScreen**](http://docs.google.com/javax/swing/SwingUtilities.html#convertPointToScreen(java.awt.Point,%20java.awt.Component))([Point](http://docs.google.com/java/awt/Point.html) p, [Component](http://docs.google.com/java/awt/Component.html) c)            Convert a point from a component's coordinate system to screen coordinates. |
| static [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**convertRectangle**](http://docs.google.com/javax/swing/SwingUtilities.html#convertRectangle(java.awt.Component,%20java.awt.Rectangle,%20java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) source, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) aRectangle, [Component](http://docs.google.com/java/awt/Component.html) destination)            Convert the rectangle aRectangle in source coordinate system to destination coordinate system. |
| static [Component](http://docs.google.com/java/awt/Component.html) | [**findFocusOwner**](http://docs.google.com/javax/swing/SwingUtilities.html#findFocusOwner(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)  **Deprecated.** *As of 1.4, replaced by KeyboardFocusManager.getFocusOwner().* |
| static [Accessible](http://docs.google.com/javax/accessibility/Accessible.html) | [**getAccessibleAt**](http://docs.google.com/javax/swing/SwingUtilities.html#getAccessibleAt(java.awt.Component,%20java.awt.Point))([Component](http://docs.google.com/java/awt/Component.html) c, [Point](http://docs.google.com/java/awt/Point.html) p)            Returns the Accessible child contained at the local coordinate Point, if one exists. |
| static [Accessible](http://docs.google.com/javax/accessibility/Accessible.html) | [**getAccessibleChild**](http://docs.google.com/javax/swing/SwingUtilities.html#getAccessibleChild(java.awt.Component,%20int))([Component](http://docs.google.com/java/awt/Component.html) c, int i)            Return the nth Accessible child of the object. |
| static int | [**getAccessibleChildrenCount**](http://docs.google.com/javax/swing/SwingUtilities.html#getAccessibleChildrenCount(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Returns the number of accessible children in the object. |
| static int | [**getAccessibleIndexInParent**](http://docs.google.com/javax/swing/SwingUtilities.html#getAccessibleIndexInParent(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Get the index of this object in its accessible parent. |
| static [AccessibleStateSet](http://docs.google.com/javax/accessibility/AccessibleStateSet.html) | [**getAccessibleStateSet**](http://docs.google.com/javax/swing/SwingUtilities.html#getAccessibleStateSet(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Get the state of this object. |
| static [Container](http://docs.google.com/java/awt/Container.html) | [**getAncestorNamed**](http://docs.google.com/javax/swing/SwingUtilities.html#getAncestorNamed(java.lang.String,%20java.awt.Component))([String](http://docs.google.com/java/lang/String.html) name, [Component](http://docs.google.com/java/awt/Component.html) comp)            Convenience method for searching above comp in the component hierarchy and returns the first object of name it finds. |
| static [Container](http://docs.google.com/java/awt/Container.html) | [**getAncestorOfClass**](http://docs.google.com/javax/swing/SwingUtilities.html#getAncestorOfClass(java.lang.Class,%20java.awt.Component))([Class](http://docs.google.com/java/lang/Class.html)<?> c, [Component](http://docs.google.com/java/awt/Component.html) comp)            Convenience method for searching above comp in the component hierarchy and returns the first object of class c it finds. |
| static [Component](http://docs.google.com/java/awt/Component.html) | [**getDeepestComponentAt**](http://docs.google.com/javax/swing/SwingUtilities.html#getDeepestComponentAt(java.awt.Component,%20int,%20int))([Component](http://docs.google.com/java/awt/Component.html) parent, int x, int y)            Returns the deepest visible descendent Component of parent that contains the location x, y. |
| static [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getLocalBounds**](http://docs.google.com/javax/swing/SwingUtilities.html#getLocalBounds(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) aComponent)            Return the rectangle (0,0,bounds.width,bounds.height) for the component aComponent |
| static [Component](http://docs.google.com/java/awt/Component.html) | [**getRoot**](http://docs.google.com/javax/swing/SwingUtilities.html#getRoot(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Returns the root component for the current component tree. |
| static [JRootPane](http://docs.google.com/javax/swing/JRootPane.html) | [**getRootPane**](http://docs.google.com/javax/swing/SwingUtilities.html#getRootPane(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            If c is a JRootPane descendant return its JRootPane ancestor. |
| static [ActionMap](http://docs.google.com/javax/swing/ActionMap.html) | [**getUIActionMap**](http://docs.google.com/javax/swing/SwingUtilities.html#getUIActionMap(javax.swing.JComponent))([JComponent](http://docs.google.com/javax/swing/JComponent.html) component)            Returns the ActionMap provided by the UI in component component. |
| static [InputMap](http://docs.google.com/javax/swing/InputMap.html) | [**getUIInputMap**](http://docs.google.com/javax/swing/SwingUtilities.html#getUIInputMap(javax.swing.JComponent,%20int))([JComponent](http://docs.google.com/javax/swing/JComponent.html) component, int condition)            Returns the InputMap provided by the UI for condition condition in component component. |
| static [Window](http://docs.google.com/java/awt/Window.html) | [**getWindowAncestor**](http://docs.google.com/javax/swing/SwingUtilities.html#getWindowAncestor(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Returns the first Window ancestor of c, or null if c is not contained inside a Window. |
| static void | [**invokeAndWait**](http://docs.google.com/javax/swing/SwingUtilities.html#invokeAndWait(java.lang.Runnable))([Runnable](http://docs.google.com/java/lang/Runnable.html) doRun)            Causes doRun.run() to be executed synchronously on the AWT event dispatching thread. |
| static void | [**invokeLater**](http://docs.google.com/javax/swing/SwingUtilities.html#invokeLater(java.lang.Runnable))([Runnable](http://docs.google.com/java/lang/Runnable.html) doRun)            Causes *doRun.run()* to be executed asynchronously on the AWT event dispatching thread. |
| static boolean | [**isDescendingFrom**](http://docs.google.com/javax/swing/SwingUtilities.html#isDescendingFrom(java.awt.Component,%20java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) a, [Component](http://docs.google.com/java/awt/Component.html) b)            Return true if a component a descends from a component b |
| static boolean | [**isEventDispatchThread**](http://docs.google.com/javax/swing/SwingUtilities.html#isEventDispatchThread())()            Returns true if the current thread is an AWT event dispatching thread. |
| static boolean | [**isLeftMouseButton**](http://docs.google.com/javax/swing/SwingUtilities.html#isLeftMouseButton(java.awt.event.MouseEvent))([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) anEvent)            Returns true if the mouse event specifies the left mouse button. |
| static boolean | [**isMiddleMouseButton**](http://docs.google.com/javax/swing/SwingUtilities.html#isMiddleMouseButton(java.awt.event.MouseEvent))([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) anEvent)            Returns true if the mouse event specifies the middle mouse button. |
| static boolean | [**isRectangleContainingRectangle**](http://docs.google.com/javax/swing/SwingUtilities.html#isRectangleContainingRectangle(java.awt.Rectangle,%20java.awt.Rectangle))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) a, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) b)            Return true if a contains b |
| static boolean | [**isRightMouseButton**](http://docs.google.com/javax/swing/SwingUtilities.html#isRightMouseButton(java.awt.event.MouseEvent))([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) anEvent)            Returns true if the mouse event specifies the right mouse button. |
| static [String](http://docs.google.com/java/lang/String.html) | [**layoutCompoundLabel**](http://docs.google.com/javax/swing/SwingUtilities.html#layoutCompoundLabel(java.awt.FontMetrics,%20java.lang.String,%20javax.swing.Icon,%20int,%20int,%20int,%20int,%20java.awt.Rectangle,%20java.awt.Rectangle,%20java.awt.Rectangle,%20int))([FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) fm, [String](http://docs.google.com/java/lang/String.html) text, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, int verticalAlignment, int horizontalAlignment, int verticalTextPosition, int horizontalTextPosition, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) viewR, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconR, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textR, int textIconGap)            Compute and return the location of the icons origin, the location of origin of the text baseline, and a possibly clipped version of the compound labels string. |
| static [String](http://docs.google.com/java/lang/String.html) | [**layoutCompoundLabel**](http://docs.google.com/javax/swing/SwingUtilities.html#layoutCompoundLabel(javax.swing.JComponent,%20java.awt.FontMetrics,%20java.lang.String,%20javax.swing.Icon,%20int,%20int,%20int,%20int,%20java.awt.Rectangle,%20java.awt.Rectangle,%20java.awt.Rectangle,%20int))([JComponent](http://docs.google.com/javax/swing/JComponent.html) c, [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) fm, [String](http://docs.google.com/java/lang/String.html) text, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, int verticalAlignment, int horizontalAlignment, int verticalTextPosition, int horizontalTextPosition, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) viewR, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconR, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textR, int textIconGap)            Compute and return the location of the icons origin, the location of origin of the text baseline, and a possibly clipped version of the compound labels string. |
| static boolean | [**notifyAction**](http://docs.google.com/javax/swing/SwingUtilities.html#notifyAction(javax.swing.Action,%20javax.swing.KeyStroke,%20java.awt.event.KeyEvent,%20java.lang.Object,%20int))([Action](http://docs.google.com/javax/swing/Action.html) action, [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) ks, [KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) event, [Object](http://docs.google.com/java/lang/Object.html) sender, int modifiers)            Invokes actionPerformed on action if action is enabled (and non-null). |
| static void | [**paintComponent**](http://docs.google.com/javax/swing/SwingUtilities.html#paintComponent(java.awt.Graphics,%20java.awt.Component,%20java.awt.Container,%20int,%20int,%20int,%20int))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [Component](http://docs.google.com/java/awt/Component.html) c, [Container](http://docs.google.com/java/awt/Container.html) p, int x, int y, int w, int h)            Paints a component to the specified Graphics. |
| static void | [**paintComponent**](http://docs.google.com/javax/swing/SwingUtilities.html#paintComponent(java.awt.Graphics,%20java.awt.Component,%20java.awt.Container,%20java.awt.Rectangle))([Graphics](http://docs.google.com/java/awt/Graphics.html) g, [Component](http://docs.google.com/java/awt/Component.html) c, [Container](http://docs.google.com/java/awt/Container.html) p, [Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)            Paints a component to the specified Graphics. |
| static boolean | [**processKeyBindings**](http://docs.google.com/javax/swing/SwingUtilities.html#processKeyBindings(java.awt.event.KeyEvent))([KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) event)            Process the key bindings for the Component associated with event. |
| static void | [**replaceUIActionMap**](http://docs.google.com/javax/swing/SwingUtilities.html#replaceUIActionMap(javax.swing.JComponent,%20javax.swing.ActionMap))([JComponent](http://docs.google.com/javax/swing/JComponent.html) component, [ActionMap](http://docs.google.com/javax/swing/ActionMap.html) uiActionMap)            Convenience method to change the UI ActionMap for component to uiActionMap. |
| static void | [**replaceUIInputMap**](http://docs.google.com/javax/swing/SwingUtilities.html#replaceUIInputMap(javax.swing.JComponent,%20int,%20javax.swing.InputMap))([JComponent](http://docs.google.com/javax/swing/JComponent.html) component, int type, [InputMap](http://docs.google.com/javax/swing/InputMap.html) uiInputMap)            Convenience method to change the UI InputMap for component to uiInputMap. |
| static void | [**updateComponentTreeUI**](http://docs.google.com/javax/swing/SwingUtilities.html#updateComponentTreeUI(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            A simple minded look and feel change: ask each node in the tree to updateUI() -- that is, to initialize its UI property with the current look and feel. |
| static [Window](http://docs.google.com/java/awt/Window.html) | [**windowForComponent**](http://docs.google.com/javax/swing/SwingUtilities.html#windowForComponent(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Returns the first Window ancestor of c, or null if c is not contained inside a Window. |

| **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)) |

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

### isRectangleContainingRectangle

public static final boolean **isRectangleContainingRectangle**([Rectangle](http://docs.google.com/java/awt/Rectangle.html) a,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) b)

Return true if a contains b

### getLocalBounds

public static [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **getLocalBounds**([Component](http://docs.google.com/java/awt/Component.html) aComponent)

Return the rectangle (0,0,bounds.width,bounds.height) for the component aComponent

### getWindowAncestor

public static [Window](http://docs.google.com/java/awt/Window.html) **getWindowAncestor**([Component](http://docs.google.com/java/awt/Component.html) c)

Returns the first Window ancestor of c, or null if c is not contained inside a Window.

**Parameters:**c - Component to get Window ancestor of. **Returns:**the first Window ancestor of c, or null if c is not contained inside a Window.**Since:** 1.3

### convertPoint

public static [Point](http://docs.google.com/java/awt/Point.html) **convertPoint**([Component](http://docs.google.com/java/awt/Component.html) source,  
 [Point](http://docs.google.com/java/awt/Point.html) aPoint,  
 [Component](http://docs.google.com/java/awt/Component.html) destination)

Convert a aPoint in source coordinate system to destination coordinate system. If source is null, aPoint is assumed to be in destination's root component coordinate system. If destination is null, aPoint will be converted to source's root component coordinate system. If both source and destination are null, return aPoint without any conversion.

### convertPoint

public static [Point](http://docs.google.com/java/awt/Point.html) **convertPoint**([Component](http://docs.google.com/java/awt/Component.html) source,  
 int x,  
 int y,  
 [Component](http://docs.google.com/java/awt/Component.html) destination)

Convert the point (x,y) in source coordinate system to destination coordinate system. If source is null, (x,y) is assumed to be in destination's root component coordinate system. If destination is null, (x,y) will be converted to source's root component coordinate system. If both source and destination are null, return (x,y) without any conversion.

### convertRectangle

public static [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **convertRectangle**([Component](http://docs.google.com/java/awt/Component.html) source,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) aRectangle,  
 [Component](http://docs.google.com/java/awt/Component.html) destination)

Convert the rectangle aRectangle in source coordinate system to destination coordinate system. If source is null, aRectangle is assumed to be in destination's root component coordinate system. If destination is null, aRectangle will be converted to source's root component coordinate system. If both source and destination are null, return aRectangle without any conversion.

### getAncestorOfClass

public static [Container](http://docs.google.com/java/awt/Container.html) **getAncestorOfClass**([Class](http://docs.google.com/java/lang/Class.html)<?> c,  
 [Component](http://docs.google.com/java/awt/Component.html) comp)

Convenience method for searching above comp in the component hierarchy and returns the first object of class c it finds. Can return null, if a class c cannot be found.

### getAncestorNamed

public static [Container](http://docs.google.com/java/awt/Container.html) **getAncestorNamed**([String](http://docs.google.com/java/lang/String.html) name,  
 [Component](http://docs.google.com/java/awt/Component.html) comp)

Convenience method for searching above comp in the component hierarchy and returns the first object of name it finds. Can return null, if name cannot be found.

### getDeepestComponentAt

public static [Component](http://docs.google.com/java/awt/Component.html) **getDeepestComponentAt**([Component](http://docs.google.com/java/awt/Component.html) parent,  
 int x,  
 int y)

Returns the deepest visible descendent Component of parent that contains the location x, y. If parent does not contain the specified location, then null is returned. If parent is not a container, or none of parent's visible descendents contain the specified location, parent is returned.

**Parameters:**parent - the root component to begin the searchx - the x target locationy - the y target location

### convertMouseEvent

public static [MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) **convertMouseEvent**([Component](http://docs.google.com/java/awt/Component.html) source,  
 [MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) sourceEvent,  
 [Component](http://docs.google.com/java/awt/Component.html) destination)

Returns a MouseEvent similar to sourceEvent except that its x and y members have been converted to destination's coordinate system. If source is null, sourceEvent x and y members are assumed to be into destination's root component coordinate system. If destination is null, the returned MouseEvent will be in source's coordinate system. sourceEvent will not be changed. A new event is returned. the source field of the returned event will be set to destination if destination is non-null use the translateMouseEvent() method to translate a mouse event from one component to another without changing the source.

### convertPointToScreen

public static void **convertPointToScreen**([Point](http://docs.google.com/java/awt/Point.html) p,  
 [Component](http://docs.google.com/java/awt/Component.html) c)

Convert a point from a component's coordinate system to screen coordinates.

**Parameters:**p - a Point object (converted to the new coordinate system)c - a Component object

### convertPointFromScreen

public static void **convertPointFromScreen**([Point](http://docs.google.com/java/awt/Point.html) p,  
 [Component](http://docs.google.com/java/awt/Component.html) c)

Convert a point from a screen coordinates to a component's coordinate system

**Parameters:**p - a Point object (converted to the new coordinate system)c - a Component object

### windowForComponent

public static [Window](http://docs.google.com/java/awt/Window.html) **windowForComponent**([Component](http://docs.google.com/java/awt/Component.html) c)

Returns the first Window ancestor of c, or null if c is not contained inside a Window.

Note: This method provides the same functionality as getWindowAncestor.

**Parameters:**c - Component to get Window ancestor of. **Returns:**the first Window ancestor of c, or null if c is not contained inside a Window.

### isDescendingFrom

public static boolean **isDescendingFrom**([Component](http://docs.google.com/java/awt/Component.html) a,  
 [Component](http://docs.google.com/java/awt/Component.html) b)

Return true if a component a descends from a component b

### computeIntersection

public static [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **computeIntersection**(int x,  
 int y,  
 int width,  
 int height,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) dest)

Convenience to calculate the intersection of two rectangles without allocating a new rectangle. If the two rectangles don't intersect, then the returned rectangle begins at (0,0) and has zero width and height.

**Parameters:**x - the X coordinate of the first rectangle's top-left pointy - the Y coordinate of the first rectangle's top-left pointwidth - the width of the first rectangleheight - the height of the first rectangledest - the second rectangle **Returns:**dest, modified to specify the intersection

### computeUnion

public static [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **computeUnion**(int x,  
 int y,  
 int width,  
 int height,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) dest)

Convenience method that calculates the union of two rectangles without allocating a new rectangle.

**Parameters:**x - the x-coordinate of the first rectangley - the y-coordinate of the first rectanglewidth - the width of the first rectangleheight - the height of the first rectangledest - the coordinates of the second rectangle; the union of the two rectangles is returned in this rectangle **Returns:**the dest Rectangle

### computeDifference

public static [Rectangle](http://docs.google.com/java/awt/Rectangle.html)[] **computeDifference**([Rectangle](http://docs.google.com/java/awt/Rectangle.html) rectA,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) rectB)

Convenience returning an array of rect representing the regions within rectA that do not overlap with rectB. If the two Rects do not overlap, returns an empty array

### isLeftMouseButton

public static boolean **isLeftMouseButton**([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) anEvent)

Returns true if the mouse event specifies the left mouse button.

**Parameters:**anEvent - a MouseEvent object **Returns:**true if the left mouse button was active

### isMiddleMouseButton

public static boolean **isMiddleMouseButton**([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) anEvent)

Returns true if the mouse event specifies the middle mouse button.

**Parameters:**anEvent - a MouseEvent object **Returns:**true if the middle mouse button was active

### isRightMouseButton

public static boolean **isRightMouseButton**([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) anEvent)

Returns true if the mouse event specifies the right mouse button.

**Parameters:**anEvent - a MouseEvent object **Returns:**true if the right mouse button was active

### computeStringWidth

public static int **computeStringWidth**([FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) fm,  
 [String](http://docs.google.com/java/lang/String.html) str)

Compute the width of the string using a font with the specified "metrics" (sizes).

**Parameters:**fm - a FontMetrics object to compute withstr - the String to compute **Returns:**an int containing the string width

### layoutCompoundLabel

public static [String](http://docs.google.com/java/lang/String.html) **layoutCompoundLabel**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c,  
 [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) fm,  
 [String](http://docs.google.com/java/lang/String.html) text,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 int verticalAlignment,  
 int horizontalAlignment,  
 int verticalTextPosition,  
 int horizontalTextPosition,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) viewR,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconR,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textR,  
 int textIconGap)

Compute and return the location of the icons origin, the location of origin of the text baseline, and a possibly clipped version of the compound labels string. Locations are computed relative to the viewR rectangle. The JComponents orientation (LEADING/TRAILING) will also be taken into account and translated into LEFT/RIGHT values accordingly.

### layoutCompoundLabel

public static [String](http://docs.google.com/java/lang/String.html) **layoutCompoundLabel**([FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) fm,  
 [String](http://docs.google.com/java/lang/String.html) text,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 int verticalAlignment,  
 int horizontalAlignment,  
 int verticalTextPosition,  
 int horizontalTextPosition,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) viewR,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) iconR,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) textR,  
 int textIconGap)

Compute and return the location of the icons origin, the location of origin of the text baseline, and a possibly clipped version of the compound labels string. Locations are computed relative to the viewR rectangle. This layoutCompoundLabel() does not know how to handle LEADING/TRAILING values in horizontalTextPosition (they will default to RIGHT) and in horizontalAlignment (they will default to CENTER). Use the other version of layoutCompoundLabel() instead.

### paintComponent

public static void **paintComponent**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [Component](http://docs.google.com/java/awt/Component.html) c,  
 [Container](http://docs.google.com/java/awt/Container.html) p,  
 int x,  
 int y,  
 int w,  
 int h)

Paints a component to the specified Graphics. This method is primarily useful to render Components that don't exist as part of the visible containment hierarchy, but are used for rendering. For example, if you are doing your own rendering and want to render some text (or even HTML), you could make use of JLabel's text rendering support and have it paint directly by way of this method, without adding the label to the visible containment hierarchy.

This method makes use of CellRendererPane to handle the actual painting, and is only recommended if you use one component for rendering. If you make use of multiple components to handle the rendering, as JTable does, use CellRendererPane directly. Otherwise, as described below, you could end up with a CellRendererPane per Component.

If c's parent is not a CellRendererPane, a new CellRendererPane is created, c is added to it, and the CellRendererPane is added to p. If c's parent is a CellRendererPane and the CellRendererPanes parent is not p, it is added to p.

The component should either descend from JComponent or be another kind of lightweight component. A lightweight component is one whose "lightweight" property (returned by the Component isLightweight method) is true. If the Component is not lightweight, bad things map happen: crashes, exceptions, painting problems...

**Parameters:**g - the Graphics object to draw onc - the Component to drawp - the intermediate Containerx - an int specifying the left side of the area draw in, in pixels, measured from the left edge of the graphics contexty - an int specifying the top of the area to draw in, in pixels measured down from the top edge of the graphics contextw - an int specifying the width of the area draw in, in pixelsh - an int specifying the height of the area draw in, in pixels**See Also:**[CellRendererPane](http://docs.google.com/javax/swing/CellRendererPane.html), [Component.isLightweight()](http://docs.google.com/java/awt/Component.html#isLightweight())

### paintComponent

public static void **paintComponent**([Graphics](http://docs.google.com/java/awt/Graphics.html) g,  
 [Component](http://docs.google.com/java/awt/Component.html) c,  
 [Container](http://docs.google.com/java/awt/Container.html) p,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)

Paints a component to the specified Graphics. This is a cover method for [paintComponent(Graphics,Component,Container,int,int,int,int)](http://docs.google.com/javax/swing/SwingUtilities.html#paintComponent(java.awt.Graphics,%20java.awt.Component,%20java.awt.Container,%20int,%20int,%20int,%20int)). Refer to it for more information.

**Parameters:**g - the Graphics object to draw onc - the Component to drawp - the intermediate Containerr - the Rectangle to draw in**See Also:**[paintComponent(Graphics,Component,Container,int,int,int,int)](http://docs.google.com/javax/swing/SwingUtilities.html#paintComponent(java.awt.Graphics,%20java.awt.Component,%20java.awt.Container,%20int,%20int,%20int,%20int)), [CellRendererPane](http://docs.google.com/javax/swing/CellRendererPane.html)

### updateComponentTreeUI

public static void **updateComponentTreeUI**([Component](http://docs.google.com/java/awt/Component.html) c)

A simple minded look and feel change: ask each node in the tree to updateUI() -- that is, to initialize its UI property with the current look and feel.

### invokeLater

public static void **invokeLater**([Runnable](http://docs.google.com/java/lang/Runnable.html) doRun)

Causes *doRun.run()* to be executed asynchronously on the AWT event dispatching thread. This will happen after all pending AWT events have been processed. This method should be used when an application thread needs to update the GUI. In the following example the invokeLater call queues the Runnable object doHelloWorld on the event dispatching thread and then prints a message.

Runnable doHelloWorld = new Runnable() {  
 public void run() {  
 System.out.println("Hello World on " + Thread.currentThread());  
 }  
 };  
  
 SwingUtilities.invokeLater(doHelloWorld);  
 System.out.println("This might well be displayed before the other message.");

If invokeLater is called from the event dispatching thread -- for example, from a JButton's ActionListener -- the *doRun.run()* will still be deferred until all pending events have been processed. Note that if the *doRun.run()* throws an uncaught exception the event dispatching thread will unwind (not the current thread).

Additional documentation and examples for this method can be found in [How to Use Threads](http://java.sun.com/docs/books/tutorial/uiswing/misc/threads.html), in *The Java Tutorial*.

As of 1.3 this method is just a cover for java.awt.EventQueue.invokeLater().

Unlike the rest of Swing, this method can be invoked from any thread.

**See Also:**[invokeAndWait(java.lang.Runnable)](http://docs.google.com/javax/swing/SwingUtilities.html#invokeAndWait(java.lang.Runnable))

### invokeAndWait

public static void **invokeAndWait**([Runnable](http://docs.google.com/java/lang/Runnable.html) doRun)  
 throws [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html),  
 [InvocationTargetException](http://docs.google.com/java/lang/reflect/InvocationTargetException.html)

Causes doRun.run() to be executed synchronously on the AWT event dispatching thread. This call blocks until all pending AWT events have been processed and (then) doRun.run() returns. This method should be used when an application thread needs to update the GUI. It should'nt be called from the EventDispatchThread. Here's an example that creates a new application thread that uses invokeAndWait to print a string from the event dispatching thread and then, when that's finished, print a string from the application thread.

final Runnable doHelloWorld = new Runnable() {  
 public void run() {  
 System.out.println("Hello World on " + Thread.currentThread());  
 }  
 };  
  
 Thread appThread = new Thread() {  
 public void run() {  
 try {  
 SwingUtilities.invokeAndWait(doHelloWorld);  
 }  
 catch (Exception e) {  
 e.printStackTrace();  
 }  
 System.out.println("Finished on " + Thread.currentThread());  
 }  
 };  
 appThread.start();

Note that if the Runnable.run method throws an uncaught exception (on the event dispatching thread) it's caught and rethrown, as an InvocationTargetException, on the caller's thread.

Additional documentation and examples for this method can be found in [How to Use Threads](http://java.sun.com/docs/books/tutorial/uiswing/misc/threads.html), in *The Java Tutorial*.

As of 1.3 this method is just a cover for java.awt.EventQueue.invokeAndWait().

**Throws:** [InterruptedException](http://docs.google.com/java/lang/InterruptedException.html) - if we're interrupted while waiting for the event dispatching thread to finish excecuting doRun.run() [InvocationTargetException](http://docs.google.com/java/lang/reflect/InvocationTargetException.html) - if an exception is thrown while running doRun**See Also:**[invokeLater(java.lang.Runnable)](http://docs.google.com/javax/swing/SwingUtilities.html#invokeLater(java.lang.Runnable))

### isEventDispatchThread

public static boolean **isEventDispatchThread**()

Returns true if the current thread is an AWT event dispatching thread.

As of 1.3 this method is just a cover for java.awt.EventQueue.isDispatchThread().

**Returns:**true if the current thread is an AWT event dispatching thread

### getAccessibleIndexInParent

public static int **getAccessibleIndexInParent**([Component](http://docs.google.com/java/awt/Component.html) c)

Get the index of this object in its accessible parent.

Note: as of the Java 2 platform v1.3, it is recommended that developers call Component.AccessibleAWTComponent.getAccessibleIndexInParent() instead of using this method.

**Returns:**-1 of this object does not have an accessible parent. Otherwise, the index of the child in its accessible parent.

### getAccessibleAt

public static [Accessible](http://docs.google.com/javax/accessibility/Accessible.html) **getAccessibleAt**([Component](http://docs.google.com/java/awt/Component.html) c,  
 [Point](http://docs.google.com/java/awt/Point.html) p)

Returns the Accessible child contained at the local coordinate Point, if one exists. Otherwise returns null.

**Returns:**the Accessible at the specified location, if it exists; otherwise null

### getAccessibleStateSet

public static [AccessibleStateSet](http://docs.google.com/javax/accessibility/AccessibleStateSet.html) **getAccessibleStateSet**([Component](http://docs.google.com/java/awt/Component.html) c)

Get the state of this object.

Note: as of the Java 2 platform v1.3, it is recommended that developers call Component.AccessibleAWTComponent.getAccessibleIndexInParent() instead of using this method.

**Returns:**an instance of AccessibleStateSet containing the current state set of the object**See Also:**[AccessibleState](http://docs.google.com/javax/accessibility/AccessibleState.html)

### getAccessibleChildrenCount

public static int **getAccessibleChildrenCount**([Component](http://docs.google.com/java/awt/Component.html) c)

Returns the number of accessible children in the object. If all of the children of this object implement Accessible, than this method should return the number of children of this object.

Note: as of the Java 2 platform v1.3, it is recommended that developers call Component.AccessibleAWTComponent.getAccessibleIndexInParent() instead of using this method.

**Returns:**the number of accessible children in the object.

### getAccessibleChild

public static [Accessible](http://docs.google.com/javax/accessibility/Accessible.html) **getAccessibleChild**([Component](http://docs.google.com/java/awt/Component.html) c,  
 int i)

Return the nth Accessible child of the object.

Note: as of the Java 2 platform v1.3, it is recommended that developers call Component.AccessibleAWTComponent.getAccessibleIndexInParent() instead of using this method.

**Parameters:**i - zero-based index of child **Returns:**the nth Accessible child of the object

### findFocusOwner

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public static [Component](http://docs.google.com/java/awt/Component.html) **findFocusOwner**([Component](http://docs.google.com/java/awt/Component.html) c)

**Deprecated.** *As of 1.4, replaced by KeyboardFocusManager.getFocusOwner().*

Return the child Component of the specified Component that is the focus owner, if any.

**Parameters:**c - the root of the Component hierarchy to search for the focus owner **Returns:**the focus owner, or null if there is no focus owner, or if the focus owner is not comp, or a descendant of comp**See Also:**[KeyboardFocusManager.getFocusOwner()](http://docs.google.com/java/awt/KeyboardFocusManager.html#getFocusOwner())

### getRootPane

public static [JRootPane](http://docs.google.com/javax/swing/JRootPane.html) **getRootPane**([Component](http://docs.google.com/java/awt/Component.html) c)

If c is a JRootPane descendant return its JRootPane ancestor. If c is a RootPaneContainer then return its JRootPane.

**Returns:**the JRootPane for Component c or null.

### getRoot

public static [Component](http://docs.google.com/java/awt/Component.html) **getRoot**([Component](http://docs.google.com/java/awt/Component.html) c)

Returns the root component for the current component tree.

**Returns:**the first ancestor of c that's a Window or the last Applet ancestor

### processKeyBindings

public static boolean **processKeyBindings**([KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) event)

Process the key bindings for the Component associated with event. This method is only useful if event.getComponent() does not descend from JComponent, or your are not invoking super.processKeyEvent from within your JComponent subclass. JComponent automatically processes bindings from within its processKeyEvent method, hence you rarely need to directly invoke this method.

**Parameters:**event - KeyEvent used to identify which bindings to process, as well as which Component has focus. **Returns:**true if a binding has found and processed**Since:** 1.4

### notifyAction

public static boolean **notifyAction**([Action](http://docs.google.com/javax/swing/Action.html) action,  
 [KeyStroke](http://docs.google.com/javax/swing/KeyStroke.html) ks,  
 [KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) event,  
 [Object](http://docs.google.com/java/lang/Object.html) sender,  
 int modifiers)

Invokes actionPerformed on action if action is enabled (and non-null). The command for the ActionEvent is determined by:

1. If the action was registered via registerKeyboardAction, then the command string passed in (null will be used if null was passed in).
2. Action value with name Action.ACTION\_COMMAND\_KEY, unless null.
3. String value of the KeyEvent, unless getKeyChar returns KeyEvent.CHAR\_UNDEFINED..

This will return true if action is non-null and actionPerformed is invoked on it.

**Since:** 1.3

### replaceUIInputMap

public static void **replaceUIInputMap**([JComponent](http://docs.google.com/javax/swing/JComponent.html) component,  
 int type,  
 [InputMap](http://docs.google.com/javax/swing/InputMap.html) uiInputMap)

Convenience method to change the UI InputMap for component to uiInputMap. If uiInputMap is null, this removes any previously installed UI InputMap.

**Since:** 1.3

### replaceUIActionMap

public static void **replaceUIActionMap**([JComponent](http://docs.google.com/javax/swing/JComponent.html) component,  
 [ActionMap](http://docs.google.com/javax/swing/ActionMap.html) uiActionMap)

Convenience method to change the UI ActionMap for component to uiActionMap. If uiActionMap is null, this removes any previously installed UI ActionMap.

**Since:** 1.3

### getUIInputMap

public static [InputMap](http://docs.google.com/javax/swing/InputMap.html) **getUIInputMap**([JComponent](http://docs.google.com/javax/swing/JComponent.html) component,  
 int condition)

Returns the InputMap provided by the UI for condition condition in component component.

This will return null if the UI has not installed a InputMap of the specified type.

**Since:** 1.3

### getUIActionMap

public static [ActionMap](http://docs.google.com/javax/swing/ActionMap.html) **getUIActionMap**([JComponent](http://docs.google.com/javax/swing/JComponent.html) component)

Returns the ActionMap provided by the UI in component component.

This will return null if the UI has not installed an ActionMap.

**Since:** 1.3

### calculateInnerArea

public static [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **calculateInnerArea**([JComponent](http://docs.google.com/javax/swing/JComponent.html) c,  
 [Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)

Stores the position and size of the inner painting area of the specified component in r and returns r. The position and size specify the bounds of the component, adjusted so as not to include the border area (the insets). This method is useful for classes that implement painting code.

**Parameters:**c - the JComponent in question; if null, this method returns nullr - the Rectangle instance to be modified; may be null **Returns:**null if the Component is null; otherwise, returns the passed-in rectangle (if non-null) or a new rectangle specifying position and size information**Since:** 1.4

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/SwingUtilities.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/javax/swing/SwingConstants.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/SwingWorker.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/SwingUtilities.html)    [**NO FRAMES**](http://docs.google.com/SwingUtilities.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | CONSTR | [METHOD](#tyjcwt) | DETAIL: FIELD | CONSTR | [METHOD](#1t3h5sf) |

[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).