| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Component.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/Color.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Component.AccessibleAWTComponent.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Component.html)    [**NO FRAMES**](http://docs.google.com/Component.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#1ksv4uv) | [METHOD](#2jxsxqh) |

## **java.awt**

Class Component

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

**All Implemented Interfaces:** [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [MenuContainer](http://docs.google.com/java/awt/MenuContainer.html), [Serializable](http://docs.google.com/java/io/Serializable.html) **Direct Known Subclasses:** [Button](http://docs.google.com/java/awt/Button.html), [Canvas](http://docs.google.com/java/awt/Canvas.html), [Checkbox](http://docs.google.com/java/awt/Checkbox.html), [Choice](http://docs.google.com/java/awt/Choice.html), [Container](http://docs.google.com/java/awt/Container.html), [Label](http://docs.google.com/java/awt/Label.html), [List](http://docs.google.com/java/awt/List.html), [Scrollbar](http://docs.google.com/java/awt/Scrollbar.html), [TextComponent](http://docs.google.com/java/awt/TextComponent.html)

public abstract class **Component**extends [Object](http://docs.google.com/java/lang/Object.html)implements [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [MenuContainer](http://docs.google.com/java/awt/MenuContainer.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

A *component* is an object having a graphical representation that can be displayed on the screen and that can interact with the user. Examples of components are the buttons, checkboxes, and scrollbars of a typical graphical user interface.

The Component class is the abstract superclass of the nonmenu-related Abstract Window Toolkit components. Class Component can also be extended directly to create a lightweight component. A lightweight component is a component that is not associated with a native opaque window.

### Serialization

It is important to note that only AWT listeners which conform to the Serializable protocol will be saved when the object is stored. If an AWT object has listeners that aren't marked serializable, they will be dropped at writeObject time. Developers will need, as always, to consider the implications of making an object serializable. One situation to watch out for is this:

import java.awt.\*;  
 import java.awt.event.\*;  
 import java.io.Serializable;  
   
 class MyApp implements ActionListener, Serializable  
 {  
 BigObjectThatShouldNotBeSerializedWithAButton bigOne;  
 Button aButton = new Button();  
   
 MyApp()  
 {  
 // Oops, now aButton has a listener with a reference  
 // to bigOne!  
 aButton.addActionListener(this);  
 }  
   
 public void actionPerformed(ActionEvent e)  
 {  
 System.out.println("Hello There");  
 }  
 }

In this example, serializing aButton by itself will cause MyApp and everything it refers to to be serialized as well. The problem is that the listener is serializable by coincidence, not by design. To separate the decisions about MyApp and the ActionListener being serializable one can use a nested class, as in the following example:

import java.awt.\*;  
 import java.awt.event.\*;  
 import java.io.Serializable;  
  
 class MyApp java.io.Serializable  
 {  
 BigObjectThatShouldNotBeSerializedWithAButton bigOne;  
 Button aButton = new Button();  
  
 static class MyActionListener implements ActionListener  
 {  
 public void actionPerformed(ActionEvent e)  
 {  
 System.out.println("Hello There");  
 }  
 }  
   
 MyApp()  
 {  
 aButton.addActionListener(new MyActionListener());  
 }  
 }

**Note**: For more information on the paint mechanisms utilitized by AWT and Swing, including information on how to write the most efficient painting code, see [Painting in AWT and Swing](http://java.sun.com/products/jfc/tsc/articles/painting/index.html).

For details on the focus subsystem, see  [How to Use the Focus Subsystem](http://java.sun.com/docs/books/tutorial/uiswing/misc/focus.html), a section in *The Java Tutorial*, and the [Focus Specification](http://docs.google.com/java/awt/doc-files/FocusSpec.html) for more information.

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

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**Component.AccessibleAWTComponent**](http://docs.google.com/java/awt/Component.AccessibleAWTComponent.html)            Inner class of Component used to provide default support for accessibility. |
| static class | [**Component.BaselineResizeBehavior**](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html)            Enumeration of the common ways the baseline of a component can change as the size changes. |
| protected  class | [**Component.BltBufferStrategy**](http://docs.google.com/java/awt/Component.BltBufferStrategy.html)            Inner class for blitting offscreen surfaces to a component. |
| protected  class | [**Component.FlipBufferStrategy**](http://docs.google.com/java/awt/Component.FlipBufferStrategy.html)            Inner class for flipping buffers on a component. |

| **Field Summary** | |
| --- | --- |
| static float | [**BOTTOM\_ALIGNMENT**](http://docs.google.com/java/awt/Component.html#BOTTOM_ALIGNMENT)            Ease-of-use constant for getAlignmentY. |
| static float | [**CENTER\_ALIGNMENT**](http://docs.google.com/java/awt/Component.html#CENTER_ALIGNMENT)            Ease-of-use constant for getAlignmentY and getAlignmentX. |
| static float | [**LEFT\_ALIGNMENT**](http://docs.google.com/java/awt/Component.html#LEFT_ALIGNMENT)            Ease-of-use constant for getAlignmentX. |
| static float | [**RIGHT\_ALIGNMENT**](http://docs.google.com/java/awt/Component.html#RIGHT_ALIGNMENT)            Ease-of-use constant for getAlignmentX. |
| static float | [**TOP\_ALIGNMENT**](http://docs.google.com/java/awt/Component.html#TOP_ALIGNMENT)            Ease-of-use constant for getAlignmentY(). |

| **Fields inherited from interface java.awt.image.**[**ImageObserver**](http://docs.google.com/java/awt/image/ImageObserver.html) |
| --- |
| [ABORT](http://docs.google.com/java/awt/image/ImageObserver.html#ABORT), [ALLBITS](http://docs.google.com/java/awt/image/ImageObserver.html#ALLBITS), [ERROR](http://docs.google.com/java/awt/image/ImageObserver.html#ERROR), [FRAMEBITS](http://docs.google.com/java/awt/image/ImageObserver.html#FRAMEBITS), [HEIGHT](http://docs.google.com/java/awt/image/ImageObserver.html#HEIGHT), [PROPERTIES](http://docs.google.com/java/awt/image/ImageObserver.html#PROPERTIES), [SOMEBITS](http://docs.google.com/java/awt/image/ImageObserver.html#SOMEBITS), [WIDTH](http://docs.google.com/java/awt/image/ImageObserver.html#WIDTH) |

| **Constructor Summary** | |
| --- | --- |
| protected | [**Component**](http://docs.google.com/java/awt/Component.html#Component())()            Constructs a new component. |

| **Method Summary** | |
| --- | --- |
| boolean | [**action**](http://docs.google.com/java/awt/Component.html#action(java.awt.Event,%20java.lang.Object))([Event](http://docs.google.com/java/awt/Event.html) evt, [Object](http://docs.google.com/java/lang/Object.html) what)  **Deprecated.** *As of JDK version 1.1, should register this component as ActionListener on component which fires action events.* |
| void | [**add**](http://docs.google.com/java/awt/Component.html#add(java.awt.PopupMenu))([PopupMenu](http://docs.google.com/java/awt/PopupMenu.html) popup)            Adds the specified popup menu to the component. |
| void | [**addComponentListener**](http://docs.google.com/java/awt/Component.html#addComponentListener(java.awt.event.ComponentListener))([ComponentListener](http://docs.google.com/java/awt/event/ComponentListener.html) l)            Adds the specified component listener to receive component events from this component. |
| void | [**addFocusListener**](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener))([FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) l)            Adds the specified focus listener to receive focus events from this component when this component gains input focus. |
| void | [**addHierarchyBoundsListener**](http://docs.google.com/java/awt/Component.html#addHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener))([HierarchyBoundsListener](http://docs.google.com/java/awt/event/HierarchyBoundsListener.html) l)            Adds the specified hierarchy bounds listener to receive hierarchy bounds events from this component when the hierarchy to which this container belongs changes. |
| void | [**addHierarchyListener**](http://docs.google.com/java/awt/Component.html#addHierarchyListener(java.awt.event.HierarchyListener))([HierarchyListener](http://docs.google.com/java/awt/event/HierarchyListener.html) l)            Adds the specified hierarchy listener to receive hierarchy changed events from this component when the hierarchy to which this container belongs changes. |
| void | [**addInputMethodListener**](http://docs.google.com/java/awt/Component.html#addInputMethodListener(java.awt.event.InputMethodListener))([InputMethodListener](http://docs.google.com/java/awt/event/InputMethodListener.html) l)            Adds the specified input method listener to receive input method events from this component. |
| void | [**addKeyListener**](http://docs.google.com/java/awt/Component.html#addKeyListener(java.awt.event.KeyListener))([KeyListener](http://docs.google.com/java/awt/event/KeyListener.html) l)            Adds the specified key listener to receive key events from this component. |
| void | [**addMouseListener**](http://docs.google.com/java/awt/Component.html#addMouseListener(java.awt.event.MouseListener))([MouseListener](http://docs.google.com/java/awt/event/MouseListener.html) l)            Adds the specified mouse listener to receive mouse events from this component. |
| void | [**addMouseMotionListener**](http://docs.google.com/java/awt/Component.html#addMouseMotionListener(java.awt.event.MouseMotionListener))([MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html) l)            Adds the specified mouse motion listener to receive mouse motion events from this component. |
| void | [**addMouseWheelListener**](http://docs.google.com/java/awt/Component.html#addMouseWheelListener(java.awt.event.MouseWheelListener))([MouseWheelListener](http://docs.google.com/java/awt/event/MouseWheelListener.html) l)            Adds the specified mouse wheel listener to receive mouse wheel events from this component. |
| void | [**addNotify**](http://docs.google.com/java/awt/Component.html#addNotify())()            Makes this Component displayable by connecting it to a native screen resource. |
| void | [**addPropertyChangeListener**](http://docs.google.com/java/awt/Component.html#addPropertyChangeListener(java.beans.PropertyChangeListener))([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)            Adds a PropertyChangeListener to the listener list. |
| void | [**addPropertyChangeListener**](http://docs.google.com/java/awt/Component.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener))([String](http://docs.google.com/java/lang/String.html) propertyName, [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)            Adds a PropertyChangeListener to the listener list for a specific property. |
| void | [**applyComponentOrientation**](http://docs.google.com/java/awt/Component.html#applyComponentOrientation(java.awt.ComponentOrientation))([ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html) orientation)            Sets the ComponentOrientation property of this component and all components contained within it. |
| boolean | [**areFocusTraversalKeysSet**](http://docs.google.com/java/awt/Component.html#areFocusTraversalKeysSet(int))(int id)            Returns whether the Set of focus traversal keys for the given focus traversal operation has been explicitly defined for this Component. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**bounds**](http://docs.google.com/java/awt/Component.html#bounds())()  **Deprecated.** *As of JDK version 1.1, replaced by getBounds().* |
| int | [**checkImage**](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20java.awt.image.ImageObserver))([Image](http://docs.google.com/java/awt/Image.html) image, [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)            Returns the status of the construction of a screen representation of the specified image. |
| int | [**checkImage**](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver))([Image](http://docs.google.com/java/awt/Image.html) image, int width, int height, [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)            Returns the status of the construction of a screen representation of the specified image. |
| protected  [AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) | [**coalesceEvents**](http://docs.google.com/java/awt/Component.html#coalesceEvents(java.awt.AWTEvent,%20java.awt.AWTEvent))([AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) existingEvent, [AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) newEvent)            Potentially coalesce an event being posted with an existing event. |
| boolean | [**contains**](http://docs.google.com/java/awt/Component.html#contains(int,%20int))(int x, int y)            Checks whether this component "contains" the specified point, where x and y are defined to be relative to the coordinate system of this component. |
| boolean | [**contains**](http://docs.google.com/java/awt/Component.html#contains(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) p)            Checks whether this component "contains" the specified point, where the point's *x* and *y* coordinates are defined to be relative to the coordinate system of this component. |
| [Image](http://docs.google.com/java/awt/Image.html) | [**createImage**](http://docs.google.com/java/awt/Component.html#createImage(java.awt.image.ImageProducer))([ImageProducer](http://docs.google.com/java/awt/image/ImageProducer.html) producer)            Creates an image from the specified image producer. |
| [Image](http://docs.google.com/java/awt/Image.html) | [**createImage**](http://docs.google.com/java/awt/Component.html#createImage(int,%20int))(int width, int height)            Creates an off-screen drawable image to be used for double buffering. |
| [VolatileImage](http://docs.google.com/java/awt/image/VolatileImage.html) | [**createVolatileImage**](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int))(int width, int height)            Creates a volatile off-screen drawable image to be used for double buffering. |
| [VolatileImage](http://docs.google.com/java/awt/image/VolatileImage.html) | [**createVolatileImage**](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int,%20java.awt.ImageCapabilities))(int width, int height, [ImageCapabilities](http://docs.google.com/java/awt/ImageCapabilities.html) caps)            Creates a volatile off-screen drawable image, with the given capabilities. |
| void | [**deliverEvent**](http://docs.google.com/java/awt/Component.html#deliverEvent(java.awt.Event))([Event](http://docs.google.com/java/awt/Event.html) e)  **Deprecated.** *As of JDK version 1.1, replaced by dispatchEvent(AWTEvent e).* |
| void | [**disable**](http://docs.google.com/java/awt/Component.html#disable())()  **Deprecated.** *As of JDK version 1.1, replaced by setEnabled(boolean).* |
| protected  void | [**disableEvents**](http://docs.google.com/java/awt/Component.html#disableEvents(long))(long eventsToDisable)            Disables the events defined by the specified event mask parameter from being delivered to this component. |
| void | [**dispatchEvent**](http://docs.google.com/java/awt/Component.html#dispatchEvent(java.awt.AWTEvent))([AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) e)            Dispatches an event to this component or one of its sub components. |
| void | [**doLayout**](http://docs.google.com/java/awt/Component.html#doLayout())()            Prompts the layout manager to lay out this component. |
| void | [**enable**](http://docs.google.com/java/awt/Component.html#enable())()  **Deprecated.** *As of JDK version 1.1, replaced by setEnabled(boolean).* |
| void | [**enable**](http://docs.google.com/java/awt/Component.html#enable(boolean))(boolean b)  **Deprecated.** *As of JDK version 1.1, replaced by setEnabled(boolean).* |
| protected  void | [**enableEvents**](http://docs.google.com/java/awt/Component.html#enableEvents(long))(long eventsToEnable)            Enables the events defined by the specified event mask parameter to be delivered to this component. |
| void | [**enableInputMethods**](http://docs.google.com/java/awt/Component.html#enableInputMethods(boolean))(boolean enable)            Enables or disables input method support for this component. |
| protected  void | [**firePropertyChange**](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20boolean,%20boolean))([String](http://docs.google.com/java/lang/String.html) propertyName, boolean oldValue, boolean newValue)            Support for reporting bound property changes for boolean properties. |
| void | [**firePropertyChange**](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20byte,%20byte))([String](http://docs.google.com/java/lang/String.html) propertyName, byte oldValue, byte newValue)            Reports a bound property change. |
| void | [**firePropertyChange**](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20char,%20char))([String](http://docs.google.com/java/lang/String.html) propertyName, char oldValue, char newValue)            Reports a bound property change. |
| void | [**firePropertyChange**](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20double,%20double))([String](http://docs.google.com/java/lang/String.html) propertyName, double oldValue, double newValue)            Reports a bound property change. |
| void | [**firePropertyChange**](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20float,%20float))([String](http://docs.google.com/java/lang/String.html) propertyName, float oldValue, float newValue)            Reports a bound property change. |
| protected  void | [**firePropertyChange**](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20int,%20int))([String](http://docs.google.com/java/lang/String.html) propertyName, int oldValue, int newValue)            Support for reporting bound property changes for integer properties. |
| void | [**firePropertyChange**](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20long,%20long))([String](http://docs.google.com/java/lang/String.html) propertyName, long oldValue, long newValue)            Reports a bound property change. |
| protected  void | [**firePropertyChange**](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object))([String](http://docs.google.com/java/lang/String.html) propertyName, [Object](http://docs.google.com/java/lang/Object.html) oldValue, [Object](http://docs.google.com/java/lang/Object.html) newValue)            Support for reporting bound property changes for Object properties. |
| void | [**firePropertyChange**](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20short,%20short))([String](http://docs.google.com/java/lang/String.html) propertyName, short oldValue, short newValue)            Reports a bound property change. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/java/awt/Component.html#getAccessibleContext())()            Gets the AccessibleContext associated with this Component. |
| float | [**getAlignmentX**](http://docs.google.com/java/awt/Component.html#getAlignmentX())()            Returns the alignment along the x axis. |
| float | [**getAlignmentY**](http://docs.google.com/java/awt/Component.html#getAlignmentY())()            Returns the alignment along the y axis. |
| [Color](http://docs.google.com/java/awt/Color.html) | [**getBackground**](http://docs.google.com/java/awt/Component.html#getBackground())()            Gets the background color of this component. |
| int | [**getBaseline**](http://docs.google.com/java/awt/Component.html#getBaseline(int,%20int))(int width, int height)            Returns the baseline. |
| [Component.BaselineResizeBehavior](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html) | [**getBaselineResizeBehavior**](http://docs.google.com/java/awt/Component.html#getBaselineResizeBehavior())()            Returns an enum indicating how the baseline of the component changes as the size changes. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getBounds**](http://docs.google.com/java/awt/Component.html#getBounds())()            Gets the bounds of this component in the form of a Rectangle object. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getBounds**](http://docs.google.com/java/awt/Component.html#getBounds(java.awt.Rectangle))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) rv)            Stores the bounds of this component into "return value" **rv** and return **rv**. |
| [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) | [**getColorModel**](http://docs.google.com/java/awt/Component.html#getColorModel())()            Gets the instance of ColorModel used to display the component on the output device. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getComponentAt**](http://docs.google.com/java/awt/Component.html#getComponentAt(int,%20int))(int x, int y)            Determines if this component or one of its immediate subcomponents contains the (*x*, *y*) location, and if so, returns the containing component. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getComponentAt**](http://docs.google.com/java/awt/Component.html#getComponentAt(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) p)            Returns the component or subcomponent that contains the specified point. |
| [ComponentListener](http://docs.google.com/java/awt/event/ComponentListener.html)[] | [**getComponentListeners**](http://docs.google.com/java/awt/Component.html#getComponentListeners())()            Returns an array of all the component listeners registered on this component. |
| [ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html) | [**getComponentOrientation**](http://docs.google.com/java/awt/Component.html#getComponentOrientation())()            Retrieves the language-sensitive orientation that is to be used to order the elements or text within this component. |
| [Cursor](http://docs.google.com/java/awt/Cursor.html) | [**getCursor**](http://docs.google.com/java/awt/Component.html#getCursor())()            Gets the cursor set in the component. |
| [DropTarget](http://docs.google.com/java/awt/dnd/DropTarget.html) | [**getDropTarget**](http://docs.google.com/java/awt/Component.html#getDropTarget())()            Gets the DropTarget associated with this Component. |
| [Container](http://docs.google.com/java/awt/Container.html) | [**getFocusCycleRootAncestor**](http://docs.google.com/java/awt/Component.html#getFocusCycleRootAncestor())()            Returns the Container which is the focus cycle root of this Component's focus traversal cycle. |
| [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html)[] | [**getFocusListeners**](http://docs.google.com/java/awt/Component.html#getFocusListeners())()            Returns an array of all the focus listeners registered on this component. |
| [Set](http://docs.google.com/java/util/Set.html)<[AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html)> | [**getFocusTraversalKeys**](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeys(int))(int id)            Returns the Set of focus traversal keys for a given traversal operation for this Component. |
| boolean | [**getFocusTraversalKeysEnabled**](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeysEnabled())()            Returns whether focus traversal keys are enabled for this Component. |
| [Font](http://docs.google.com/java/awt/Font.html) | [**getFont**](http://docs.google.com/java/awt/Component.html#getFont())()            Gets the font of this component. |
| [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) | [**getFontMetrics**](http://docs.google.com/java/awt/Component.html#getFontMetrics(java.awt.Font))([Font](http://docs.google.com/java/awt/Font.html) font)            Gets the font metrics for the specified font. |
| [Color](http://docs.google.com/java/awt/Color.html) | [**getForeground**](http://docs.google.com/java/awt/Component.html#getForeground())()            Gets the foreground color of this component. |
| [Graphics](http://docs.google.com/java/awt/Graphics.html) | [**getGraphics**](http://docs.google.com/java/awt/Component.html#getGraphics())()            Creates a graphics context for this component. |
| [GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html) | [**getGraphicsConfiguration**](http://docs.google.com/java/awt/Component.html#getGraphicsConfiguration())()            Gets the GraphicsConfiguration associated with this Component. |
| int | [**getHeight**](http://docs.google.com/java/awt/Component.html#getHeight())()            Returns the current height of this component. |
| [HierarchyBoundsListener](http://docs.google.com/java/awt/event/HierarchyBoundsListener.html)[] | [**getHierarchyBoundsListeners**](http://docs.google.com/java/awt/Component.html#getHierarchyBoundsListeners())()            Returns an array of all the hierarchy bounds listeners registered on this component. |
| [HierarchyListener](http://docs.google.com/java/awt/event/HierarchyListener.html)[] | [**getHierarchyListeners**](http://docs.google.com/java/awt/Component.html#getHierarchyListeners())()            Returns an array of all the hierarchy listeners registered on this component. |
| boolean | [**getIgnoreRepaint**](http://docs.google.com/java/awt/Component.html#getIgnoreRepaint())() |
| [InputContext](http://docs.google.com/java/awt/im/InputContext.html) | [**getInputContext**](http://docs.google.com/java/awt/Component.html#getInputContext())()            Gets the input context used by this component for handling the communication with input methods when text is entered in this component. |
| [InputMethodListener](http://docs.google.com/java/awt/event/InputMethodListener.html)[] | [**getInputMethodListeners**](http://docs.google.com/java/awt/Component.html#getInputMethodListeners())()            Returns an array of all the input method listeners registered on this component. |
| [InputMethodRequests](http://docs.google.com/java/awt/im/InputMethodRequests.html) | [**getInputMethodRequests**](http://docs.google.com/java/awt/Component.html#getInputMethodRequests())()            Gets the input method request handler which supports requests from input methods for this component. |
| [KeyListener](http://docs.google.com/java/awt/event/KeyListener.html)[] | [**getKeyListeners**](http://docs.google.com/java/awt/Component.html#getKeyListeners())()            Returns an array of all the key listeners registered on this component. |
| | <T extends [EventListener](http://docs.google.com/java/util/EventListener.html)>  T[] | | --- | | [**getListeners**](http://docs.google.com/java/awt/Component.html#getListeners(java.lang.Class))([Class](http://docs.google.com/java/lang/Class.html)<T> listenerType)            Returns an array of all the objects currently registered as *Foo*Listeners upon this Component. |
| [Locale](http://docs.google.com/java/util/Locale.html) | [**getLocale**](http://docs.google.com/java/awt/Component.html#getLocale())()            Gets the locale of this component. |
| [Point](http://docs.google.com/java/awt/Point.html) | [**getLocation**](http://docs.google.com/java/awt/Component.html#getLocation())()            Gets the location of this component in the form of a point specifying the component's top-left corner. |
| [Point](http://docs.google.com/java/awt/Point.html) | [**getLocation**](http://docs.google.com/java/awt/Component.html#getLocation(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) rv)            Stores the x,y origin of this component into "return value" **rv** and return **rv**. |
| [Point](http://docs.google.com/java/awt/Point.html) | [**getLocationOnScreen**](http://docs.google.com/java/awt/Component.html#getLocationOnScreen())()            Gets the location of this component in the form of a point specifying the component's top-left corner in the screen's coordinate space. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMaximumSize**](http://docs.google.com/java/awt/Component.html#getMaximumSize())()            Gets the maximum size of this component. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getMinimumSize**](http://docs.google.com/java/awt/Component.html#getMinimumSize())()            Gets the mininimum size of this component. |
| [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html)[] | [**getMouseListeners**](http://docs.google.com/java/awt/Component.html#getMouseListeners())()            Returns an array of all the mouse listeners registered on this component. |
| [MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html)[] | [**getMouseMotionListeners**](http://docs.google.com/java/awt/Component.html#getMouseMotionListeners())()            Returns an array of all the mouse motion listeners registered on this component. |
| [Point](http://docs.google.com/java/awt/Point.html) | [**getMousePosition**](http://docs.google.com/java/awt/Component.html#getMousePosition())()            Returns the position of the mouse pointer in this Component's coordinate space if the Component is directly under the mouse pointer, otherwise returns null. |
| [MouseWheelListener](http://docs.google.com/java/awt/event/MouseWheelListener.html)[] | [**getMouseWheelListeners**](http://docs.google.com/java/awt/Component.html#getMouseWheelListeners())()            Returns an array of all the mouse wheel listeners registered on this component. |
| [String](http://docs.google.com/java/lang/String.html) | [**getName**](http://docs.google.com/java/awt/Component.html#getName())()            Gets the name of the component. |
| [Container](http://docs.google.com/java/awt/Container.html) | [**getParent**](http://docs.google.com/java/awt/Component.html#getParent())()            Gets the parent of this component. |
| java.awt.peer.ComponentPeer | [**getPeer**](http://docs.google.com/java/awt/Component.html#getPeer())()  **Deprecated.** *As of JDK version 1.1, programs should not directly manipulate peers; replaced by boolean isDisplayable().* |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getPreferredSize**](http://docs.google.com/java/awt/Component.html#getPreferredSize())()            Gets the preferred size of this component. |
| [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html)[] | [**getPropertyChangeListeners**](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners())()            Returns an array of all the property change listeners registered on this component. |
| [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html)[] | [**getPropertyChangeListeners**](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners(java.lang.String))([String](http://docs.google.com/java/lang/String.html) propertyName)            Returns an array of all the listeners which have been associated with the named property. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getSize**](http://docs.google.com/java/awt/Component.html#getSize())()            Returns the size of this component in the form of a Dimension object. |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**getSize**](http://docs.google.com/java/awt/Component.html#getSize(java.awt.Dimension))([Dimension](http://docs.google.com/java/awt/Dimension.html) rv)            Stores the width/height of this component into "return value" **rv** and return **rv**. |
| [Toolkit](http://docs.google.com/java/awt/Toolkit.html) | [**getToolkit**](http://docs.google.com/java/awt/Component.html#getToolkit())()            Gets the toolkit of this component. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getTreeLock**](http://docs.google.com/java/awt/Component.html#getTreeLock())()            Gets this component's locking object (the object that owns the thread sychronization monitor) for AWT component-tree and layout operations. |
| int | [**getWidth**](http://docs.google.com/java/awt/Component.html#getWidth())()            Returns the current width of this component. |
| int | [**getX**](http://docs.google.com/java/awt/Component.html#getX())()            Returns the current x coordinate of the components origin. |
| int | [**getY**](http://docs.google.com/java/awt/Component.html#getY())()            Returns the current y coordinate of the components origin. |
| boolean | [**gotFocus**](http://docs.google.com/java/awt/Component.html#gotFocus(java.awt.Event,%20java.lang.Object))([Event](http://docs.google.com/java/awt/Event.html) evt, [Object](http://docs.google.com/java/lang/Object.html) what)  **Deprecated.** *As of JDK version 1.1, replaced by processFocusEvent(FocusEvent).* |
| boolean | [**handleEvent**](http://docs.google.com/java/awt/Component.html#handleEvent(java.awt.Event))([Event](http://docs.google.com/java/awt/Event.html) evt)  **Deprecated.** *As of JDK version 1.1 replaced by processEvent(AWTEvent).* |
| boolean | [**hasFocus**](http://docs.google.com/java/awt/Component.html#hasFocus())()            Returns true if this Component is the focus owner. |
| void | [**hide**](http://docs.google.com/java/awt/Component.html#hide())()  **Deprecated.** *As of JDK version 1.1, replaced by setVisible(boolean).* |
| boolean | [**imageUpdate**](http://docs.google.com/java/awt/Component.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int))([Image](http://docs.google.com/java/awt/Image.html) img, int infoflags, int x, int y, int w, int h)            Repaints the component when the image has changed. |
| boolean | [**inside**](http://docs.google.com/java/awt/Component.html#inside(int,%20int))(int x, int y)  **Deprecated.** *As of JDK version 1.1, replaced by contains(int, int).* |
| void | [**invalidate**](http://docs.google.com/java/awt/Component.html#invalidate())()            Invalidates this component. |
| boolean | [**isBackgroundSet**](http://docs.google.com/java/awt/Component.html#isBackgroundSet())()            Returns whether the background color has been explicitly set for this Component. |
| boolean | [**isCursorSet**](http://docs.google.com/java/awt/Component.html#isCursorSet())()            Returns whether the cursor has been explicitly set for this Component. |
| boolean | [**isDisplayable**](http://docs.google.com/java/awt/Component.html#isDisplayable())()            Determines whether this component is displayable. |
| boolean | [**isDoubleBuffered**](http://docs.google.com/java/awt/Component.html#isDoubleBuffered())()            Returns true if this component is painted to an offscreen image ("buffer") that's copied to the screen later. |
| boolean | [**isEnabled**](http://docs.google.com/java/awt/Component.html#isEnabled())()            Determines whether this component is enabled. |
| boolean | [**isFocusable**](http://docs.google.com/java/awt/Component.html#isFocusable())()            Returns whether this Component can be focused. |
| boolean | [**isFocusCycleRoot**](http://docs.google.com/java/awt/Component.html#isFocusCycleRoot(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) container)            Returns whether the specified Container is the focus cycle root of this Component's focus traversal cycle. |
| boolean | [**isFocusOwner**](http://docs.google.com/java/awt/Component.html#isFocusOwner())()            Returns true if this Component is the focus owner. |
| boolean | [**isFocusTraversable**](http://docs.google.com/java/awt/Component.html#isFocusTraversable())()  **Deprecated.** *As of 1.4, replaced by isFocusable().* |
| boolean | [**isFontSet**](http://docs.google.com/java/awt/Component.html#isFontSet())()            Returns whether the font has been explicitly set for this Component. |
| boolean | [**isForegroundSet**](http://docs.google.com/java/awt/Component.html#isForegroundSet())()            Returns whether the foreground color has been explicitly set for this Component. |
| boolean | [**isLightweight**](http://docs.google.com/java/awt/Component.html#isLightweight())()            A lightweight component doesn't have a native toolkit peer. |
| boolean | [**isMaximumSizeSet**](http://docs.google.com/java/awt/Component.html#isMaximumSizeSet())()            Returns true if the maximum size has been set to a non-null value otherwise returns false. |
| boolean | [**isMinimumSizeSet**](http://docs.google.com/java/awt/Component.html#isMinimumSizeSet())()            Returns whether or not setMinimumSize has been invoked with a non-null value. |
| boolean | [**isOpaque**](http://docs.google.com/java/awt/Component.html#isOpaque())()            Returns true if this component is completely opaque, returns false by default. |
| boolean | [**isPreferredSizeSet**](http://docs.google.com/java/awt/Component.html#isPreferredSizeSet())()            Returns true if the preferred size has been set to a non-null value otherwise returns false. |
| boolean | [**isShowing**](http://docs.google.com/java/awt/Component.html#isShowing())()            Determines whether this component is showing on screen. |
| boolean | [**isValid**](http://docs.google.com/java/awt/Component.html#isValid())()            Determines whether this component is valid. |
| boolean | [**isVisible**](http://docs.google.com/java/awt/Component.html#isVisible())()            Determines whether this component should be visible when its parent is visible. |
| boolean | [**keyDown**](http://docs.google.com/java/awt/Component.html#keyDown(java.awt.Event,%20int))([Event](http://docs.google.com/java/awt/Event.html) evt, int key)  **Deprecated.** *As of JDK version 1.1, replaced by processKeyEvent(KeyEvent).* |
| boolean | [**keyUp**](http://docs.google.com/java/awt/Component.html#keyUp(java.awt.Event,%20int))([Event](http://docs.google.com/java/awt/Event.html) evt, int key)  **Deprecated.** *As of JDK version 1.1, replaced by processKeyEvent(KeyEvent).* |
| void | [**layout**](http://docs.google.com/java/awt/Component.html#layout())()  **Deprecated.** *As of JDK version 1.1, replaced by doLayout().* |
| void | [**list**](http://docs.google.com/java/awt/Component.html#list())()            Prints a listing of this component to the standard system output stream System.out. |
| void | [**list**](http://docs.google.com/java/awt/Component.html#list(java.io.PrintStream))([PrintStream](http://docs.google.com/java/io/PrintStream.html) out)            Prints a listing of this component to the specified output stream. |
| void | [**list**](http://docs.google.com/java/awt/Component.html#list(java.io.PrintStream,%20int))([PrintStream](http://docs.google.com/java/io/PrintStream.html) out, int indent)            Prints out a list, starting at the specified indentation, to the specified print stream. |
| void | [**list**](http://docs.google.com/java/awt/Component.html#list(java.io.PrintWriter))([PrintWriter](http://docs.google.com/java/io/PrintWriter.html) out)            Prints a listing to the specified print writer. |
| void | [**list**](http://docs.google.com/java/awt/Component.html#list(java.io.PrintWriter,%20int))([PrintWriter](http://docs.google.com/java/io/PrintWriter.html) out, int indent)            Prints out a list, starting at the specified indentation, to the specified print writer. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**locate**](http://docs.google.com/java/awt/Component.html#locate(int,%20int))(int x, int y)  **Deprecated.** *As of JDK version 1.1, replaced by getComponentAt(int, int).* |
| [Point](http://docs.google.com/java/awt/Point.html) | [**location**](http://docs.google.com/java/awt/Component.html#location())()  **Deprecated.** *As of JDK version 1.1, replaced by getLocation().* |
| boolean | [**lostFocus**](http://docs.google.com/java/awt/Component.html#lostFocus(java.awt.Event,%20java.lang.Object))([Event](http://docs.google.com/java/awt/Event.html) evt, [Object](http://docs.google.com/java/lang/Object.html) what)  **Deprecated.** *As of JDK version 1.1, replaced by processFocusEvent(FocusEvent).* |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**minimumSize**](http://docs.google.com/java/awt/Component.html#minimumSize())()  **Deprecated.** *As of JDK version 1.1, replaced by getMinimumSize().* |
| boolean | [**mouseDown**](http://docs.google.com/java/awt/Component.html#mouseDown(java.awt.Event,%20int,%20int))([Event](http://docs.google.com/java/awt/Event.html) evt, int x, int y)  **Deprecated.** *As of JDK version 1.1, replaced by processMouseEvent(MouseEvent).* |
| boolean | [**mouseDrag**](http://docs.google.com/java/awt/Component.html#mouseDrag(java.awt.Event,%20int,%20int))([Event](http://docs.google.com/java/awt/Event.html) evt, int x, int y)  **Deprecated.** *As of JDK version 1.1, replaced by processMouseMotionEvent(MouseEvent).* |
| boolean | [**mouseEnter**](http://docs.google.com/java/awt/Component.html#mouseEnter(java.awt.Event,%20int,%20int))([Event](http://docs.google.com/java/awt/Event.html) evt, int x, int y)  **Deprecated.** *As of JDK version 1.1, replaced by processMouseEvent(MouseEvent).* |
| boolean | [**mouseExit**](http://docs.google.com/java/awt/Component.html#mouseExit(java.awt.Event,%20int,%20int))([Event](http://docs.google.com/java/awt/Event.html) evt, int x, int y)  **Deprecated.** *As of JDK version 1.1, replaced by processMouseEvent(MouseEvent).* |
| boolean | [**mouseMove**](http://docs.google.com/java/awt/Component.html#mouseMove(java.awt.Event,%20int,%20int))([Event](http://docs.google.com/java/awt/Event.html) evt, int x, int y)  **Deprecated.** *As of JDK version 1.1, replaced by processMouseMotionEvent(MouseEvent).* |
| boolean | [**mouseUp**](http://docs.google.com/java/awt/Component.html#mouseUp(java.awt.Event,%20int,%20int))([Event](http://docs.google.com/java/awt/Event.html) evt, int x, int y)  **Deprecated.** *As of JDK version 1.1, replaced by processMouseEvent(MouseEvent).* |
| void | [**move**](http://docs.google.com/java/awt/Component.html#move(int,%20int))(int x, int y)  **Deprecated.** *As of JDK version 1.1, replaced by setLocation(int, int).* |
| void | [**nextFocus**](http://docs.google.com/java/awt/Component.html#nextFocus())()  **Deprecated.** *As of JDK version 1.1, replaced by transferFocus().* |
| void | [**paint**](http://docs.google.com/java/awt/Component.html#paint(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Paints this component. |
| void | [**paintAll**](http://docs.google.com/java/awt/Component.html#paintAll(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Paints this component and all of its subcomponents. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/java/awt/Component.html#paramString())()            Returns a string representing the state of this component. |
| boolean | [**postEvent**](http://docs.google.com/java/awt/Component.html#postEvent(java.awt.Event))([Event](http://docs.google.com/java/awt/Event.html) e)  **Deprecated.** *As of JDK version 1.1, replaced by dispatchEvent(AWTEvent).* |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**preferredSize**](http://docs.google.com/java/awt/Component.html#preferredSize())()  **Deprecated.** *As of JDK version 1.1, replaced by getPreferredSize().* |
| boolean | [**prepareImage**](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20java.awt.image.ImageObserver))([Image](http://docs.google.com/java/awt/Image.html) image, [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)            Prepares an image for rendering on this component. |
| boolean | [**prepareImage**](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver))([Image](http://docs.google.com/java/awt/Image.html) image, int width, int height, [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)            Prepares an image for rendering on this component at the specified width and height. |
| void | [**print**](http://docs.google.com/java/awt/Component.html#print(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Prints this component. |
| void | [**printAll**](http://docs.google.com/java/awt/Component.html#printAll(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Prints this component and all of its subcomponents. |
| protected  void | [**processComponentEvent**](http://docs.google.com/java/awt/Component.html#processComponentEvent(java.awt.event.ComponentEvent))([ComponentEvent](http://docs.google.com/java/awt/event/ComponentEvent.html) e)            Processes component events occurring on this component by dispatching them to any registered ComponentListener objects. |
| protected  void | [**processEvent**](http://docs.google.com/java/awt/Component.html#processEvent(java.awt.AWTEvent))([AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) e)            Processes events occurring on this component. |
| protected  void | [**processFocusEvent**](http://docs.google.com/java/awt/Component.html#processFocusEvent(java.awt.event.FocusEvent))([FocusEvent](http://docs.google.com/java/awt/event/FocusEvent.html) e)            Processes focus events occurring on this component by dispatching them to any registered FocusListener objects. |
| protected  void | [**processHierarchyBoundsEvent**](http://docs.google.com/java/awt/Component.html#processHierarchyBoundsEvent(java.awt.event.HierarchyEvent))([HierarchyEvent](http://docs.google.com/java/awt/event/HierarchyEvent.html) e)            Processes hierarchy bounds events occurring on this component by dispatching them to any registered HierarchyBoundsListener objects. |
| protected  void | [**processHierarchyEvent**](http://docs.google.com/java/awt/Component.html#processHierarchyEvent(java.awt.event.HierarchyEvent))([HierarchyEvent](http://docs.google.com/java/awt/event/HierarchyEvent.html) e)            Processes hierarchy events occurring on this component by dispatching them to any registered HierarchyListener objects. |
| protected  void | [**processInputMethodEvent**](http://docs.google.com/java/awt/Component.html#processInputMethodEvent(java.awt.event.InputMethodEvent))([InputMethodEvent](http://docs.google.com/java/awt/event/InputMethodEvent.html) e)            Processes input method events occurring on this component by dispatching them to any registered InputMethodListener objects. |
| protected  void | [**processKeyEvent**](http://docs.google.com/java/awt/Component.html#processKeyEvent(java.awt.event.KeyEvent))([KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) e)            Processes key events occurring on this component by dispatching them to any registered KeyListener objects. |
| protected  void | [**processMouseEvent**](http://docs.google.com/java/awt/Component.html#processMouseEvent(java.awt.event.MouseEvent))([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) e)            Processes mouse events occurring on this component by dispatching them to any registered MouseListener objects. |
| protected  void | [**processMouseMotionEvent**](http://docs.google.com/java/awt/Component.html#processMouseMotionEvent(java.awt.event.MouseEvent))([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) e)            Processes mouse motion events occurring on this component by dispatching them to any registered MouseMotionListener objects. |
| protected  void | [**processMouseWheelEvent**](http://docs.google.com/java/awt/Component.html#processMouseWheelEvent(java.awt.event.MouseWheelEvent))([MouseWheelEvent](http://docs.google.com/java/awt/event/MouseWheelEvent.html) e)            Processes mouse wheel events occurring on this component by dispatching them to any registered MouseWheelListener objects. |
| void | [**remove**](http://docs.google.com/java/awt/Component.html#remove(java.awt.MenuComponent))([MenuComponent](http://docs.google.com/java/awt/MenuComponent.html) popup)            Removes the specified popup menu from the component. |
| void | [**removeComponentListener**](http://docs.google.com/java/awt/Component.html#removeComponentListener(java.awt.event.ComponentListener))([ComponentListener](http://docs.google.com/java/awt/event/ComponentListener.html) l)            Removes the specified component listener so that it no longer receives component events from this component. |
| void | [**removeFocusListener**](http://docs.google.com/java/awt/Component.html#removeFocusListener(java.awt.event.FocusListener))([FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) l)            Removes the specified focus listener so that it no longer receives focus events from this component. |
| void | [**removeHierarchyBoundsListener**](http://docs.google.com/java/awt/Component.html#removeHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener))([HierarchyBoundsListener](http://docs.google.com/java/awt/event/HierarchyBoundsListener.html) l)            Removes the specified hierarchy bounds listener so that it no longer receives hierarchy bounds events from this component. |
| void | [**removeHierarchyListener**](http://docs.google.com/java/awt/Component.html#removeHierarchyListener(java.awt.event.HierarchyListener))([HierarchyListener](http://docs.google.com/java/awt/event/HierarchyListener.html) l)            Removes the specified hierarchy listener so that it no longer receives hierarchy changed events from this component. |
| void | [**removeInputMethodListener**](http://docs.google.com/java/awt/Component.html#removeInputMethodListener(java.awt.event.InputMethodListener))([InputMethodListener](http://docs.google.com/java/awt/event/InputMethodListener.html) l)            Removes the specified input method listener so that it no longer receives input method events from this component. |
| void | [**removeKeyListener**](http://docs.google.com/java/awt/Component.html#removeKeyListener(java.awt.event.KeyListener))([KeyListener](http://docs.google.com/java/awt/event/KeyListener.html) l)            Removes the specified key listener so that it no longer receives key events from this component. |
| void | [**removeMouseListener**](http://docs.google.com/java/awt/Component.html#removeMouseListener(java.awt.event.MouseListener))([MouseListener](http://docs.google.com/java/awt/event/MouseListener.html) l)            Removes the specified mouse listener so that it no longer receives mouse events from this component. |
| void | [**removeMouseMotionListener**](http://docs.google.com/java/awt/Component.html#removeMouseMotionListener(java.awt.event.MouseMotionListener))([MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html) l)            Removes the specified mouse motion listener so that it no longer receives mouse motion events from this component. |
| void | [**removeMouseWheelListener**](http://docs.google.com/java/awt/Component.html#removeMouseWheelListener(java.awt.event.MouseWheelListener))([MouseWheelListener](http://docs.google.com/java/awt/event/MouseWheelListener.html) l)            Removes the specified mouse wheel listener so that it no longer receives mouse wheel events from this component. |
| void | [**removeNotify**](http://docs.google.com/java/awt/Component.html#removeNotify())()            Makes this Component undisplayable by destroying it native screen resource. |
| void | [**removePropertyChangeListener**](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.beans.PropertyChangeListener))([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)            Removes a PropertyChangeListener from the listener list. |
| void | [**removePropertyChangeListener**](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener))([String](http://docs.google.com/java/lang/String.html) propertyName, [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)            Removes a PropertyChangeListener from the listener list for a specific property. |
| void | [**repaint**](http://docs.google.com/java/awt/Component.html#repaint())()            Repaints this component. |
| void | [**repaint**](http://docs.google.com/java/awt/Component.html#repaint(int,%20int,%20int,%20int))(int x, int y, int width, int height)            Repaints the specified rectangle of this component. |
| void | [**repaint**](http://docs.google.com/java/awt/Component.html#repaint(long))(long tm)            Repaints the component. |
| void | [**repaint**](http://docs.google.com/java/awt/Component.html#repaint(long,%20int,%20int,%20int,%20int))(long tm, int x, int y, int width, int height)            Repaints the specified rectangle of this component within tm milliseconds. |
| void | [**requestFocus**](http://docs.google.com/java/awt/Component.html#requestFocus())()            Requests that this Component get the input focus, and that this Component's top-level ancestor become the focused Window. |
| protected  boolean | [**requestFocus**](http://docs.google.com/java/awt/Component.html#requestFocus(boolean))(boolean temporary)            Requests that this Component get the input focus, and that this Component's top-level ancestor become the focused Window. |
| boolean | [**requestFocusInWindow**](http://docs.google.com/java/awt/Component.html#requestFocusInWindow())()            Requests that this Component get the input focus, if this Component's top-level ancestor is already the focused Window. |
| protected  boolean | [**requestFocusInWindow**](http://docs.google.com/java/awt/Component.html#requestFocusInWindow(boolean))(boolean temporary)            Requests that this Component get the input focus, if this Component's top-level ancestor is already the focused Window. |
| void | [**reshape**](http://docs.google.com/java/awt/Component.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/Component.html#resize(java.awt.Dimension))([Dimension](http://docs.google.com/java/awt/Dimension.html) d)  **Deprecated.** *As of JDK version 1.1, replaced by setSize(Dimension).* |
| void | [**resize**](http://docs.google.com/java/awt/Component.html#resize(int,%20int))(int width, int height)  **Deprecated.** *As of JDK version 1.1, replaced by setSize(int, int).* |
| void | [**setBackground**](http://docs.google.com/java/awt/Component.html#setBackground(java.awt.Color))([Color](http://docs.google.com/java/awt/Color.html) c)            Sets the background color of this component. |
| void | [**setBounds**](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%20int))(int x, int y, int width, int height)            Moves and resizes this component. |
| void | [**setBounds**](http://docs.google.com/java/awt/Component.html#setBounds(java.awt.Rectangle))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)            Moves and resizes this component to conform to the new bounding rectangle r. |
| void | [**setComponentOrientation**](http://docs.google.com/java/awt/Component.html#setComponentOrientation(java.awt.ComponentOrientation))([ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html) o)            Sets the language-sensitive orientation that is to be used to order the elements or text within this component. |
| void | [**setCursor**](http://docs.google.com/java/awt/Component.html#setCursor(java.awt.Cursor))([Cursor](http://docs.google.com/java/awt/Cursor.html) cursor)            Sets the cursor image to the specified cursor. |
| void | [**setDropTarget**](http://docs.google.com/java/awt/Component.html#setDropTarget(java.awt.dnd.DropTarget))([DropTarget](http://docs.google.com/java/awt/dnd/DropTarget.html) dt)            Associate a DropTarget with this component. |
| void | [**setEnabled**](http://docs.google.com/java/awt/Component.html#setEnabled(boolean))(boolean b)            Enables or disables this component, depending on the value of the parameter b. |
| void | [**setFocusable**](http://docs.google.com/java/awt/Component.html#setFocusable(boolean))(boolean focusable)            Sets the focusable state of this Component to the specified value. |
| void | [**setFocusTraversalKeys**](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeys(int,%20java.util.Set))(int id, [Set](http://docs.google.com/java/util/Set.html)<? extends [AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html)> keystrokes)            Sets the focus traversal keys for a given traversal operation for this Component. |
| void | [**setFocusTraversalKeysEnabled**](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeysEnabled(boolean))(boolean focusTraversalKeysEnabled)            Sets whether focus traversal keys are enabled for this Component. |
| void | [**setFont**](http://docs.google.com/java/awt/Component.html#setFont(java.awt.Font))([Font](http://docs.google.com/java/awt/Font.html) f)            Sets the font of this component. |
| void | [**setForeground**](http://docs.google.com/java/awt/Component.html#setForeground(java.awt.Color))([Color](http://docs.google.com/java/awt/Color.html) c)            Sets the foreground color of this component. |
| void | [**setIgnoreRepaint**](http://docs.google.com/java/awt/Component.html#setIgnoreRepaint(boolean))(boolean ignoreRepaint)            Sets whether or not paint messages received from the operating system should be ignored. |
| void | [**setLocale**](http://docs.google.com/java/awt/Component.html#setLocale(java.util.Locale))([Locale](http://docs.google.com/java/util/Locale.html) l)            Sets the locale of this component. |
| void | [**setLocation**](http://docs.google.com/java/awt/Component.html#setLocation(int,%20int))(int x, int y)            Moves this component to a new location. |
| void | [**setLocation**](http://docs.google.com/java/awt/Component.html#setLocation(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) p)            Moves this component to a new location. |
| void | [**setMaximumSize**](http://docs.google.com/java/awt/Component.html#setMaximumSize(java.awt.Dimension))([Dimension](http://docs.google.com/java/awt/Dimension.html) maximumSize)            Sets the maximum size of this component to a constant value. |
| void | [**setMinimumSize**](http://docs.google.com/java/awt/Component.html#setMinimumSize(java.awt.Dimension))([Dimension](http://docs.google.com/java/awt/Dimension.html) minimumSize)            Sets the minimum size of this component to a constant value. |
| void | [**setName**](http://docs.google.com/java/awt/Component.html#setName(java.lang.String))([String](http://docs.google.com/java/lang/String.html) name)            Sets the name of the component to the specified string. |
| void | [**setPreferredSize**](http://docs.google.com/java/awt/Component.html#setPreferredSize(java.awt.Dimension))([Dimension](http://docs.google.com/java/awt/Dimension.html) preferredSize)            Sets the preferred size of this component to a constant value. |
| void | [**setSize**](http://docs.google.com/java/awt/Component.html#setSize(java.awt.Dimension))([Dimension](http://docs.google.com/java/awt/Dimension.html) d)            Resizes this component so that it has width d.width and height d.height. |
| void | [**setSize**](http://docs.google.com/java/awt/Component.html#setSize(int,%20int))(int width, int height)            Resizes this component so that it has width width and height height. |
| void | [**setVisible**](http://docs.google.com/java/awt/Component.html#setVisible(boolean))(boolean b)            Shows or hides this component depending on the value of parameter b. |
| void | [**show**](http://docs.google.com/java/awt/Component.html#show())()  **Deprecated.** *As of JDK version 1.1, replaced by setVisible(boolean).* |
| void | [**show**](http://docs.google.com/java/awt/Component.html#show(boolean))(boolean b)  **Deprecated.** *As of JDK version 1.1, replaced by setVisible(boolean).* |
| [Dimension](http://docs.google.com/java/awt/Dimension.html) | [**size**](http://docs.google.com/java/awt/Component.html#size())()  **Deprecated.** *As of JDK version 1.1, replaced by getSize().* |
| [String](http://docs.google.com/java/lang/String.html) | [**toString**](http://docs.google.com/java/awt/Component.html#toString())()            Returns a string representation of this component and its values. |
| void | [**transferFocus**](http://docs.google.com/java/awt/Component.html#transferFocus())()            Transfers the focus to the next component, as though this Component were the focus owner. |
| void | [**transferFocusBackward**](http://docs.google.com/java/awt/Component.html#transferFocusBackward())()            Transfers the focus to the previous component, as though this Component were the focus owner. |
| void | [**transferFocusUpCycle**](http://docs.google.com/java/awt/Component.html#transferFocusUpCycle())()            Transfers the focus up one focus traversal cycle. |
| void | [**update**](http://docs.google.com/java/awt/Component.html#update(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Updates this component. |
| void | [**validate**](http://docs.google.com/java/awt/Component.html#validate())()            Ensures that this component has a valid layout. |

| **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()), [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)) |

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

### TOP\_ALIGNMENT

public static final float **TOP\_ALIGNMENT**

Ease-of-use constant for getAlignmentY(). Specifies an alignment to the top of the component.

**See Also:**[getAlignmentY()](http://docs.google.com/java/awt/Component.html#getAlignmentY()), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Component.TOP_ALIGNMENT)

### CENTER\_ALIGNMENT

public static final float **CENTER\_ALIGNMENT**

Ease-of-use constant for getAlignmentY and getAlignmentX. Specifies an alignment to the center of the component

**See Also:**[getAlignmentX()](http://docs.google.com/java/awt/Component.html#getAlignmentX()), [getAlignmentY()](http://docs.google.com/java/awt/Component.html#getAlignmentY()), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Component.CENTER_ALIGNMENT)

### BOTTOM\_ALIGNMENT

public static final float **BOTTOM\_ALIGNMENT**

Ease-of-use constant for getAlignmentY. Specifies an alignment to the bottom of the component.

**See Also:**[getAlignmentY()](http://docs.google.com/java/awt/Component.html#getAlignmentY()), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Component.BOTTOM_ALIGNMENT)

### LEFT\_ALIGNMENT

public static final float **LEFT\_ALIGNMENT**

Ease-of-use constant for getAlignmentX. Specifies an alignment to the left side of the component.

**See Also:**[getAlignmentX()](http://docs.google.com/java/awt/Component.html#getAlignmentX()), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Component.LEFT_ALIGNMENT)

### RIGHT\_ALIGNMENT

public static final float **RIGHT\_ALIGNMENT**

Ease-of-use constant for getAlignmentX. Specifies an alignment to the right side of the component.

**See Also:**[getAlignmentX()](http://docs.google.com/java/awt/Component.html#getAlignmentX()), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Component.RIGHT_ALIGNMENT)

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

### Component

protected **Component**()

Constructs a new component. Class Component can be extended directly to create a lightweight component that does not utilize an opaque native window. A lightweight component must be hosted by a native container somewhere higher up in the component tree (for example, by a Frame object).

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

### getName

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

Gets the name of the component.

**Returns:**this component's name**Since:** JDK1.1 **See Also:**[setName(java.lang.String)](http://docs.google.com/java/awt/Component.html#setName(java.lang.String))

### setName

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

Sets the name of the component to the specified string.

**Parameters:**name - the string that is to be this component's name**Since:** JDK1.1 **See Also:**[getName()](http://docs.google.com/java/awt/Component.html#getName())

### getParent

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

Gets the parent of this component.

**Returns:**the parent container of this component**Since:** JDK1.0

### getPeer

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public java.awt.peer.ComponentPeer **getPeer**()

**Deprecated.** *As of JDK version 1.1, programs should not directly manipulate peers; replaced by boolean isDisplayable().*

### setDropTarget

public void **setDropTarget**([DropTarget](http://docs.google.com/java/awt/dnd/DropTarget.html) dt)

Associate a DropTarget with this component. The Component will receive drops only if it is enabled.

**Parameters:**dt - The DropTarget**See Also:**[isEnabled()](http://docs.google.com/java/awt/Component.html#isEnabled())

### getDropTarget

public [DropTarget](http://docs.google.com/java/awt/dnd/DropTarget.html) **getDropTarget**()

Gets the DropTarget associated with this Component.

### getGraphicsConfiguration

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

Gets the GraphicsConfiguration associated with this Component. If the Component has not been assigned a specific GraphicsConfiguration, the GraphicsConfiguration of the Component object's top-level container is returned. If the Component has been created, but not yet added to a Container, this method returns null.

**Returns:**the GraphicsConfiguration used by this Component or null**Since:** 1.3

### getTreeLock

public final [Object](http://docs.google.com/java/lang/Object.html) **getTreeLock**()

Gets this component's locking object (the object that owns the thread sychronization monitor) for AWT component-tree and layout operations.

**Returns:**this component's locking object

### getToolkit

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

Gets the toolkit of this component. Note that the frame that contains a component controls which toolkit is used by that component. Therefore if the component is moved from one frame to another, the toolkit it uses may change.

**Returns:**the toolkit of this component**Since:** JDK1.0

### isValid

public boolean **isValid**()

Determines whether this component is valid. A component is valid when it is correctly sized and positioned within its parent container and all its children are also valid. In order to account for peers' size requirements, components are invalidated before they are first shown on the screen. By the time the parent container is fully realized, all its components will be valid.

**Returns:**true if the component is valid, false otherwise**Since:** JDK1.0 **See Also:**[validate()](http://docs.google.com/java/awt/Component.html#validate()), [invalidate()](http://docs.google.com/java/awt/Component.html#invalidate())

### isDisplayable

public boolean **isDisplayable**()

Determines whether this component is displayable. A component is displayable when it is connected to a native screen resource.

A component is made displayable either when it is added to a displayable containment hierarchy or when its containment hierarchy is made displayable. A containment hierarchy is made displayable when its ancestor window is either packed or made visible.

A component is made undisplayable either when it is removed from a displayable containment hierarchy or when its containment hierarchy is made undisplayable. A containment hierarchy is made undisplayable when its ancestor window is disposed.

**Returns:**true if the component is displayable, false otherwise**Since:** 1.2 **See Also:**[Container.add(Component)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [Window.pack()](http://docs.google.com/java/awt/Window.html#pack()), [Window.show()](http://docs.google.com/java/awt/Window.html#show()), [Container.remove(Component)](http://docs.google.com/java/awt/Container.html#remove(java.awt.Component)), [Window.dispose()](http://docs.google.com/java/awt/Window.html#dispose())

### isVisible

public boolean **isVisible**()

Determines whether this component should be visible when its parent is visible. Components are initially visible, with the exception of top level components such as Frame objects.

**Returns:**true if the component is visible, false otherwise**Since:** JDK1.0 **See Also:**[setVisible(boolean)](http://docs.google.com/java/awt/Component.html#setVisible(boolean))

### getMousePosition

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

Returns the position of the mouse pointer in this Component's coordinate space if the Component is directly under the mouse pointer, otherwise returns null. If the Component is not showing on the screen, this method returns null even if the mouse pointer is above the area where the Component would be displayed. If the Component is partially or fully obscured by other Components or native windows, this method returns a non-null value only if the mouse pointer is located above the unobscured part of the Component.

For Containers it returns a non-null value if the mouse is above the Container itself or above any of its descendants. Use [Container.getMousePosition(boolean)](http://docs.google.com/java/awt/Container.html#getMousePosition(boolean)) if you need to exclude children.

Sometimes the exact mouse coordinates are not important, and the only thing that matters is whether a specific Component is under the mouse pointer. If the return value of this method is null, mouse pointer is not directly above the Component.

**Returns:**mouse coordinates relative to this Component, or null **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true**Since:** 1.5 **See Also:**[isShowing()](http://docs.google.com/java/awt/Component.html#isShowing()), [Container.getMousePosition(boolean)](http://docs.google.com/java/awt/Container.html#getMousePosition(boolean))

### isShowing

public boolean **isShowing**()

Determines whether this component is showing on screen. This means that the component must be visible, and it must be in a container that is visible and showing.

**Note:** sometimes there is no way to detect whether the Component is actually visible to the user. This can happen when:

* the component has been added to a visible ScrollPane but the Component is not currently in the scroll pane's view port.
* the Component is obscured by another Component or Container.

**Returns:**true if the component is showing, false otherwise**Since:** JDK1.0 **See Also:**[setVisible(boolean)](http://docs.google.com/java/awt/Component.html#setVisible(boolean))

### isEnabled

public boolean **isEnabled**()

Determines whether this component is enabled. An enabled component can respond to user input and generate events. Components are enabled initially by default. A component may be enabled or disabled by calling its setEnabled method.

**Returns:**true if the component is enabled, false otherwise**Since:** JDK1.0 **See Also:**[setEnabled(boolean)](http://docs.google.com/java/awt/Component.html#setEnabled(boolean))

### setEnabled

public void **setEnabled**(boolean b)

Enables or disables this component, depending on the value of the parameter b. An enabled component can respond to user input and generate events. Components are enabled initially by default.

Note: Disabling a lightweight component does not prevent it from receiving MouseEvents.

Note: Disabling a heavyweight container prevents all components in this container from receiving any input events. But disabling a lightweight container affects only this container.

**Parameters:**b - If true, this component is enabled; otherwise this component is disabled**Since:** JDK1.1 **See Also:**[isEnabled()](http://docs.google.com/java/awt/Component.html#isEnabled()), [isLightweight()](http://docs.google.com/java/awt/Component.html#isLightweight())

### enable

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **enable**()

**Deprecated.** *As of JDK version 1.1, replaced by setEnabled(boolean).*

### enable

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **enable**(boolean b)

**Deprecated.** *As of JDK version 1.1, replaced by setEnabled(boolean).*

### disable

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **disable**()

**Deprecated.** *As of JDK version 1.1, replaced by setEnabled(boolean).*

### isDoubleBuffered

public boolean **isDoubleBuffered**()

Returns true if this component is painted to an offscreen image ("buffer") that's copied to the screen later. Component subclasses that support double buffering should override this method to return true if double buffering is enabled.

**Returns:**false by default

### enableInputMethods

public void **enableInputMethods**(boolean enable)

Enables or disables input method support for this component. If input method support is enabled and the component also processes key events, incoming events are offered to the current input method and will only be processed by the component or dispatched to its listeners if the input method does not consume them. By default, input method support is enabled.

**Parameters:**enable - true to enable, false to disable**Since:** 1.2 **See Also:**[processKeyEvent(java.awt.event.KeyEvent)](http://docs.google.com/java/awt/Component.html#processKeyEvent(java.awt.event.KeyEvent))

### setVisible

public void **setVisible**(boolean b)

Shows or hides this component depending on the value of parameter b.

**Parameters:**b - if true, shows this component; otherwise, hides this component**Since:** JDK1.1 **See Also:**[isVisible()](http://docs.google.com/java/awt/Component.html#isVisible())

### show

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **show**()

**Deprecated.** *As of JDK version 1.1, replaced by setVisible(boolean).*

### show

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **show**(boolean b)

**Deprecated.** *As of JDK version 1.1, replaced by setVisible(boolean).*

### hide

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **hide**()

**Deprecated.** *As of JDK version 1.1, replaced by setVisible(boolean).*

### getForeground

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

Gets the foreground color of this component.

**Returns:**this component's foreground color; if this component does not have a foreground color, the foreground color of its parent is returned**Since:** JDK1.0 **See Also:**[setForeground(java.awt.Color)](http://docs.google.com/java/awt/Component.html#setForeground(java.awt.Color))

### setForeground

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

Sets the foreground color of this component.

**Parameters:**c - the color to become this component's foreground color; if this parameter is null then this component will inherit the foreground color of its parent**Since:** JDK1.0 **See Also:**[getForeground()](http://docs.google.com/java/awt/Component.html#getForeground())

### isForegroundSet

public boolean **isForegroundSet**()

Returns whether the foreground color has been explicitly set for this Component. If this method returns false, this Component is inheriting its foreground color from an ancestor.

**Returns:**true if the foreground color has been explicitly set for this Component; false otherwise.**Since:** 1.4

### getBackground

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

Gets the background color of this component.

**Returns:**this component's background color; if this component does not have a background color, the background color of its parent is returned**Since:** JDK1.0 **See Also:**[setBackground(java.awt.Color)](http://docs.google.com/java/awt/Component.html#setBackground(java.awt.Color))

### setBackground

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

Sets the background color of this component.

The background color affects each component differently and the parts of the component that are affected by the background color may differ between operating systems.

**Parameters:**c - the color to become this component's color; if this parameter is null, then this component will inherit the background color of its parent**Since:** JDK1.0 **See Also:**[getBackground()](http://docs.google.com/java/awt/Component.html#getBackground())

### isBackgroundSet

public boolean **isBackgroundSet**()

Returns whether the background color has been explicitly set for this Component. If this method returns false, this Component is inheriting its background color from an ancestor.

**Returns:**true if the background color has been explicitly set for this Component; false otherwise.**Since:** 1.4

### getFont

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

Gets the font of this component.

**Specified by:**[getFont](http://docs.google.com/java/awt/MenuContainer.html#getFont()) in interface [MenuContainer](http://docs.google.com/java/awt/MenuContainer.html) **Returns:**this component's font; if a font has not been set for this component, the font of its parent is returned**Since:** JDK1.0 **See Also:**[setFont(java.awt.Font)](http://docs.google.com/java/awt/Component.html#setFont(java.awt.Font))

### setFont

public void **setFont**([Font](http://docs.google.com/java/awt/Font.html) f)

Sets the font of this component.

**Parameters:**f - the font to become this component's font; if this parameter is null then this component will inherit the font of its parent**Since:** JDK1.0 **See Also:**[getFont()](http://docs.google.com/java/awt/Component.html#getFont())

### isFontSet

public boolean **isFontSet**()

Returns whether the font has been explicitly set for this Component. If this method returns false, this Component is inheriting its font from an ancestor.

**Returns:**true if the font has been explicitly set for this Component; false otherwise.**Since:** 1.4

### getLocale

public [Locale](http://docs.google.com/java/util/Locale.html) **getLocale**()

Gets the locale of this component.

**Returns:**this component's locale; if this component does not have a locale, the locale of its parent is returned **Throws:** [IllegalComponentStateException](http://docs.google.com/java/awt/IllegalComponentStateException.html) - if the Component does not have its own locale and has not yet been added to a containment hierarchy such that the locale can be determined from the containing parent**Since:** JDK1.1 **See Also:**[setLocale(java.util.Locale)](http://docs.google.com/java/awt/Component.html#setLocale(java.util.Locale))

### setLocale

public void **setLocale**([Locale](http://docs.google.com/java/util/Locale.html) l)

Sets the locale of this component. This is a bound property.

**Parameters:**l - the locale to become this component's locale**Since:** JDK1.1 **See Also:**[getLocale()](http://docs.google.com/java/awt/Component.html#getLocale())

### getColorModel

public [ColorModel](http://docs.google.com/java/awt/image/ColorModel.html) **getColorModel**()

Gets the instance of ColorModel used to display the component on the output device.

**Returns:**the color model used by this component**Since:** JDK1.0 **See Also:**[ColorModel](http://docs.google.com/java/awt/image/ColorModel.html), ComponentPeer.getColorModel(), [Toolkit.getColorModel()](http://docs.google.com/java/awt/Toolkit.html#getColorModel())

### getLocation

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

Gets the location of this component in the form of a point specifying the component's top-left corner. The location will be relative to the parent's coordinate space.

Due to the asynchronous nature of native event handling, this method can return outdated values (for instance, after several calls of setLocation() in rapid succession). For this reason, the recommended method of obtaining a component's position is within java.awt.event.ComponentListener.componentMoved(), which is called after the operating system has finished moving the component.

**Returns:**an instance of Point representing the top-left corner of the component's bounds in the coordinate space of the component's parent**Since:** JDK1.1 **See Also:**[setLocation(int, int)](http://docs.google.com/java/awt/Component.html#setLocation(int,%20int)), [getLocationOnScreen()](http://docs.google.com/java/awt/Component.html#getLocationOnScreen())

### getLocationOnScreen

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

Gets the location of this component in the form of a point specifying the component's top-left corner in the screen's coordinate space.

**Returns:**an instance of Point representing the top-left corner of the component's bounds in the coordinate space of the screen **Throws:** IllegalComponentStateException - if the component is not showing on the screen**See Also:**[setLocation(int, int)](http://docs.google.com/java/awt/Component.html#setLocation(int,%20int)), [getLocation()](http://docs.google.com/java/awt/Component.html#getLocation())

### location

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [Point](http://docs.google.com/java/awt/Point.html) **location**()

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

### setLocation

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

Moves this component to a new location. The top-left corner of the new location is specified by the x and y parameters in the coordinate space of this component's parent.

**Parameters:**x - the *x*-coordinate of the new location's top-left corner in the parent's coordinate spacey - the *y*-coordinate of the new location's top-left corner in the parent's coordinate space**Since:** JDK1.1 **See Also:**[getLocation()](http://docs.google.com/java/awt/Component.html#getLocation()), [setBounds(int, int, int, int)](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%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).*

### setLocation

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

Moves this component to a new location. The top-left corner of the new location is specified by point p. Point p is given in the parent's coordinate space.

**Parameters:**p - the point defining the top-left corner of the new location, given in the coordinate space of this component's parent**Since:** JDK1.1 **See Also:**[getLocation()](http://docs.google.com/java/awt/Component.html#getLocation()), [setBounds(int, int, int, int)](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%20int))

### getSize

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

Returns the size of this component in the form of a Dimension object. The height field of the Dimension object contains this component's height, and the width field of the Dimension object contains this component's width.

**Returns:**a Dimension object that indicates the size of this component**Since:** JDK1.1 **See Also:**[setSize(int, int)](http://docs.google.com/java/awt/Component.html#setSize(int,%20int))

### size

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [Dimension](http://docs.google.com/java/awt/Dimension.html) **size**()

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

### setSize

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

Resizes this component so that it has width width and height height.

**Parameters:**width - the new width of this component in pixelsheight - the new height of this component in pixels**Since:** JDK1.1 **See Also:**[getSize()](http://docs.google.com/java/awt/Component.html#getSize()), [setBounds(int, int, int, int)](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%20int))

### 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).*

### setSize

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

Resizes this component so that it has width d.width and height d.height.

**Parameters:**d - the dimension specifying the new size of this component**Since:** JDK1.1 **See Also:**[setSize(int, int)](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [setBounds(int, int, int, int)](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%20int))

### resize

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **resize**([Dimension](http://docs.google.com/java/awt/Dimension.html) d)

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

### getBounds

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

Gets the bounds of this component in the form of a Rectangle object. The bounds specify this component's width, height, and location relative to its parent.

**Returns:**a rectangle indicating this component's bounds**See Also:**[setBounds(int, int, int, int)](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%20int)), [getLocation()](http://docs.google.com/java/awt/Component.html#getLocation()), [getSize()](http://docs.google.com/java/awt/Component.html#getSize())

### bounds

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [Rectangle](http://docs.google.com/java/awt/Rectangle.html) **bounds**()

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

### setBounds

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

Moves and resizes this component. The new location of the top-left corner is specified by x and y, and the new size is specified by width and height.

**Parameters:**x - the new *x*-coordinate of this componenty - the new *y*-coordinate of this componentwidth - the new width of this componentheight - the new height of this component**Since:** JDK1.1 **See Also:**[getBounds()](http://docs.google.com/java/awt/Component.html#getBounds()), [setLocation(int, int)](http://docs.google.com/java/awt/Component.html#setLocation(int,%20int)), [setLocation(Point)](http://docs.google.com/java/awt/Component.html#setLocation(java.awt.Point)), [setSize(int, int)](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [setSize(Dimension)](http://docs.google.com/java/awt/Component.html#setSize(java.awt.Dimension))

### 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).*

### setBounds

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

Moves and resizes this component to conform to the new bounding rectangle r. This component's new position is specified by r.x and r.y, and its new size is specified by r.width and r.height

**Parameters:**r - the new bounding rectangle for this component**Since:** JDK1.1 **See Also:**[getBounds()](http://docs.google.com/java/awt/Component.html#getBounds()), [setLocation(int, int)](http://docs.google.com/java/awt/Component.html#setLocation(int,%20int)), [setLocation(Point)](http://docs.google.com/java/awt/Component.html#setLocation(java.awt.Point)), [setSize(int, int)](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [setSize(Dimension)](http://docs.google.com/java/awt/Component.html#setSize(java.awt.Dimension))

### getX

public int **getX**()

Returns the current x coordinate of the components origin. This method is preferable to writing component.getBounds().x, or component.getLocation().x because it doesn't cause any heap allocations.

**Returns:**the current x coordinate of the components origin**Since:** 1.2

### getY

public int **getY**()

Returns the current y coordinate of the components origin. This method is preferable to writing component.getBounds().y, or component.getLocation().y because it doesn't cause any heap allocations.

**Returns:**the current y coordinate of the components origin**Since:** 1.2

### getWidth

public int **getWidth**()

Returns the current width of this component. This method is preferable to writing component.getBounds().width, or component.getSize().width because it doesn't cause any heap allocations.

**Returns:**the current width of this component**Since:** 1.2

### getHeight

public int **getHeight**()

Returns the current height of this component. This method is preferable to writing component.getBounds().height, or component.getSize().height because it doesn't cause any heap allocations.

**Returns:**the current height of this component**Since:** 1.2

### getBounds

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

Stores the bounds of this component into "return value" **rv** and return **rv**. If rv is null a new Rectangle is allocated. This version of getBounds is useful if the caller wants to avoid allocating a new Rectangle object on the heap.

**Parameters:**rv - the return value, modified to the components bounds **Returns:**rv

### getSize

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

Stores the width/height of this component into "return value" **rv** and return **rv**. If rv is null a new Dimension object is allocated. This version of getSize is useful if the caller wants to avoid allocating a new Dimension object on the heap.

**Parameters:**rv - the return value, modified to the components size **Returns:**rv

### getLocation

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

Stores the x,y origin of this component into "return value" **rv** and return **rv**. If rv is null a new Point is allocated. This version of getLocation is useful if the caller wants to avoid allocating a new Point object on the heap.

**Parameters:**rv - the return value, modified to the components location **Returns:**rv

### isOpaque

public boolean **isOpaque**()

Returns true if this component is completely opaque, returns false by default.

An opaque component paints every pixel within its rectangular region. A non-opaque component paints only some of its pixels, allowing the pixels underneath it to "show through". A component that does not fully paint its pixels therefore provides a degree of transparency. Only lightweight components can be transparent.

Subclasses that guarantee to always completely paint their contents should override this method and return true. All of the "heavyweight" AWT components are opaque.

**Returns:**true if this component is completely opaque**Since:** 1.2 **See Also:**[isLightweight()](http://docs.google.com/java/awt/Component.html#isLightweight())

### isLightweight

public boolean **isLightweight**()

A lightweight component doesn't have a native toolkit peer. Subclasses of Component and Container, other than the ones defined in this package like Button or Scrollbar, are lightweight. All of the Swing components are lightweights.

This method will always return false if this component is not displayable because it is impossible to determine the weight of an undisplayable component.

**Returns:**true if this component has a lightweight peer; false if it has a native peer or no peer**Since:** 1.2 **See Also:**[isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable())

### setPreferredSize

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

Sets the preferred size of this component to a constant value. Subsequent calls to getPreferredSize will always return this value. Setting the preferred size to null restores the default behavior.

**Parameters:**preferredSize - The new preferred size, or null**Since:** 1.5 **See Also:**[getPreferredSize()](http://docs.google.com/java/awt/Component.html#getPreferredSize()), [isPreferredSizeSet()](http://docs.google.com/java/awt/Component.html#isPreferredSizeSet())

### isPreferredSizeSet

public boolean **isPreferredSizeSet**()

Returns true if the preferred size has been set to a non-null value otherwise returns false.

**Returns:**true if setPreferredSize has been invoked with a non-null value.**Since:** 1.5

### getPreferredSize

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

Gets the preferred size of this component.

**Returns:**a dimension object indicating this component's preferred size**See Also:**[getMinimumSize()](http://docs.google.com/java/awt/Component.html#getMinimumSize()), [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html)

### preferredSize

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [Dimension](http://docs.google.com/java/awt/Dimension.html) **preferredSize**()

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

### setMinimumSize

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

Sets the minimum size of this component to a constant value. Subsequent calls to getMinimumSize will always return this value. Setting the minimum size to null restores the default behavior.

**Parameters:**minimumSize - the new minimum size of this component**Since:** 1.5 **See Also:**[getMinimumSize()](http://docs.google.com/java/awt/Component.html#getMinimumSize()), [isMinimumSizeSet()](http://docs.google.com/java/awt/Component.html#isMinimumSizeSet())

### isMinimumSizeSet

public boolean **isMinimumSizeSet**()

Returns whether or not setMinimumSize has been invoked with a non-null value.

**Returns:**true if setMinimumSize has been invoked with a non-null value.**Since:** 1.5

### getMinimumSize

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

Gets the mininimum size of this component.

**Returns:**a dimension object indicating this component's minimum size**See Also:**[getPreferredSize()](http://docs.google.com/java/awt/Component.html#getPreferredSize()), [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html)

### minimumSize

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [Dimension](http://docs.google.com/java/awt/Dimension.html) **minimumSize**()

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

### setMaximumSize

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

Sets the maximum size of this component to a constant value. Subsequent calls to getMaximumSize will always return this value. Setting the maximum size to null restores the default behavior.

**Parameters:**maximumSize - a Dimension containing the desired maximum allowable size**Since:** 1.5 **See Also:**[getMaximumSize()](http://docs.google.com/java/awt/Component.html#getMaximumSize()), [isMaximumSizeSet()](http://docs.google.com/java/awt/Component.html#isMaximumSizeSet())

### isMaximumSizeSet

public boolean **isMaximumSizeSet**()

Returns true if the maximum size has been set to a non-null value otherwise returns false.

**Returns:**true if maximumSize is non-null, false otherwise**Since:** 1.5

### getMaximumSize

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

Gets the maximum size of this component.

**Returns:**a dimension object indicating this component's maximum size**See Also:**[getMinimumSize()](http://docs.google.com/java/awt/Component.html#getMinimumSize()), [getPreferredSize()](http://docs.google.com/java/awt/Component.html#getPreferredSize()), [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html)

### getAlignmentX

public float **getAlignmentX**()

Returns the alignment along the x axis. This specifies how the component would like to be aligned relative to other components. The value should be a number between 0 and 1 where 0 represents alignment along the origin, 1 is aligned the furthest away from the origin, 0.5 is centered, etc.

### getAlignmentY

public float **getAlignmentY**()

Returns the alignment along the y axis. This specifies how the component would like to be aligned relative to other components. The value should be a number between 0 and 1 where 0 represents alignment along the origin, 1 is aligned the furthest away from the origin, 0.5 is centered, etc.

### getBaseline

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

Returns the baseline. The baseline is measured from the top of the component. This method is primarily meant for LayoutManagers to align components along their baseline. A return value less than 0 indicates this component does not have a reasonable baseline and that LayoutManagers should not align this component on its baseline.

The default implementation returns -1. Subclasses that support baseline should override appropriately. If a value >= 0 is returned, then the component has a valid baseline for any size >= the minimum size and getBaselineResizeBehavior can be used to determine how the baseline changes with size.

**Parameters:**width - the width to get the baseline forheight - the height to get the baseline for **Returns:**the baseline or < 0 indicating there is no reasonable baseline **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if width or height is < 0**Since:** 1.6 **See Also:**[getBaselineResizeBehavior()](http://docs.google.com/java/awt/Component.html#getBaselineResizeBehavior()), [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html)

### getBaselineResizeBehavior

public [Component.BaselineResizeBehavior](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html) **getBaselineResizeBehavior**()

Returns an enum indicating how the baseline of the component changes as the size changes. This method is primarily meant for layout managers and GUI builders.

The default implementation returns BaselineResizeBehavior.OTHER. Subclasses that have a baseline should override appropriately. Subclasses should never return null; if the baseline can not be calculated return BaselineResizeBehavior.OTHER. Callers should first ask for the baseline using getBaseline and if a value >= 0 is returned use this method. It is acceptable for this method to return a value other than BaselineResizeBehavior.OTHER even if getBaseline returns a value less than 0.

**Returns:**an enum indicating how the baseline changes as the component size changes**Since:** 1.6 **See Also:**[getBaseline(int, int)](http://docs.google.com/java/awt/Component.html#getBaseline(int,%20int))

### doLayout

public void **doLayout**()

Prompts the layout manager to lay out this component. This is usually called when the component (more specifically, container) is validated.

**See Also:**[validate()](http://docs.google.com/java/awt/Component.html#validate()), [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html)

### layout

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **layout**()

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

### validate

public void **validate**()

Ensures that this component has a valid layout. This method is primarily intended to operate on instances of Container.

**Since:** JDK1.0 **See Also:**[invalidate()](http://docs.google.com/java/awt/Component.html#invalidate()), [doLayout()](http://docs.google.com/java/awt/Component.html#doLayout()), [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html), [Container.validate()](http://docs.google.com/java/awt/Container.html#validate())

### invalidate

public void **invalidate**()

Invalidates this component. This component and all parents above it are marked as needing to be laid out. This method can be called often, so it needs to execute quickly.

**Since:** JDK1.0 **See Also:**[validate()](http://docs.google.com/java/awt/Component.html#validate()), [doLayout()](http://docs.google.com/java/awt/Component.html#doLayout()), [LayoutManager](http://docs.google.com/java/awt/LayoutManager.html)

### getGraphics

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

Creates a graphics context for this component. This method will return null if this component is currently not displayable.

**Returns:**a graphics context for this component, or null if it has none**Since:** JDK1.0 **See Also:**[paint(java.awt.Graphics)](http://docs.google.com/java/awt/Component.html#paint(java.awt.Graphics))

### getFontMetrics

public [FontMetrics](http://docs.google.com/java/awt/FontMetrics.html) **getFontMetrics**([Font](http://docs.google.com/java/awt/Font.html) font)

Gets the font metrics for the specified font. Warning: Since Font metrics are affected by the [FontRenderContext](http://docs.google.com/java/awt/font/FontRenderContext.html) and this method does not provide one, it can return only metrics for the default render context which may not match that used when rendering on the Component if [Graphics2D](http://docs.google.com/java/awt/Graphics2D.html) functionality is being used. Instead metrics can be obtained at rendering time by calling [Graphics.getFontMetrics()](http://docs.google.com/java/awt/Graphics.html#getFontMetrics()) or text measurement APIs on the [Font](http://docs.google.com/java/awt/Font.html) class.

**Parameters:**font - the font for which font metrics is to be obtained **Returns:**the font metrics for font**Since:** JDK1.0 **See Also:**[getFont()](http://docs.google.com/java/awt/Component.html#getFont()), [getPeer()](http://docs.google.com/java/awt/Component.html#getPeer()), ComponentPeer.getFontMetrics(Font), [Toolkit.getFontMetrics(Font)](http://docs.google.com/java/awt/Toolkit.html#getFontMetrics(java.awt.Font))

### setCursor

public void **setCursor**([Cursor](http://docs.google.com/java/awt/Cursor.html) cursor)

Sets the cursor image to the specified cursor. This cursor image is displayed when the contains method for this component returns true for the current cursor location, and this Component is visible, displayable, and enabled. Setting the cursor of a Container causes that cursor to be displayed within all of the container's subcomponents, except for those that have a non-null cursor.

The method may have no visual effect if the Java platform implementation and/or the native system do not support changing the mouse cursor shape.

**Parameters:**cursor - One of the constants defined by the Cursor class; if this parameter is null then this component will inherit the cursor of its parent**Since:** JDK1.1 **See Also:**[isEnabled()](http://docs.google.com/java/awt/Component.html#isEnabled()), [isShowing()](http://docs.google.com/java/awt/Component.html#isShowing()), [getCursor()](http://docs.google.com/java/awt/Component.html#getCursor()), [contains(int, int)](http://docs.google.com/java/awt/Component.html#contains(int,%20int)), [Toolkit.createCustomCursor(java.awt.Image, java.awt.Point, java.lang.String)](http://docs.google.com/java/awt/Toolkit.html#createCustomCursor(java.awt.Image,%20java.awt.Point,%20java.lang.String)), [Cursor](http://docs.google.com/java/awt/Cursor.html)

### getCursor

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

Gets the cursor set in the component. If the component does not have a cursor set, the cursor of its parent is returned. If no cursor is set in the entire hierarchy, Cursor.DEFAULT\_CURSOR is returned.

**Since:** JDK1.1 **See Also:**[setCursor(java.awt.Cursor)](http://docs.google.com/java/awt/Component.html#setCursor(java.awt.Cursor))

### isCursorSet

public boolean **isCursorSet**()

Returns whether the cursor has been explicitly set for this Component. If this method returns false, this Component is inheriting its cursor from an ancestor.

**Returns:**true if the cursor has been explicitly set for this Component; false otherwise.**Since:** 1.4

### paint

public void **paint**([Graphics](http://docs.google.com/java/awt/Graphics.html) g)

Paints this component.

This method is called when the contents of the component should be painted; such as when the component is first being shown or is damaged and in need of repair. The clip rectangle in the Graphics parameter is set to the area which needs to be painted. Subclasses of Component that override this method need not call super.paint(g).

For performance reasons, Components with zero width or height aren't considered to need painting when they are first shown, and also aren't considered to need repair.

**Note**: For more information on the paint mechanisms utilitized by AWT and Swing, including information on how to write the most efficient painting code, see [Painting in AWT and Swing](http://java.sun.com/products/jfc/tsc/articles/painting/index.html).

**Parameters:**g - the graphics context to use for painting**Since:** JDK1.0 **See Also:**[update(java.awt.Graphics)](http://docs.google.com/java/awt/Component.html#update(java.awt.Graphics))

### update

public void **update**([Graphics](http://docs.google.com/java/awt/Graphics.html) g)

Updates this component.

If this component is not a lightweight component, the AWT calls the update method in response to a call to repaint. You can assume that the background is not cleared.

The update method of Component calls this component's paint method to redraw this component. This method is commonly overridden by subclasses which need to do additional work in response to a call to repaint. Subclasses of Component that override this method should either call super.update(g), or call paint(g) directly from their update method.

The origin of the graphics context, its (0, 0) coordinate point, is the top-left corner of this component. The clipping region of the graphics context is the bounding rectangle of this component.

**Note**: For more information on the paint mechanisms utilitized by AWT and Swing, including information on how to write the most efficient painting code, see [Painting in AWT and Swing](http://java.sun.com/products/jfc/tsc/articles/painting/index.html).

**Parameters:**g - the specified context to use for updating**Since:** JDK1.0 **See Also:**[paint(java.awt.Graphics)](http://docs.google.com/java/awt/Component.html#paint(java.awt.Graphics)), [repaint()](http://docs.google.com/java/awt/Component.html#repaint())

### paintAll

public void **paintAll**([Graphics](http://docs.google.com/java/awt/Graphics.html) g)

Paints this component and all of its subcomponents.

The origin of the graphics context, its (0, 0) coordinate point, is the top-left corner of this component. The clipping region of the graphics context is the bounding rectangle of this component.

**Parameters:**g - the graphics context to use for painting**Since:** JDK1.0 **See Also:**[paint(java.awt.Graphics)](http://docs.google.com/java/awt/Component.html#paint(java.awt.Graphics))

### repaint

public void **repaint**()

Repaints this component.

If this component is a lightweight component, this method causes a call to this component's paint method as soon as possible. Otherwise, this method causes a call to this component's update method as soon as possible.

**Note**: For more information on the paint mechanisms utilitized by AWT and Swing, including information on how to write the most efficient painting code, see [Painting in AWT and Swing](http://java.sun.com/products/jfc/tsc/articles/painting/index.html).

**Since:** JDK1.0 **See Also:**[update(Graphics)](http://docs.google.com/java/awt/Component.html#update(java.awt.Graphics))

### repaint

public void **repaint**(long tm)

Repaints the component. If this component is a lightweight component, this results in a call to paint within tm milliseconds.

**Note**: For more information on the paint mechanisms utilitized by AWT and Swing, including information on how to write the most efficient painting code, see [Painting in AWT and Swing](http://java.sun.com/products/jfc/tsc/articles/painting/index.html).

**Parameters:**tm - maximum time in milliseconds before update**Since:** JDK1.0 **See Also:**[paint(java.awt.Graphics)](http://docs.google.com/java/awt/Component.html#paint(java.awt.Graphics)), [update(Graphics)](http://docs.google.com/java/awt/Component.html#update(java.awt.Graphics))

### repaint

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

Repaints the specified rectangle of this component.

If this component is a lightweight component, this method causes a call to this component's paint method as soon as possible. Otherwise, this method causes a call to this component's update method as soon as possible.

**Note**: For more information on the paint mechanisms utilitized by AWT and Swing, including information on how to write the most efficient painting code, see [Painting in AWT and Swing](http://java.sun.com/products/jfc/tsc/articles/painting/index.html).

**Parameters:**x - the *x* coordinatey - the *y* coordinatewidth - the widthheight - the height**Since:** JDK1.0 **See Also:**[update(Graphics)](http://docs.google.com/java/awt/Component.html#update(java.awt.Graphics))

### repaint

public void **repaint**(long tm,  
 int x,  
 int y,  
 int width,  
 int height)

Repaints the specified rectangle of this component within tm milliseconds.

If this component is a lightweight component, this method causes a call to this component's paint method. Otherwise, this method causes a call to this component's update method.

**Note**: For more information on the paint mechanisms utilitized by AWT and Swing, including information on how to write the most efficient painting code, see [Painting in AWT and Swing](http://java.sun.com/products/jfc/tsc/articles/painting/index.html).

**Parameters:**tm - maximum time in milliseconds before updatex - the *x* coordinatey - the *y* coordinatewidth - the widthheight - the height**Since:** JDK1.0 **See Also:**[update(Graphics)](http://docs.google.com/java/awt/Component.html#update(java.awt.Graphics))

### print

public void **print**([Graphics](http://docs.google.com/java/awt/Graphics.html) g)

Prints this component. Applications should override this method for components that must do special processing before being printed or should be printed differently than they are painted.

The default implementation of this method calls the paint method.

The origin of the graphics context, its (0, 0) coordinate point, is the top-left corner of this component. The clipping region of the graphics context is the bounding rectangle of this component.

**Parameters:**g - the graphics context to use for printing**Since:** JDK1.0 **See Also:**[paint(Graphics)](http://docs.google.com/java/awt/Component.html#paint(java.awt.Graphics))

### printAll

public void **printAll**([Graphics](http://docs.google.com/java/awt/Graphics.html) g)

Prints this component and all of its subcomponents.

The origin of the graphics context, its (0, 0) coordinate point, is the top-left corner of this component. The clipping region of the graphics context is the bounding rectangle of this component.

**Parameters:**g - the graphics context to use for printing**Since:** JDK1.0 **See Also:**[print(Graphics)](http://docs.google.com/java/awt/Component.html#print(java.awt.Graphics))

### imageUpdate

public boolean **imageUpdate**([Image](http://docs.google.com/java/awt/Image.html) img,  
 int infoflags,  
 int x,  
 int y,  
 int w,  
 int h)

Repaints the component when the image has changed. This imageUpdate method of an ImageObserver is called when more information about an image which had been previously requested using an asynchronous routine such as the drawImage method of Graphics becomes available. See the definition of imageUpdate for more information on this method and its arguments.

The imageUpdate method of Component incrementally draws an image on the component as more of the bits of the image are available.

If the system property awt.image.incrementaldraw is missing or has the value true, the image is incrementally drawn. If the system property has any other value, then the image is not drawn until it has been completely loaded.

Also, if incremental drawing is in effect, the value of the system property awt.image.redrawrate is interpreted as an integer to give the maximum redraw rate, in milliseconds. If the system property is missing or cannot be interpreted as an integer, the redraw rate is once every 100ms.

The interpretation of the x, y, width, and height arguments depends on the value of the infoflags argument.

**Specified by:**[imageUpdate](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)) in interface [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) **Parameters:**img - the image being observedinfoflags - see imageUpdate for more informationx - the *x* coordinatey - the *y* coordinatew - the widthh - the height **Returns:**false if the infoflags indicate that the image is completely loaded; true otherwise.**Since:** JDK1.0 **See Also:**[ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [Graphics.drawImage(Image, int, int, Color, java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Graphics.html#drawImage(java.awt.Image,%20int,%20int,%20java.awt.Color,%20java.awt.image.ImageObserver)), [Graphics.drawImage(Image, int, int, java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Graphics.html#drawImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [Graphics.drawImage(Image, int, int, int, int, Color, java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Graphics.html#drawImage(java.awt.Image,%20int,%20int,%20int,%20int,%20java.awt.Color,%20java.awt.image.ImageObserver)), [Graphics.drawImage(Image, int, int, int, int, java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Graphics.html#drawImage(java.awt.Image,%20int,%20int,%20int,%20int,%20java.awt.image.ImageObserver)), [ImageObserver.imageUpdate(java.awt.Image, int, int, int, int, int)](http://docs.google.com/java/awt/image/ImageObserver.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int))

### createImage

public [Image](http://docs.google.com/java/awt/Image.html) **createImage**([ImageProducer](http://docs.google.com/java/awt/image/ImageProducer.html) producer)

Creates an image from the specified image producer.

**Parameters:**producer - the image producer **Returns:**the image produced**Since:** JDK1.0

### createImage

public [Image](http://docs.google.com/java/awt/Image.html) **createImage**(int width,  
 int height)

Creates an off-screen drawable image to be used for double buffering.

**Parameters:**width - the specified widthheight - the specified height **Returns:**an off-screen drawable image, which can be used for double buffering. The return value may be null if the component is not displayable. This will always happen if GraphicsEnvironment.isHeadless() returns true.**Since:** JDK1.0 **See Also:**[isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable()), [GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### createVolatileImage

public [VolatileImage](http://docs.google.com/java/awt/image/VolatileImage.html) **createVolatileImage**(int width,  
 int height)

Creates a volatile off-screen drawable image to be used for double buffering.

**Parameters:**width - the specified width.height - the specified height. **Returns:**an off-screen drawable image, which can be used for double buffering. The return value may be null if the component is not displayable. This will always happen if GraphicsEnvironment.isHeadless() returns true.**Since:** 1.4 **See Also:**[VolatileImage](http://docs.google.com/java/awt/image/VolatileImage.html), [isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable()), [GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### createVolatileImage

public [VolatileImage](http://docs.google.com/java/awt/image/VolatileImage.html) **createVolatileImage**(int width,  
 int height,  
 [ImageCapabilities](http://docs.google.com/java/awt/ImageCapabilities.html) caps)  
 throws [AWTException](http://docs.google.com/java/awt/AWTException.html)

Creates a volatile off-screen drawable image, with the given capabilities. The contents of this image may be lost at any time due to operating system issues, so the image must be managed via the VolatileImage interface.

**Parameters:**width - the specified width.height - the specified height.caps - the image capabilities **Returns:**a VolatileImage object, which can be used to manage surface contents loss and capabilities. **Throws:** [AWTException](http://docs.google.com/java/awt/AWTException.html) - if an image with the specified capabilities cannot be created**Since:** 1.4 **See Also:**[VolatileImage](http://docs.google.com/java/awt/image/VolatileImage.html)

### prepareImage

public boolean **prepareImage**([Image](http://docs.google.com/java/awt/Image.html) image,  
 [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)

Prepares an image for rendering on this component. The image data is downloaded asynchronously in another thread and the appropriate screen representation of the image is generated.

**Parameters:**image - the Image for which to prepare a screen representationobserver - the ImageObserver object to be notified as the image is being prepared **Returns:**true if the image has already been fully prepared; false otherwise**Since:** JDK1.0

### prepareImage

public boolean **prepareImage**([Image](http://docs.google.com/java/awt/Image.html) image,  
 int width,  
 int height,  
 [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)

Prepares an image for rendering on this component at the specified width and height.

The image data is downloaded asynchronously in another thread, and an appropriately scaled screen representation of the image is generated.

**Parameters:**image - the instance of Image for which to prepare a screen representationwidth - the width of the desired screen representationheight - the height of the desired screen representationobserver - the ImageObserver object to be notified as the image is being prepared **Returns:**true if the image has already been fully prepared; false otherwise**Since:** JDK1.0 **See Also:**[ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html)

### checkImage

public int **checkImage**([Image](http://docs.google.com/java/awt/Image.html) image,  
 [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)

Returns the status of the construction of a screen representation of the specified image.

This method does not cause the image to begin loading. An application must use the prepareImage method to force the loading of an image.

Information on the flags returned by this method can be found with the discussion of the ImageObserver interface.

**Parameters:**image - the Image object whose status is being checkedobserver - the ImageObserver object to be notified as the image is being prepared **Returns:**the bitwise inclusive **OR** of ImageObserver flags indicating what information about the image is currently available**Since:** JDK1.0 **See Also:**[prepareImage(Image, int, int, java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [Toolkit.checkImage(Image, int, int, java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Toolkit.html#checkImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html)

### checkImage

public int **checkImage**([Image](http://docs.google.com/java/awt/Image.html) image,  
 int width,  
 int height,  
 [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html) observer)

Returns the status of the construction of a screen representation of the specified image.

This method does not cause the image to begin loading. An application must use the prepareImage method to force the loading of an image.

The checkImage method of Component calls its peer's checkImage method to calculate the flags. If this component does not yet have a peer, the component's toolkit's checkImage method is called instead.

Information on the flags returned by this method can be found with the discussion of the ImageObserver interface.

**Parameters:**image - the Image object whose status is being checkedwidth - the width of the scaled version whose status is to be checkedheight - the height of the scaled version whose status is to be checkedobserver - the ImageObserver object to be notified as the image is being prepared **Returns:**the bitwise inclusive **OR** of ImageObserver flags indicating what information about the image is currently available**Since:** JDK1.0 **See Also:**[prepareImage(Image, int, int, java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [Toolkit.checkImage(Image, int, int, java.awt.image.ImageObserver)](http://docs.google.com/java/awt/Toolkit.html#checkImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html)

### setIgnoreRepaint

public void **setIgnoreRepaint**(boolean ignoreRepaint)

Sets whether or not paint messages received from the operating system should be ignored. This does not affect paint events generated in software by the AWT, unless they are an immediate response to an OS-level paint message.

This is useful, for example, if running under full-screen mode and better performance is desired, or if page-flipping is used as the buffer strategy.

**Since:** 1.4 **See Also:**[getIgnoreRepaint()](http://docs.google.com/java/awt/Component.html#getIgnoreRepaint()), [Canvas.createBufferStrategy(int)](http://docs.google.com/java/awt/Canvas.html#createBufferStrategy(int)), [Window.createBufferStrategy(int)](http://docs.google.com/java/awt/Window.html#createBufferStrategy(int)), [BufferStrategy](http://docs.google.com/java/awt/image/BufferStrategy.html), [GraphicsDevice.setFullScreenWindow(java.awt.Window)](http://docs.google.com/java/awt/GraphicsDevice.html#setFullScreenWindow(java.awt.Window))

### getIgnoreRepaint

public boolean **getIgnoreRepaint**()

**Returns:**whether or not paint messages received from the operating system should be ignored.**Since:** 1.4 **See Also:**[setIgnoreRepaint(boolean)](http://docs.google.com/java/awt/Component.html#setIgnoreRepaint(boolean))

### contains

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

Checks whether this component "contains" the specified point, where x and y are defined to be relative to the coordinate system of this component.

**Parameters:**x - the *x* coordinate of the pointy - the *y* coordinate of the point**Since:** JDK1.1 **See Also:**[getComponentAt(int, int)](http://docs.google.com/java/awt/Component.html#getComponentAt(int,%20int))

### 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).*

### contains

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

Checks whether this component "contains" the specified point, where the point's *x* and *y* coordinates are defined to be relative to the coordinate system of this component.

**Parameters:**p - the point**Since:** JDK1.1 **See Also:**[getComponentAt(Point)](http://docs.google.com/java/awt/Component.html#getComponentAt(java.awt.Point))

### getComponentAt

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

Determines if this component or one of its immediate subcomponents contains the (*x*, *y*) location, and if so, returns the containing component. This method only looks one level deep. If the point (*x*, *y*) is inside a subcomponent that itself has subcomponents, it does not go looking down the subcomponent tree.

The locate method of Component simply returns the component itself if the (*x*, *y*) coordinate location is inside its bounding box, and null otherwise.

**Parameters:**x - the *x* coordinatey - the *y* coordinate **Returns:**the component or subcomponent that contains the (*x*, *y*) location; null if the location is outside this component**Since:** JDK1.0 **See Also:**[contains(int, int)](http://docs.google.com/java/awt/Component.html#contains(int,%20int))

### locate

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

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

### getComponentAt

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

Returns the component or subcomponent that contains the specified point.

**Parameters:**p - the point**Since:** JDK1.1 **See Also:**[contains(int, int)](http://docs.google.com/java/awt/Component.html#contains(int,%20int))

### deliverEvent

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **deliverEvent**([Event](http://docs.google.com/java/awt/Event.html) e)

**Deprecated.** *As of JDK version 1.1, replaced by dispatchEvent(AWTEvent e).*

### dispatchEvent

public final void **dispatchEvent**([AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) e)

Dispatches an event to this component or one of its sub components. Calls processEvent before returning for 1.1-style events which have been enabled for the Component.

**Parameters:**e - the event

### postEvent

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public boolean **postEvent**([Event](http://docs.google.com/java/awt/Event.html) e)

**Deprecated.** *As of JDK version 1.1, replaced by dispatchEvent(AWTEvent).*

**Specified by:**[postEvent](http://docs.google.com/java/awt/MenuContainer.html#postEvent(java.awt.Event)) in interface [MenuContainer](http://docs.google.com/java/awt/MenuContainer.html)

### addComponentListener

public void **addComponentListener**([ComponentListener](http://docs.google.com/java/awt/event/ComponentListener.html) l)

Adds the specified component listener to receive component events from this component. If listener l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the component listener**Since:** JDK1.1 **See Also:**[ComponentEvent](http://docs.google.com/java/awt/event/ComponentEvent.html), [ComponentListener](http://docs.google.com/java/awt/event/ComponentListener.html), [removeComponentListener(java.awt.event.ComponentListener)](http://docs.google.com/java/awt/Component.html#removeComponentListener(java.awt.event.ComponentListener)), [getComponentListeners()](http://docs.google.com/java/awt/Component.html#getComponentListeners())

### removeComponentListener

public void **removeComponentListener**([ComponentListener](http://docs.google.com/java/awt/event/ComponentListener.html) l)

Removes the specified component listener so that it no longer receives component events from this component. This method performs no function, nor does it throw an exception, if the listener specified by the argument was not previously added to this component. If listener l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the component listener**Since:** JDK1.1 **See Also:**[ComponentEvent](http://docs.google.com/java/awt/event/ComponentEvent.html), [ComponentListener](http://docs.google.com/java/awt/event/ComponentListener.html), [addComponentListener(java.awt.event.ComponentListener)](http://docs.google.com/java/awt/Component.html#addComponentListener(java.awt.event.ComponentListener)), [getComponentListeners()](http://docs.google.com/java/awt/Component.html#getComponentListeners())

### getComponentListeners

public [ComponentListener](http://docs.google.com/java/awt/event/ComponentListener.html)[] **getComponentListeners**()

Returns an array of all the component listeners registered on this component.

**Returns:**all of this comonent's ComponentListeners or an empty array if no component listeners are currently registered**Since:** 1.4 **See Also:**[addComponentListener(java.awt.event.ComponentListener)](http://docs.google.com/java/awt/Component.html#addComponentListener(java.awt.event.ComponentListener)), [removeComponentListener(java.awt.event.ComponentListener)](http://docs.google.com/java/awt/Component.html#removeComponentListener(java.awt.event.ComponentListener))

### addFocusListener

public void **addFocusListener**([FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) l)

Adds the specified focus listener to receive focus events from this component when this component gains input focus. If listener l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the focus listener**Since:** JDK1.1 **See Also:**[FocusEvent](http://docs.google.com/java/awt/event/FocusEvent.html), [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html), [removeFocusListener(java.awt.event.FocusListener)](http://docs.google.com/java/awt/Component.html#removeFocusListener(java.awt.event.FocusListener)), [getFocusListeners()](http://docs.google.com/java/awt/Component.html#getFocusListeners())

### removeFocusListener

public void **removeFocusListener**([FocusListener](http://docs.google.com/java/awt/event/FocusListener.html) l)

Removes the specified focus listener so that it no longer receives focus events from this component. This method performs no function, nor does it throw an exception, if the listener specified by the argument was not previously added to this component. If listener l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the focus listener**Since:** JDK1.1 **See Also:**[FocusEvent](http://docs.google.com/java/awt/event/FocusEvent.html), [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html), [addFocusListener(java.awt.event.FocusListener)](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [getFocusListeners()](http://docs.google.com/java/awt/Component.html#getFocusListeners())

### getFocusListeners

public [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html)[] **getFocusListeners**()

Returns an array of all the focus listeners registered on this component.

**Returns:**all of this component's FocusListeners or an empty array if no component listeners are currently registered**Since:** 1.4 **See Also:**[addFocusListener(java.awt.event.FocusListener)](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [removeFocusListener(java.awt.event.FocusListener)](http://docs.google.com/java/awt/Component.html#removeFocusListener(java.awt.event.FocusListener))

### addHierarchyListener

public void **addHierarchyListener**([HierarchyListener](http://docs.google.com/java/awt/event/HierarchyListener.html) l)

Adds the specified hierarchy listener to receive hierarchy changed events from this component when the hierarchy to which this container belongs changes. If listener l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the hierarchy listener**Since:** 1.3 **See Also:**[HierarchyEvent](http://docs.google.com/java/awt/event/HierarchyEvent.html), [HierarchyListener](http://docs.google.com/java/awt/event/HierarchyListener.html), [removeHierarchyListener(java.awt.event.HierarchyListener)](http://docs.google.com/java/awt/Component.html#removeHierarchyListener(java.awt.event.HierarchyListener)), [getHierarchyListeners()](http://docs.google.com/java/awt/Component.html#getHierarchyListeners())

### removeHierarchyListener

public void **removeHierarchyListener**([HierarchyListener](http://docs.google.com/java/awt/event/HierarchyListener.html) l)

Removes the specified hierarchy listener so that it no longer receives hierarchy changed events from this component. This method performs no function, nor does it throw an exception, if the listener specified by the argument was not previously added to this component. If listener l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the hierarchy listener**Since:** 1.3 **See Also:**[HierarchyEvent](http://docs.google.com/java/awt/event/HierarchyEvent.html), [HierarchyListener](http://docs.google.com/java/awt/event/HierarchyListener.html), [addHierarchyListener(java.awt.event.HierarchyListener)](http://docs.google.com/java/awt/Component.html#addHierarchyListener(java.awt.event.HierarchyListener)), [getHierarchyListeners()](http://docs.google.com/java/awt/Component.html#getHierarchyListeners())

### getHierarchyListeners

public [HierarchyListener](http://docs.google.com/java/awt/event/HierarchyListener.html)[] **getHierarchyListeners**()

Returns an array of all the hierarchy listeners registered on this component.

**Returns:**all of this component's HierarchyListeners or an empty array if no hierarchy listeners are currently registered**Since:** 1.4 **See Also:**[addHierarchyListener(java.awt.event.HierarchyListener)](http://docs.google.com/java/awt/Component.html#addHierarchyListener(java.awt.event.HierarchyListener)), [removeHierarchyListener(java.awt.event.HierarchyListener)](http://docs.google.com/java/awt/Component.html#removeHierarchyListener(java.awt.event.HierarchyListener))

### addHierarchyBoundsListener

public void **addHierarchyBoundsListener**([HierarchyBoundsListener](http://docs.google.com/java/awt/event/HierarchyBoundsListener.html) l)

Adds the specified hierarchy bounds listener to receive hierarchy bounds events from this component when the hierarchy to which this container belongs changes. If listener l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the hierarchy bounds listener**Since:** 1.3 **See Also:**[HierarchyEvent](http://docs.google.com/java/awt/event/HierarchyEvent.html), [HierarchyBoundsListener](http://docs.google.com/java/awt/event/HierarchyBoundsListener.html), [removeHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)](http://docs.google.com/java/awt/Component.html#removeHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [getHierarchyBoundsListeners()](http://docs.google.com/java/awt/Component.html#getHierarchyBoundsListeners())

### removeHierarchyBoundsListener

public void **removeHierarchyBoundsListener**([HierarchyBoundsListener](http://docs.google.com/java/awt/event/HierarchyBoundsListener.html) l)

Removes the specified hierarchy bounds listener so that it no longer receives hierarchy bounds events from this component. This method performs no function, nor does it throw an exception, if the listener specified by the argument was not previously added to this component. If listener l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the hierarchy bounds listener**Since:** 1.3 **See Also:**[HierarchyEvent](http://docs.google.com/java/awt/event/HierarchyEvent.html), [HierarchyBoundsListener](http://docs.google.com/java/awt/event/HierarchyBoundsListener.html), [addHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)](http://docs.google.com/java/awt/Component.html#addHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [getHierarchyBoundsListeners()](http://docs.google.com/java/awt/Component.html#getHierarchyBoundsListeners())

### getHierarchyBoundsListeners

public [HierarchyBoundsListener](http://docs.google.com/java/awt/event/HierarchyBoundsListener.html)[] **getHierarchyBoundsListeners**()

Returns an array of all the hierarchy bounds listeners registered on this component.

**Returns:**all of this component's HierarchyBoundsListeners or an empty array if no hierarchy bounds listeners are currently registered**Since:** 1.4 **See Also:**[addHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)](http://docs.google.com/java/awt/Component.html#addHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [removeHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)](http://docs.google.com/java/awt/Component.html#removeHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener))

### addKeyListener

public void **addKeyListener**([KeyListener](http://docs.google.com/java/awt/event/KeyListener.html) l)

Adds the specified key listener to receive key events from this component. If l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the key listener.**Since:** JDK1.1 **See Also:**[KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html), [KeyListener](http://docs.google.com/java/awt/event/KeyListener.html), [removeKeyListener(java.awt.event.KeyListener)](http://docs.google.com/java/awt/Component.html#removeKeyListener(java.awt.event.KeyListener)), [getKeyListeners()](http://docs.google.com/java/awt/Component.html#getKeyListeners())

### removeKeyListener

public void **removeKeyListener**([KeyListener](http://docs.google.com/java/awt/event/KeyListener.html) l)

Removes the specified key listener so that it no longer receives key events from this component. This method performs no function, nor does it throw an exception, if the listener specified by the argument was not previously added to this component. If listener l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the key listener**Since:** JDK1.1 **See Also:**[KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html), [KeyListener](http://docs.google.com/java/awt/event/KeyListener.html), [addKeyListener(java.awt.event.KeyListener)](http://docs.google.com/java/awt/Component.html#addKeyListener(java.awt.event.KeyListener)), [getKeyListeners()](http://docs.google.com/java/awt/Component.html#getKeyListeners())

### getKeyListeners

public [KeyListener](http://docs.google.com/java/awt/event/KeyListener.html)[] **getKeyListeners**()

Returns an array of all the key listeners registered on this component.

**Returns:**all of this component's KeyListeners or an empty array if no key listeners are currently registered**Since:** 1.4 **See Also:**[addKeyListener(java.awt.event.KeyListener)](http://docs.google.com/java/awt/Component.html#addKeyListener(java.awt.event.KeyListener)), [removeKeyListener(java.awt.event.KeyListener)](http://docs.google.com/java/awt/Component.html#removeKeyListener(java.awt.event.KeyListener))

### addMouseListener

public void **addMouseListener**([MouseListener](http://docs.google.com/java/awt/event/MouseListener.html) l)

Adds the specified mouse listener to receive mouse events from this component. If listener l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the mouse listener**Since:** JDK1.1 **See Also:**[MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html), [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html), [removeMouseListener(java.awt.event.MouseListener)](http://docs.google.com/java/awt/Component.html#removeMouseListener(java.awt.event.MouseListener)), [getMouseListeners()](http://docs.google.com/java/awt/Component.html#getMouseListeners())

### removeMouseListener

public void **removeMouseListener**([MouseListener](http://docs.google.com/java/awt/event/MouseListener.html) l)

Removes the specified mouse listener so that it no longer receives mouse events from this component. This method performs no function, nor does it throw an exception, if the listener specified by the argument was not previously added to this component. If listener l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the mouse listener**Since:** JDK1.1 **See Also:**[MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html), [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html), [addMouseListener(java.awt.event.MouseListener)](http://docs.google.com/java/awt/Component.html#addMouseListener(java.awt.event.MouseListener)), [getMouseListeners()](http://docs.google.com/java/awt/Component.html#getMouseListeners())

### getMouseListeners

public [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html)[] **getMouseListeners**()

Returns an array of all the mouse listeners registered on this component.

**Returns:**all of this component's MouseListeners or an empty array if no mouse listeners are currently registered**Since:** 1.4 **See Also:**[addMouseListener(java.awt.event.MouseListener)](http://docs.google.com/java/awt/Component.html#addMouseListener(java.awt.event.MouseListener)), [removeMouseListener(java.awt.event.MouseListener)](http://docs.google.com/java/awt/Component.html#removeMouseListener(java.awt.event.MouseListener))

### addMouseMotionListener

public void **addMouseMotionListener**([MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html) l)

Adds the specified mouse motion listener to receive mouse motion events from this component. If listener l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the mouse motion listener**Since:** JDK1.1 **See Also:**[MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html), [MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html), [removeMouseMotionListener(java.awt.event.MouseMotionListener)](http://docs.google.com/java/awt/Component.html#removeMouseMotionListener(java.awt.event.MouseMotionListener)), [getMouseMotionListeners()](http://docs.google.com/java/awt/Component.html#getMouseMotionListeners())

### removeMouseMotionListener

public void **removeMouseMotionListener**([MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html) l)

Removes the specified mouse motion listener so that it no longer receives mouse motion events from this component. This method performs no function, nor does it throw an exception, if the listener specified by the argument was not previously added to this component. If listener l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the mouse motion listener**Since:** JDK1.1 **See Also:**[MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html), [MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html), [addMouseMotionListener(java.awt.event.MouseMotionListener)](http://docs.google.com/java/awt/Component.html#addMouseMotionListener(java.awt.event.MouseMotionListener)), [getMouseMotionListeners()](http://docs.google.com/java/awt/Component.html#getMouseMotionListeners())

### getMouseMotionListeners

public [MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html)[] **getMouseMotionListeners**()

Returns an array of all the mouse motion listeners registered on this component.

**Returns:**all of this component's MouseMotionListeners or an empty array if no mouse motion listeners are currently registered**Since:** 1.4 **See Also:**[addMouseMotionListener(java.awt.event.MouseMotionListener)](http://docs.google.com/java/awt/Component.html#addMouseMotionListener(java.awt.event.MouseMotionListener)), [removeMouseMotionListener(java.awt.event.MouseMotionListener)](http://docs.google.com/java/awt/Component.html#removeMouseMotionListener(java.awt.event.MouseMotionListener))

### addMouseWheelListener

public void **addMouseWheelListener**([MouseWheelListener](http://docs.google.com/java/awt/event/MouseWheelListener.html) l)

Adds the specified mouse wheel listener to receive mouse wheel events from this component. Containers also receive mouse wheel events from sub-components.

For information on how mouse wheel events are dispatched, see the class description for [MouseWheelEvent](http://docs.google.com/java/awt/event/MouseWheelEvent.html).

If l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the mouse wheel listener**Since:** 1.4 **See Also:**[MouseWheelEvent](http://docs.google.com/java/awt/event/MouseWheelEvent.html), [MouseWheelListener](http://docs.google.com/java/awt/event/MouseWheelListener.html), [removeMouseWheelListener(java.awt.event.MouseWheelListener)](http://docs.google.com/java/awt/Component.html#removeMouseWheelListener(java.awt.event.MouseWheelListener)), [getMouseWheelListeners()](http://docs.google.com/java/awt/Component.html#getMouseWheelListeners())

### removeMouseWheelListener

public void **removeMouseWheelListener**([MouseWheelListener](http://docs.google.com/java/awt/event/MouseWheelListener.html) l)

Removes the specified mouse wheel listener so that it no longer receives mouse wheel events from this component. This method performs no function, nor does it throw an exception, if the listener specified by the argument was not previously added to this component. If l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the mouse wheel listener.**Since:** 1.4 **See Also:**[MouseWheelEvent](http://docs.google.com/java/awt/event/MouseWheelEvent.html), [MouseWheelListener](http://docs.google.com/java/awt/event/MouseWheelListener.html), [addMouseWheelListener(java.awt.event.MouseWheelListener)](http://docs.google.com/java/awt/Component.html#addMouseWheelListener(java.awt.event.MouseWheelListener)), [getMouseWheelListeners()](http://docs.google.com/java/awt/Component.html#getMouseWheelListeners())

### getMouseWheelListeners

public [MouseWheelListener](http://docs.google.com/java/awt/event/MouseWheelListener.html)[] **getMouseWheelListeners**()

Returns an array of all the mouse wheel listeners registered on this component.

**Returns:**all of this component's MouseWheelListeners or an empty array if no mouse wheel listeners are currently registered**Since:** 1.4 **See Also:**[addMouseWheelListener(java.awt.event.MouseWheelListener)](http://docs.google.com/java/awt/Component.html#addMouseWheelListener(java.awt.event.MouseWheelListener)), [removeMouseWheelListener(java.awt.event.MouseWheelListener)](http://docs.google.com/java/awt/Component.html#removeMouseWheelListener(java.awt.event.MouseWheelListener))

### addInputMethodListener

public void **addInputMethodListener**([InputMethodListener](http://docs.google.com/java/awt/event/InputMethodListener.html) l)

Adds the specified input method listener to receive input method events from this component. A component will only receive input method events from input methods if it also overrides getInputMethodRequests to return an InputMethodRequests instance. If listener l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the input method listener**Since:** 1.2 **See Also:**[InputMethodEvent](http://docs.google.com/java/awt/event/InputMethodEvent.html), [InputMethodListener](http://docs.google.com/java/awt/event/InputMethodListener.html), [removeInputMethodListener(java.awt.event.InputMethodListener)](http://docs.google.com/java/awt/Component.html#removeInputMethodListener(java.awt.event.InputMethodListener)), [getInputMethodListeners()](http://docs.google.com/java/awt/Component.html#getInputMethodListeners()), [getInputMethodRequests()](http://docs.google.com/java/awt/Component.html#getInputMethodRequests())

### removeInputMethodListener

public void **removeInputMethodListener**([InputMethodListener](http://docs.google.com/java/awt/event/InputMethodListener.html) l)

Removes the specified input method listener so that it no longer receives input method events from this component. This method performs no function, nor does it throw an exception, if the listener specified by the argument was not previously added to this component. If listener l is null, no exception is thrown and no action is performed.

Refer to [AWT Threading Issues](http://docs.google.com/doc-files/AWTThreadIssues.html#ListenersThreads) for details on AWT's threading model.

**Parameters:**l - the input method listener**Since:** 1.2 **See Also:**[InputMethodEvent](http://docs.google.com/java/awt/event/InputMethodEvent.html), [InputMethodListener](http://docs.google.com/java/awt/event/InputMethodListener.html), [addInputMethodListener(java.awt.event.InputMethodListener)](http://docs.google.com/java/awt/Component.html#addInputMethodListener(java.awt.event.InputMethodListener)), [getInputMethodListeners()](http://docs.google.com/java/awt/Component.html#getInputMethodListeners())

### getInputMethodListeners

public [InputMethodListener](http://docs.google.com/java/awt/event/InputMethodListener.html)[] **getInputMethodListeners**()

Returns an array of all the input method listeners registered on this component.

**Returns:**all of this component's InputMethodListeners or an empty array if no input method listeners are currently registered**Since:** 1.4 **See Also:**[addInputMethodListener(java.awt.event.InputMethodListener)](http://docs.google.com/java/awt/Component.html#addInputMethodListener(java.awt.event.InputMethodListener)), [removeInputMethodListener(java.awt.event.InputMethodListener)](http://docs.google.com/java/awt/Component.html#removeInputMethodListener(java.awt.event.InputMethodListener))

### getListeners

public <T extends [EventListener](http://docs.google.com/java/util/EventListener.html)> T[] **getListeners**([Class](http://docs.google.com/java/lang/Class.html)<T> listenerType)

Returns an array of all the objects currently registered as *Foo*Listeners upon this Component. *Foo*Listeners are registered using the add*Foo*Listener method.

You can specify the listenerType argument with a class literal, such as *Foo*Listener.class. For example, you can query a Component c for its mouse listeners with the following code:

MouseListener[] mls = (MouseListener[])(c.getListeners(MouseListener.class));

If no such listeners exist, this method returns an empty array.

**Parameters:**listenerType - the type of listeners requested; this parameter should specify an interface that descends from java.util.EventListener **Returns:**an array of all objects registered as *Foo*Listeners on this component, or an empty array if no such listeners have been added **Throws:** [ClassCastException](http://docs.google.com/java/lang/ClassCastException.html) - if listenerType doesn't specify a class or interface that implements java.util.EventListener**Since:** 1.3 **See Also:**[getComponentListeners()](http://docs.google.com/java/awt/Component.html#getComponentListeners()), [getFocusListeners()](http://docs.google.com/java/awt/Component.html#getFocusListeners()), [getHierarchyListeners()](http://docs.google.com/java/awt/Component.html#getHierarchyListeners()), [getHierarchyBoundsListeners()](http://docs.google.com/java/awt/Component.html#getHierarchyBoundsListeners()), [getKeyListeners()](http://docs.google.com/java/awt/Component.html#getKeyListeners()), [getMouseListeners()](http://docs.google.com/java/awt/Component.html#getMouseListeners()), [getMouseMotionListeners()](http://docs.google.com/java/awt/Component.html#getMouseMotionListeners()), [getMouseWheelListeners()](http://docs.google.com/java/awt/Component.html#getMouseWheelListeners()), [getInputMethodListeners()](http://docs.google.com/java/awt/Component.html#getInputMethodListeners()), [getPropertyChangeListeners()](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners())

### getInputMethodRequests

public [InputMethodRequests](http://docs.google.com/java/awt/im/InputMethodRequests.html) **getInputMethodRequests**()

Gets the input method request handler which supports requests from input methods for this component. A component that supports on-the-spot text input must override this method to return an InputMethodRequests instance. At the same time, it also has to handle input method events.

**Returns:**the input method request handler for this component, null by default**Since:** 1.2 **See Also:**[addInputMethodListener(java.awt.event.InputMethodListener)](http://docs.google.com/java/awt/Component.html#addInputMethodListener(java.awt.event.InputMethodListener))

### getInputContext

public [InputContext](http://docs.google.com/java/awt/im/InputContext.html) **getInputContext**()

Gets the input context used by this component for handling the communication with input methods when text is entered in this component. By default, the input context used for the parent component is returned. Components may override this to return a private input context.

**Returns:**the input context used by this component; null if no context can be determined**Since:** 1.2

### enableEvents

protected final void **enableEvents**(long eventsToEnable)

Enables the events defined by the specified event mask parameter to be delivered to this component.

Event types are automatically enabled when a listener for that event type is added to the component.

This method only needs to be invoked by subclasses of Component which desire to have the specified event types delivered to processEvent regardless of whether or not a listener is registered.

**Parameters:**eventsToEnable - the event mask defining the event types**Since:** JDK1.1 **See Also:**[processEvent(java.awt.AWTEvent)](http://docs.google.com/java/awt/Component.html#processEvent(java.awt.AWTEvent)), [disableEvents(long)](http://docs.google.com/java/awt/Component.html#disableEvents(long)), [AWTEvent](http://docs.google.com/java/awt/AWTEvent.html)

### disableEvents

protected final void **disableEvents**(long eventsToDisable)

Disables the events defined by the specified event mask parameter from being delivered to this component.

**Parameters:**eventsToDisable - the event mask defining the event types**Since:** JDK1.1 **See Also:**[enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long))

### coalesceEvents

protected [AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) **coalesceEvents**([AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) existingEvent,  
 [AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) newEvent)

Potentially coalesce an event being posted with an existing event. This method is called by EventQueue.postEvent if an event with the same ID as the event to be posted is found in the queue (both events must have this component as their source). This method either returns a coalesced event which replaces the existing event (and the new event is then discarded), or null to indicate that no combining should be done (add the second event to the end of the queue). Either event parameter may be modified and returned, as the other one is discarded unless null is returned.

This implementation of coalesceEvents coalesces two event types: mouse move (and drag) events, and paint (and update) events. For mouse move events the last event is always returned, causing intermediate moves to be discarded. For paint events, the new event is coalesced into a complex RepaintArea in the peer. The new AWTEvent is always returned.

**Parameters:**existingEvent - the event already on the EventQueuenewEvent - the event being posted to the EventQueue **Returns:**a coalesced event, or null indicating that no coalescing was done

### processEvent

protected void **processEvent**([AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) e)

Processes events occurring on this component. By default this method calls the appropriate process<event type>Event method for the given class of event.

Note that if the event parameter is null the behavior is unspecified and may result in an exception.

**Parameters:**e - the event**Since:** JDK1.1 **See Also:**[processComponentEvent(java.awt.event.ComponentEvent)](http://docs.google.com/java/awt/Component.html#processComponentEvent(java.awt.event.ComponentEvent)), [processFocusEvent(java.awt.event.FocusEvent)](http://docs.google.com/java/awt/Component.html#processFocusEvent(java.awt.event.FocusEvent)), [processKeyEvent(java.awt.event.KeyEvent)](http://docs.google.com/java/awt/Component.html#processKeyEvent(java.awt.event.KeyEvent)), [processMouseEvent(java.awt.event.MouseEvent)](http://docs.google.com/java/awt/Component.html#processMouseEvent(java.awt.event.MouseEvent)), [processMouseMotionEvent(java.awt.event.MouseEvent)](http://docs.google.com/java/awt/Component.html#processMouseMotionEvent(java.awt.event.MouseEvent)), [processInputMethodEvent(java.awt.event.InputMethodEvent)](http://docs.google.com/java/awt/Component.html#processInputMethodEvent(java.awt.event.InputMethodEvent)), [processHierarchyEvent(java.awt.event.HierarchyEvent)](http://docs.google.com/java/awt/Component.html#processHierarchyEvent(java.awt.event.HierarchyEvent)), [processMouseWheelEvent(java.awt.event.MouseWheelEvent)](http://docs.google.com/java/awt/Component.html#processMouseWheelEvent(java.awt.event.MouseWheelEvent))

### processComponentEvent

protected void **processComponentEvent**([ComponentEvent](http://docs.google.com/java/awt/event/ComponentEvent.html) e)

Processes component events occurring on this component by dispatching them to any registered ComponentListener objects.

This method is not called unless component events are enabled for this component. Component events are enabled when one of the following occurs:

* A ComponentListener object is registered via addComponentListener.
* Component events are enabled via enableEvents.

Note that if the event parameter is null the behavior is unspecified and may result in an exception.

**Parameters:**e - the component event**Since:** JDK1.1 **See Also:**[ComponentEvent](http://docs.google.com/java/awt/event/ComponentEvent.html), [ComponentListener](http://docs.google.com/java/awt/event/ComponentListener.html), [addComponentListener(java.awt.event.ComponentListener)](http://docs.google.com/java/awt/Component.html#addComponentListener(java.awt.event.ComponentListener)), [enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long))

### processFocusEvent

protected void **processFocusEvent**([FocusEvent](http://docs.google.com/java/awt/event/FocusEvent.html) e)

Processes focus events occurring on this component by dispatching them to any registered FocusListener objects.

This method is not called unless focus events are enabled for this component. Focus events are enabled when one of the following occurs:

* A FocusListener object is registered via addFocusListener.
* Focus events are enabled via enableEvents.

If focus events are enabled for a Component, the current KeyboardFocusManager determines whether or not a focus event should be dispatched to registered FocusListener objects. If the events are to be dispatched, the KeyboardFocusManager calls the Component's dispatchEvent method, which results in a call to the Component's processFocusEvent method.

If focus events are enabled for a Component, calling the Component's dispatchEvent method with a FocusEvent as the argument will result in a call to the Component's processFocusEvent method regardless of the current KeyboardFocusManager.

Note that if the event parameter is null the behavior is unspecified and may result in an exception.

**Parameters:**e - the focus event**Since:** JDK1.1 **See Also:**[FocusEvent](http://docs.google.com/java/awt/event/FocusEvent.html), [FocusListener](http://docs.google.com/java/awt/event/FocusListener.html), [KeyboardFocusManager](http://docs.google.com/java/awt/KeyboardFocusManager.html), [addFocusListener(java.awt.event.FocusListener)](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long)), [dispatchEvent(java.awt.AWTEvent)](http://docs.google.com/java/awt/Component.html#dispatchEvent(java.awt.AWTEvent))

### processKeyEvent

protected void **processKeyEvent**([KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html) e)

Processes key events occurring on this component by dispatching them to any registered KeyListener objects.

This method is not called unless key events are enabled for this component. Key events are enabled when one of the following occurs:

* A KeyListener object is registered via addKeyListener.
* Key events are enabled via enableEvents.

If key events are enabled for a Component, the current KeyboardFocusManager determines whether or not a key event should be dispatched to registered KeyListener objects. The DefaultKeyboardFocusManager will not dispatch key events to a Component that is not the focus owner or is not showing.

As of J2SE 1.4, KeyEvents are redirected to the focus owner. Please see the [Focus Specification](http://docs.google.com/doc-files/FocusSpec.html) for further information.

Calling a Component's dispatchEvent method with a KeyEvent as the argument will result in a call to the Component's processKeyEvent method regardless of the current KeyboardFocusManager as long as the component is showing, focused, and enabled, and key events are enabled on it.

If the event parameter is null the behavior is unspecified and may result in an exception.

**Parameters:**e - the key event**Since:** JDK1.1 **See Also:**[KeyEvent](http://docs.google.com/java/awt/event/KeyEvent.html), [KeyListener](http://docs.google.com/java/awt/event/KeyListener.html), [KeyboardFocusManager](http://docs.google.com/java/awt/KeyboardFocusManager.html), [DefaultKeyboardFocusManager](http://docs.google.com/java/awt/DefaultKeyboardFocusManager.html), [processEvent(java.awt.AWTEvent)](http://docs.google.com/java/awt/Component.html#processEvent(java.awt.AWTEvent)), [dispatchEvent(java.awt.AWTEvent)](http://docs.google.com/java/awt/Component.html#dispatchEvent(java.awt.AWTEvent)), [addKeyListener(java.awt.event.KeyListener)](http://docs.google.com/java/awt/Component.html#addKeyListener(java.awt.event.KeyListener)), [enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long)), [isShowing()](http://docs.google.com/java/awt/Component.html#isShowing())

### processMouseEvent

protected void **processMouseEvent**([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) e)

Processes mouse events occurring on this component by dispatching them to any registered MouseListener objects.

This method is not called unless mouse events are enabled for this component. Mouse events are enabled when one of the following occurs:

* A MouseListener object is registered via addMouseListener.
* Mouse events are enabled via enableEvents.

Note that if the event parameter is null the behavior is unspecified and may result in an exception.

**Parameters:**e - the mouse event**Since:** JDK1.1 **See Also:**[MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html), [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html), [addMouseListener(java.awt.event.MouseListener)](http://docs.google.com/java/awt/Component.html#addMouseListener(java.awt.event.MouseListener)), [enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long))

### processMouseMotionEvent

protected void **processMouseMotionEvent**([MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html) e)

Processes mouse motion events occurring on this component by dispatching them to any registered MouseMotionListener objects.

This method is not called unless mouse motion events are enabled for this component. Mouse motion events are enabled when one of the following occurs:

* A MouseMotionListener object is registered via addMouseMotionListener.
* Mouse motion events are enabled via enableEvents.

Note that if the event parameter is null the behavior is unspecified and may result in an exception.

**Parameters:**e - the mouse motion event**Since:** JDK1.1 **See Also:**[MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html), [MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html), [addMouseMotionListener(java.awt.event.MouseMotionListener)](http://docs.google.com/java/awt/Component.html#addMouseMotionListener(java.awt.event.MouseMotionListener)), [enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long))

### processMouseWheelEvent

protected void **processMouseWheelEvent**([MouseWheelEvent](http://docs.google.com/java/awt/event/MouseWheelEvent.html) e)

Processes mouse wheel events occurring on this component by dispatching them to any registered MouseWheelListener objects.

This method is not called unless mouse wheel events are enabled for this component. Mouse wheel events are enabled when one of the following occurs:

* A MouseWheelListener object is registered via addMouseWheelListener.
* Mouse wheel events are enabled via enableEvents.

For information on how mouse wheel events are dispatched, see the class description for [MouseWheelEvent](http://docs.google.com/java/awt/event/MouseWheelEvent.html).

Note that if the event parameter is null the behavior is unspecified and may result in an exception.

**Parameters:**e - the mouse wheel event**Since:** 1.4 **See Also:**[MouseWheelEvent](http://docs.google.com/java/awt/event/MouseWheelEvent.html), [MouseWheelListener](http://docs.google.com/java/awt/event/MouseWheelListener.html), [addMouseWheelListener(java.awt.event.MouseWheelListener)](http://docs.google.com/java/awt/Component.html#addMouseWheelListener(java.awt.event.MouseWheelListener)), [enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long))

### processInputMethodEvent

protected void **processInputMethodEvent**([InputMethodEvent](http://docs.google.com/java/awt/event/InputMethodEvent.html) e)

Processes input method events occurring on this component by dispatching them to any registered InputMethodListener objects.

This method is not called unless input method events are enabled for this component. Input method events are enabled when one of the following occurs:

* An InputMethodListener object is registered via addInputMethodListener.
* Input method events are enabled via enableEvents.

Note that if the event parameter is null the behavior is unspecified and may result in an exception.

**Parameters:**e - the input method event**Since:** 1.2 **See Also:**[InputMethodEvent](http://docs.google.com/java/awt/event/InputMethodEvent.html), [InputMethodListener](http://docs.google.com/java/awt/event/InputMethodListener.html), [addInputMethodListener(java.awt.event.InputMethodListener)](http://docs.google.com/java/awt/Component.html#addInputMethodListener(java.awt.event.InputMethodListener)), [enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long))

### processHierarchyEvent

protected void **processHierarchyEvent**([HierarchyEvent](http://docs.google.com/java/awt/event/HierarchyEvent.html) e)

Processes hierarchy events occurring on this component by dispatching them to any registered HierarchyListener objects.

This method is not called unless hierarchy events are enabled for this component. Hierarchy events are enabled when one of the following occurs:

* An HierarchyListener object is registered via addHierarchyListener.
* Hierarchy events are enabled via enableEvents.

Note that if the event parameter is null the behavior is unspecified and may result in an exception.

**Parameters:**e - the hierarchy event**Since:** 1.3 **See Also:**[HierarchyEvent](http://docs.google.com/java/awt/event/HierarchyEvent.html), [HierarchyListener](http://docs.google.com/java/awt/event/HierarchyListener.html), [addHierarchyListener(java.awt.event.HierarchyListener)](http://docs.google.com/java/awt/Component.html#addHierarchyListener(java.awt.event.HierarchyListener)), [enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long))

### processHierarchyBoundsEvent

protected void **processHierarchyBoundsEvent**([HierarchyEvent](http://docs.google.com/java/awt/event/HierarchyEvent.html) e)

Processes hierarchy bounds events occurring on this component by dispatching them to any registered HierarchyBoundsListener objects.

This method is not called unless hierarchy bounds events are enabled for this component. Hierarchy bounds events are enabled when one of the following occurs:

* An HierarchyBoundsListener object is registered via addHierarchyBoundsListener.
* Hierarchy bounds events are enabled via enableEvents.

Note that if the event parameter is null the behavior is unspecified and may result in an exception.

**Parameters:**e - the hierarchy event**Since:** 1.3 **See Also:**[HierarchyEvent](http://docs.google.com/java/awt/event/HierarchyEvent.html), [HierarchyBoundsListener](http://docs.google.com/java/awt/event/HierarchyBoundsListener.html), [addHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)](http://docs.google.com/java/awt/Component.html#addHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long))

### handleEvent

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public boolean **handleEvent**([Event](http://docs.google.com/java/awt/Event.html) evt)

**Deprecated.** *As of JDK version 1.1 replaced by processEvent(AWTEvent).*

### mouseDown

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public boolean **mouseDown**([Event](http://docs.google.com/java/awt/Event.html) evt,  
 int x,  
 int y)

**Deprecated.** *As of JDK version 1.1, replaced by processMouseEvent(MouseEvent).*

### mouseDrag

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public boolean **mouseDrag**([Event](http://docs.google.com/java/awt/Event.html) evt,  
 int x,  
 int y)

**Deprecated.** *As of JDK version 1.1, replaced by processMouseMotionEvent(MouseEvent).*

### mouseUp

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public boolean **mouseUp**([Event](http://docs.google.com/java/awt/Event.html) evt,  
 int x,  
 int y)

**Deprecated.** *As of JDK version 1.1, replaced by processMouseEvent(MouseEvent).*

### mouseMove

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public boolean **mouseMove**([Event](http://docs.google.com/java/awt/Event.html) evt,  
 int x,  
 int y)

**Deprecated.** *As of JDK version 1.1, replaced by processMouseMotionEvent(MouseEvent).*

### mouseEnter

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public boolean **mouseEnter**([Event](http://docs.google.com/java/awt/Event.html) evt,  
 int x,  
 int y)

**Deprecated.** *As of JDK version 1.1, replaced by processMouseEvent(MouseEvent).*

### mouseExit

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public boolean **mouseExit**([Event](http://docs.google.com/java/awt/Event.html) evt,  
 int x,  
 int y)

**Deprecated.** *As of JDK version 1.1, replaced by processMouseEvent(MouseEvent).*

### keyDown

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public boolean **keyDown**([Event](http://docs.google.com/java/awt/Event.html) evt,  
 int key)

**Deprecated.** *As of JDK version 1.1, replaced by processKeyEvent(KeyEvent).*

### keyUp

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public boolean **keyUp**([Event](http://docs.google.com/java/awt/Event.html) evt,  
 int key)

**Deprecated.** *As of JDK version 1.1, replaced by processKeyEvent(KeyEvent).*

### action

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public boolean **action**([Event](http://docs.google.com/java/awt/Event.html) evt,  
 [Object](http://docs.google.com/java/lang/Object.html) what)

**Deprecated.** *As of JDK version 1.1, should register this component as ActionListener on component which fires action events.*

### addNotify

public void **addNotify**()

Makes this Component displayable by connecting it to a native screen resource. This method is called internally by the toolkit and should not be called directly by programs.

**Since:** JDK1.0 **See Also:**[isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable()), [removeNotify()](http://docs.google.com/java/awt/Component.html#removeNotify())

### removeNotify

public void **removeNotify**()

Makes this Component undisplayable by destroying it native screen resource.

This method is called by the toolkit internally and should not be called directly by programs. Code overriding this method should call super.removeNotify as the first line of the overriding method.

**Since:** JDK1.0 **See Also:**[isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable()), [addNotify()](http://docs.google.com/java/awt/Component.html#addNotify())

### gotFocus

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public boolean **gotFocus**([Event](http://docs.google.com/java/awt/Event.html) evt,  
 [Object](http://docs.google.com/java/lang/Object.html) what)

**Deprecated.** *As of JDK version 1.1, replaced by processFocusEvent(FocusEvent).*

### lostFocus

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public boolean **lostFocus**([Event](http://docs.google.com/java/awt/Event.html) evt,  
 [Object](http://docs.google.com/java/lang/Object.html) what)

**Deprecated.** *As of JDK version 1.1, replaced by processFocusEvent(FocusEvent).*

### isFocusTraversable

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public boolean **isFocusTraversable**()

**Deprecated.** *As of 1.4, replaced by isFocusable().*

Returns whether this Component can become the focus owner.

**Returns:**true if this Component is focusable; false otherwise**Since:** JDK1.1 **See Also:**[setFocusable(boolean)](http://docs.google.com/java/awt/Component.html#setFocusable(boolean))

### isFocusable

public boolean **isFocusable**()

Returns whether this Component can be focused.

**Returns:**true if this Component is focusable; false otherwise.**Since:** 1.4 **See Also:**[setFocusable(boolean)](http://docs.google.com/java/awt/Component.html#setFocusable(boolean))

### setFocusable

public void **setFocusable**(boolean focusable)

Sets the focusable state of this Component to the specified value. This value overrides the Component's default focusability.

**Parameters:**focusable - indicates whether this Component is focusable**Since:** 1.4 **See Also:**[isFocusable()](http://docs.google.com/java/awt/Component.html#isFocusable())

### setFocusTraversalKeys

public void **setFocusTraversalKeys**(int id,  
 [Set](http://docs.google.com/java/util/Set.html)<? extends [AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html)> keystrokes)

Sets the focus traversal keys for a given traversal operation for this Component.

The default values for a Component's focus traversal keys are implementation-dependent. Sun recommends that all implementations for a particular native platform use the same default values. The recommendations for Windows and Unix are listed below. These recommendations are used in the Sun AWT implementations.

| Identifier | Meaning | Default |
| --- | --- | --- |
| KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS | Normal forward keyboard traversal | TAB on KEY\_PRESSED, CTRL-TAB on KEY\_PRESSED |
| KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS | Normal reverse keyboard traversal | SHIFT-TAB on KEY\_PRESSED, CTRL-SHIFT-TAB on KEY\_PRESSED |
| KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYS | Go up one focus traversal cycle | none |

To disable a traversal key, use an empty Set; Collections.EMPTY\_SET is recommended.

Using the AWTKeyStroke API, client code can specify on which of two specific KeyEvents, KEY\_PRESSED or KEY\_RELEASED, the focus traversal operation will occur. Regardless of which KeyEvent is specified, however, all KeyEvents related to the focus traversal key, including the associated KEY\_TYPED event, will be consumed, and will not be dispatched to any Component. It is a runtime error to specify a KEY\_TYPED event as mapping to a focus traversal operation, or to map the same event to multiple default focus traversal operations.

If a value of null is specified for the Set, this Component inherits the Set from its parent. If all ancestors of this Component have null specified for the Set, then the current KeyboardFocusManager's default Set is used.

**Parameters:**id - one of KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS, or KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYSkeystrokes - the Set of AWTKeyStroke for the specified operation **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if id is not one of KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS, or KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYS, or if keystrokes contains null, or if any Object in keystrokes is not an AWTKeyStroke, or if any keystroke represents a KEY\_TYPED event, or if any keystroke already maps to another focus traversal operation for this Component**Since:** 1.4 **See Also:**[getFocusTraversalKeys(int)](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeys(int)), [KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS](http://docs.google.com/java/awt/KeyboardFocusManager.html#FORWARD_TRAVERSAL_KEYS), [KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS](http://docs.google.com/java/awt/KeyboardFocusManager.html#BACKWARD_TRAVERSAL_KEYS), [KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYS](http://docs.google.com/java/awt/KeyboardFocusManager.html#UP_CYCLE_TRAVERSAL_KEYS)

### getFocusTraversalKeys

public [Set](http://docs.google.com/java/util/Set.html)<[AWTKeyStroke](http://docs.google.com/java/awt/AWTKeyStroke.html)> **getFocusTraversalKeys**(int id)

Returns the Set of focus traversal keys for a given traversal operation for this Component. (See setFocusTraversalKeys for a full description of each key.)

If a Set of traversal keys has not been explicitly defined for this Component, then this Component's parent's Set is returned. If no Set has been explicitly defined for any of this Component's ancestors, then the current KeyboardFocusManager's default Set is returned.

**Parameters:**id - one of KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS, or KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYS **Returns:**the Set of AWTKeyStrokes for the specified operation. The Set will be unmodifiable, and may be empty. null will never be returned. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if id is not one of KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS, or KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYS**Since:** 1.4 **See Also:**[setFocusTraversalKeys(int, java.util.Set)](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeys(int,%20java.util.Set)), [KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS](http://docs.google.com/java/awt/KeyboardFocusManager.html#FORWARD_TRAVERSAL_KEYS), [KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS](http://docs.google.com/java/awt/KeyboardFocusManager.html#BACKWARD_TRAVERSAL_KEYS), [KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYS](http://docs.google.com/java/awt/KeyboardFocusManager.html#UP_CYCLE_TRAVERSAL_KEYS)

### areFocusTraversalKeysSet

public boolean **areFocusTraversalKeysSet**(int id)

Returns whether the Set of focus traversal keys for the given focus traversal operation has been explicitly defined for this Component. If this method returns false, this Component is inheriting the Set from an ancestor, or from the current KeyboardFocusManager.

**Parameters:**id - one of KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS, or KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYS **Returns:**true if the the Set of focus traversal keys for the given focus traversal operation has been explicitly defined for this Component; false otherwise. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if id is not one of KeyboardFocusManager.FORWARD\_TRAVERSAL\_KEYS, KeyboardFocusManager.BACKWARD\_TRAVERSAL\_KEYS, or KeyboardFocusManager.UP\_CYCLE\_TRAVERSAL\_KEYS**Since:** 1.4

### setFocusTraversalKeysEnabled

public void **setFocusTraversalKeysEnabled**(boolean focusTraversalKeysEnabled)

Sets whether focus traversal keys are enabled for this Component. Components for which focus traversal keys are disabled receive key events for focus traversal keys. Components for which focus traversal keys are enabled do not see these events; instead, the events are automatically converted to traversal operations.

**Parameters:**focusTraversalKeysEnabled - whether focus traversal keys are enabled for this Component**Since:** 1.4 **See Also:**[getFocusTraversalKeysEnabled()](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeysEnabled()), [setFocusTraversalKeys(int, java.util.Set)](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeys(int,%20java.util.Set)), [getFocusTraversalKeys(int)](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeys(int))

### getFocusTraversalKeysEnabled

public boolean **getFocusTraversalKeysEnabled**()

Returns whether focus traversal keys are enabled for this Component. Components for which focus traversal keys are disabled receive key events for focus traversal keys. Components for which focus traversal keys are enabled do not see these events; instead, the events are automatically converted to traversal operations.

**Returns:**whether focus traversal keys are enabled for this Component**Since:** 1.4 **See Also:**[setFocusTraversalKeysEnabled(boolean)](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeysEnabled(boolean)), [setFocusTraversalKeys(int, java.util.Set)](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeys(int,%20java.util.Set)), [getFocusTraversalKeys(int)](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeys(int))

### requestFocus

public void **requestFocus**()

Requests that this Component get the input focus, and that this Component's top-level ancestor become the focused Window. This component must be displayable, focusable, visible and all of its ancestors (with the exception of the top-level Window) must be visible for the request to be granted. Every effort will be made to honor the request; however, in some cases it may be impossible to do so. Developers must never assume that this Component is the focus owner until this Component receives a FOCUS\_GAINED event. If this request is denied because this Component's top-level Window cannot become the focused Window, the request will be remembered and will be granted when the Window is later focused by the user.

This method cannot be used to set the focus owner to no Component at all. Use KeyboardFocusManager.clearGlobalFocusOwner() instead.

Because the focus behavior of this method is platform-dependent, developers are strongly encouraged to use requestFocusInWindow when possible.

Note: Not all focus transfers result from invoking this method. As such, a component may receive focus without this or any of the other requestFocus methods of Component being invoked.

**Since:** JDK1.0 **See Also:**[requestFocusInWindow()](http://docs.google.com/java/awt/Component.html#requestFocusInWindow()), [FocusEvent](http://docs.google.com/java/awt/event/FocusEvent.html), [addFocusListener(java.awt.event.FocusListener)](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [isFocusable()](http://docs.google.com/java/awt/Component.html#isFocusable()), [isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable()), [KeyboardFocusManager.clearGlobalFocusOwner()](http://docs.google.com/java/awt/KeyboardFocusManager.html#clearGlobalFocusOwner())

### requestFocus

protected boolean **requestFocus**(boolean temporary)

Requests that this Component get the input focus, and that this Component's top-level ancestor become the focused Window. This component must be displayable, focusable, visible and all of its ancestors (with the exception of the top-level Window) must be visible for the request to be granted. Every effort will be made to honor the request; however, in some cases it may be impossible to do so. Developers must never assume that this component is the focus owner until this component receives a FOCUS\_GAINED event. If this request is denied because this component's top-level window cannot become the focused window, the request will be remembered and will be granted when the window is later focused by the user.

This method returns a boolean value. If false is returned, the request is **guaranteed to fail**. If true is returned, the request will succeed **unless** it is vetoed, or an extraordinary event, such as disposal of the component's peer, occurs before the request can be granted by the native windowing system. Again, while a return value of true indicates that the request is likely to succeed, developers must never assume that this component is the focus owner until this component receives a FOCUS\_GAINED event.

This method cannot be used to set the focus owner to no component at all. Use KeyboardFocusManager.clearGlobalFocusOwner instead.

Because the focus behavior of this method is platform-dependent, developers are strongly encouraged to use requestFocusInWindow when possible.

Every effort will be made to ensure that FocusEvents generated as a result of this request will have the specified temporary value. However, because specifying an arbitrary temporary state may not be implementable on all native windowing systems, correct behavior for this method can be guaranteed only for lightweight Components. This method is not intended for general use, but exists instead as a hook for lightweight component libraries, such as Swing.

Note: Not all focus transfers result from invoking this method. As such, a component may receive focus without this or any of the other requestFocus methods of Component being invoked.

**Parameters:**temporary - true if the focus change is temporary, such as when the window loses the focus; for more information on temporary focus changes see the [Focus Specification](http://docs.google.com/java/awt/doc-files/FocusSpec.html) **Returns:**false if the focus change request is guaranteed to fail; true if it is likely to succeed**Since:** 1.4 **See Also:**[FocusEvent](http://docs.google.com/java/awt/event/FocusEvent.html), [addFocusListener(java.awt.event.FocusListener)](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [isFocusable()](http://docs.google.com/java/awt/Component.html#isFocusable()), [isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable()), [KeyboardFocusManager.clearGlobalFocusOwner()](http://docs.google.com/java/awt/KeyboardFocusManager.html#clearGlobalFocusOwner())

### requestFocusInWindow

public boolean **requestFocusInWindow**()

Requests that this Component get the input focus, if this Component's top-level ancestor is already the focused Window. This component must be displayable, focusable, visible and all of its ancestors (with the exception of the top-level Window) must be visible for the request to be granted. Every effort will be made to honor the request; however, in some cases it may be impossible to do so. Developers must never assume that this Component is the focus owner until this Component receives a FOCUS\_GAINED event.

This method returns a boolean value. If false is returned, the request is **guaranteed to fail**. If true is returned, the request will succeed **unless** it is vetoed, or an extraordinary event, such as disposal of the Component's peer, occurs before the request can be granted by the native windowing system. Again, while a return value of true indicates that the request is likely to succeed, developers must never assume that this Component is the focus owner until this Component receives a FOCUS\_GAINED event.

This method cannot be used to set the focus owner to no Component at all. Use KeyboardFocusManager.clearGlobalFocusOwner() instead.

The focus behavior of this method can be implemented uniformly across platforms, and thus developers are strongly encouraged to use this method over requestFocus when possible. Code which relies on requestFocus may exhibit different focus behavior on different platforms.

Note: Not all focus transfers result from invoking this method. As such, a component may receive focus without this or any of the other requestFocus methods of Component being invoked.

**Returns:**false if the focus change request is guaranteed to fail; true if it is likely to succeed**Since:** 1.4 **See Also:**[requestFocus()](http://docs.google.com/java/awt/Component.html#requestFocus()), [FocusEvent](http://docs.google.com/java/awt/event/FocusEvent.html), [addFocusListener(java.awt.event.FocusListener)](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [isFocusable()](http://docs.google.com/java/awt/Component.html#isFocusable()), [isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable()), [KeyboardFocusManager.clearGlobalFocusOwner()](http://docs.google.com/java/awt/KeyboardFocusManager.html#clearGlobalFocusOwner())

### requestFocusInWindow

protected boolean **requestFocusInWindow**(boolean temporary)

Requests that this Component get the input focus, if this Component's top-level ancestor is already the focused Window. This component must be displayable, focusable, visible and all of its ancestors (with the exception of the top-level Window) must be visible for the request to be granted. Every effort will be made to honor the request; however, in some cases it may be impossible to do so. Developers must never assume that this component is the focus owner until this component receives a FOCUS\_GAINED event.

This method returns a boolean value. If false is returned, the request is **guaranteed to fail**. If true is returned, the request will succeed **unless** it is vetoed, or an extraordinary event, such as disposal of the component's peer, occurs before the request can be granted by the native windowing system. Again, while a return value of true indicates that the request is likely to succeed, developers must never assume that this component is the focus owner until this component receives a FOCUS\_GAINED event.

This method cannot be used to set the focus owner to no component at all. Use KeyboardFocusManager.clearGlobalFocusOwner instead.

The focus behavior of this method can be implemented uniformly across platforms, and thus developers are strongly encouraged to use this method over requestFocus when possible. Code which relies on requestFocus may exhibit different focus behavior on different platforms.

Every effort will be made to ensure that FocusEvents generated as a result of this request will have the specified temporary value. However, because specifying an arbitrary temporary state may not be implementable on all native windowing systems, correct behavior for this method can be guaranteed only for lightweight components. This method is not intended for general use, but exists instead as a hook for lightweight component libraries, such as Swing.

Note: Not all focus transfers result from invoking this method. As such, a component may receive focus without this or any of the other requestFocus methods of Component being invoked.

**Parameters:**temporary - true if the focus change is temporary, such as when the window loses the focus; for more information on temporary focus changes see the [Focus Specification](http://docs.google.com/java/awt/doc-files/FocusSpec.html) **Returns:**false if the focus change request is guaranteed to fail; true if it is likely to succeed**Since:** 1.4 **See Also:**[requestFocus()](http://docs.google.com/java/awt/Component.html#requestFocus()), [FocusEvent](http://docs.google.com/java/awt/event/FocusEvent.html), [addFocusListener(java.awt.event.FocusListener)](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [isFocusable()](http://docs.google.com/java/awt/Component.html#isFocusable()), [isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable()), [KeyboardFocusManager.clearGlobalFocusOwner()](http://docs.google.com/java/awt/KeyboardFocusManager.html#clearGlobalFocusOwner())

### transferFocus

public void **transferFocus**()

Transfers the focus to the next component, as though this Component were the focus owner.

**Since:** JDK1.1 **See Also:**[requestFocus()](http://docs.google.com/java/awt/Component.html#requestFocus())

### getFocusCycleRootAncestor

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

Returns the Container which is the focus cycle root of this Component's focus traversal cycle. Each focus traversal cycle has only a single focus cycle root and each Component which is not a Container belongs to only a single focus traversal cycle. Containers which are focus cycle roots belong to two cycles: one rooted at the Container itself, and one rooted at the Container's nearest focus-cycle-root ancestor. For such Containers, this method will return the Container's nearest focus-cycle- root ancestor.

**Returns:**this Component's nearest focus-cycle-root ancestor**Since:** 1.4 **See Also:**[Container.isFocusCycleRoot()](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot())

### isFocusCycleRoot

public boolean **isFocusCycleRoot**([Container](http://docs.google.com/java/awt/Container.html) container)

Returns whether the specified Container is the focus cycle root of this Component's focus traversal cycle. Each focus traversal cycle has only a single focus cycle root and each Component which is not a Container belongs to only a single focus traversal cycle.

**Parameters:**container - the Container to be tested **Returns:**true if the specified Container is a focus-cycle- root of this Component; false otherwise**Since:** 1.4 **See Also:**[Container.isFocusCycleRoot()](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot())

### nextFocus

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **nextFocus**()

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

### transferFocusBackward

public void **transferFocusBackward**()

Transfers the focus to the previous component, as though this Component were the focus owner.

**Since:** 1.4 **See Also:**[requestFocus()](http://docs.google.com/java/awt/Component.html#requestFocus())

### transferFocusUpCycle

public void **transferFocusUpCycle**()

Transfers the focus up one focus traversal cycle. Typically, the focus owner is set to this Component's focus cycle root, and the current focus cycle root is set to the new focus owner's focus cycle root. If, however, this Component's focus cycle root is a Window, then the focus owner is set to the focus cycle root's default Component to focus, and the current focus cycle root is unchanged.

**Since:** 1.4 **See Also:**[requestFocus()](http://docs.google.com/java/awt/Component.html#requestFocus()), [Container.isFocusCycleRoot()](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot()), [Container.setFocusCycleRoot(boolean)](http://docs.google.com/java/awt/Container.html#setFocusCycleRoot(boolean))

### hasFocus

public boolean **hasFocus**()

Returns true if this Component is the focus owner. This method is obsolete, and has been replaced by isFocusOwner().

**Returns:**true if this Component is the focus owner; false otherwise**Since:** 1.2

### isFocusOwner

public boolean **isFocusOwner**()

Returns true if this Component is the focus owner.

**Returns:**true if this Component is the focus owner; false otherwise**Since:** 1.4

### add

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

Adds the specified popup menu to the component.

**Parameters:**popup - the popup menu to be added to the component. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if popup is null**Since:** JDK1.1 **See Also:**[remove(MenuComponent)](http://docs.google.com/java/awt/Component.html#remove(java.awt.MenuComponent))

### remove

public void **remove**([MenuComponent](http://docs.google.com/java/awt/MenuComponent.html) popup)

Removes the specified popup menu from the component.

**Specified by:**[remove](http://docs.google.com/java/awt/MenuContainer.html#remove(java.awt.MenuComponent)) in interface [MenuContainer](http://docs.google.com/java/awt/MenuContainer.html) **Parameters:**popup - the popup menu to be removed**Since:** JDK1.1 **See Also:**[add(PopupMenu)](http://docs.google.com/java/awt/Component.html#add(java.awt.PopupMenu))

### paramString

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

Returns a string representing the state of this component. This method is intended to be used only for debugging purposes, and the content and format of the returned string may vary between implementations. The returned string may be empty but may not be null.

**Returns:**a string representation of this component's state**Since:** JDK1.0

### toString

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

Returns a string representation of this component 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 representation of this component**Since:** JDK1.0

### list

public void **list**()

Prints a listing of this component to the standard system output stream System.out.

**Since:** JDK1.0 **See Also:**[System.out](http://docs.google.com/java/lang/System.html#out)

### list

public void **list**([PrintStream](http://docs.google.com/java/io/PrintStream.html) out)

Prints a listing of this component to the specified output stream.

**Parameters:**out - a print stream**Since:** JDK1.0

### list

public void **list**([PrintStream](http://docs.google.com/java/io/PrintStream.html) out,  
 int indent)

Prints out a list, starting at the specified indentation, to the specified print stream.

**Parameters:**out - a print streamindent - number of spaces to indent**Since:** JDK1.0 **See Also:**[PrintStream.println(java.lang.Object)](http://docs.google.com/java/io/PrintStream.html#println(java.lang.Object))

### list

public void **list**([PrintWriter](http://docs.google.com/java/io/PrintWriter.html) out)

Prints a listing to the specified print writer.

**Parameters:**out - the print writer to print to**Since:** JDK1.1

### list

public void **list**([PrintWriter](http://docs.google.com/java/io/PrintWriter.html) out,  
 int indent)

Prints out a list, starting at the specified indentation, to the specified print writer.

**Parameters:**out - the print writer to print toindent - the number of spaces to indent**Since:** JDK1.1 **See Also:**[PrintStream.println(java.lang.Object)](http://docs.google.com/java/io/PrintStream.html#println(java.lang.Object))

### addPropertyChangeListener

public void **addPropertyChangeListener**([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)

Adds a PropertyChangeListener to the listener list. The listener is registered for all bound properties of this class, including the following:

* this Component's font ("font")
* this Component's background color ("background")
* this Component's foreground color ("foreground")
* this Component's focusability ("focusable")
* this Component's focus traversal keys enabled state ("focusTraversalKeysEnabled")
* this Component's Set of FORWARD\_TRAVERSAL\_KEYS ("forwardFocusTraversalKeys")
* this Component's Set of BACKWARD\_TRAVERSAL\_KEYS ("backwardFocusTraversalKeys")
* this Component's Set of UP\_CYCLE\_TRAVERSAL\_KEYS ("upCycleFocusTraversalKeys")
* this Component's preferred size ("preferredSize")
* this Component's minimum size ("minimumSize")
* this Component's maximum size ("maximumSize")
* this Component's name ("name")

Note that if this Component is inheriting a bound property, then no event will be fired in response to a change in the inherited property.

If listener is null, no exception is thrown and no action is performed.

**Parameters:**listener - the property change listener to be added**See Also:**[removePropertyChangeListener(java.beans.PropertyChangeListener)](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.beans.PropertyChangeListener)), [getPropertyChangeListeners()](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners()), [addPropertyChangeListener(java.lang.String, java.beans.PropertyChangeListener)](http://docs.google.com/java/awt/Component.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener))

### removePropertyChangeListener

public void **removePropertyChangeListener**([PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)

Removes a PropertyChangeListener from the listener list. This method should be used to remove PropertyChangeListeners that were registered for all bound properties of this class.

If listener is null, no exception is thrown and no action is performed.

**Parameters:**listener - the PropertyChangeListener to be removed**See Also:**[addPropertyChangeListener(java.beans.PropertyChangeListener)](http://docs.google.com/java/awt/Component.html#addPropertyChangeListener(java.beans.PropertyChangeListener)), [getPropertyChangeListeners()](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners()), [removePropertyChangeListener(java.lang.String,java.beans.PropertyChangeListener)](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener))

### getPropertyChangeListeners

public [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html)[] **getPropertyChangeListeners**()

Returns an array of all the property change listeners registered on this component.

**Returns:**all of this component's PropertyChangeListeners or an empty array if no property change listeners are currently registered**Since:** 1.4 **See Also:**[addPropertyChangeListener(java.beans.PropertyChangeListener)](http://docs.google.com/java/awt/Component.html#addPropertyChangeListener(java.beans.PropertyChangeListener)), [removePropertyChangeListener(java.beans.PropertyChangeListener)](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.beans.PropertyChangeListener)), [getPropertyChangeListeners(java.lang.String)](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners(java.lang.String)), [PropertyChangeSupport.getPropertyChangeListeners()](http://docs.google.com/java/beans/PropertyChangeSupport.html#getPropertyChangeListeners())

### addPropertyChangeListener

public void **addPropertyChangeListener**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)

Adds a PropertyChangeListener to the listener list for a specific property. The specified property may be user-defined, or one of the following:

* this Component's font ("font")
* this Component's background color ("background")
* this Component's foreground color ("foreground")
* this Component's focusability ("focusable")
* this Component's focus traversal keys enabled state ("focusTraversalKeysEnabled")
* this Component's Set of FORWARD\_TRAVERSAL\_KEYS ("forwardFocusTraversalKeys")
* this Component's Set of BACKWARD\_TRAVERSAL\_KEYS ("backwardFocusTraversalKeys")
* this Component's Set of UP\_CYCLE\_TRAVERSAL\_KEYS ("upCycleFocusTraversalKeys")

Note that if this Component is inheriting a bound property, then no event will be fired in response to a change in the inherited property.

If propertyName or listener is null, no exception is thrown and no action is taken.

**Parameters:**propertyName - one of the property names listed abovelistener - the property change listener to be added**See Also:**[removePropertyChangeListener(java.lang.String, java.beans.PropertyChangeListener)](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [getPropertyChangeListeners(java.lang.String)](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners(java.lang.String)), [addPropertyChangeListener(java.lang.String, java.beans.PropertyChangeListener)](http://docs.google.com/java/awt/Component.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener))

### removePropertyChangeListener

public void **removePropertyChangeListener**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html) listener)

Removes a PropertyChangeListener from the listener list for a specific property. This method should be used to remove PropertyChangeListeners that were registered for a specific bound property.

If propertyName or listener is null, no exception is thrown and no action is taken.

**Parameters:**propertyName - a valid property namelistener - the PropertyChangeListener to be removed**See Also:**[addPropertyChangeListener(java.lang.String, java.beans.PropertyChangeListener)](http://docs.google.com/java/awt/Component.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [getPropertyChangeListeners(java.lang.String)](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners(java.lang.String)), [removePropertyChangeListener(java.beans.PropertyChangeListener)](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.beans.PropertyChangeListener))

### getPropertyChangeListeners

public [PropertyChangeListener](http://docs.google.com/java/beans/PropertyChangeListener.html)[] **getPropertyChangeListeners**([String](http://docs.google.com/java/lang/String.html) propertyName)

Returns an array of all the listeners which have been associated with the named property.

**Returns:**all of the PropertyChangeListeners associated with the named property; if no such listeners have been added or if propertyName is null, an empty array is returned**Since:** 1.4 **See Also:**[addPropertyChangeListener(java.lang.String, java.beans.PropertyChangeListener)](http://docs.google.com/java/awt/Component.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [removePropertyChangeListener(java.lang.String, java.beans.PropertyChangeListener)](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [getPropertyChangeListeners()](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners())

### firePropertyChange

protected void **firePropertyChange**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 [Object](http://docs.google.com/java/lang/Object.html) oldValue,  
 [Object](http://docs.google.com/java/lang/Object.html) newValue)

Support for reporting bound property changes for Object properties. This method can be called when a bound property has changed and it will send the appropriate PropertyChangeEvent to any registered PropertyChangeListeners.

**Parameters:**propertyName - the property whose value has changedoldValue - the property's previous valuenewValue - the property's new value

### firePropertyChange

protected void **firePropertyChange**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 boolean oldValue,  
 boolean newValue)

Support for reporting bound property changes for boolean properties. This method can be called when a bound property has changed and it will send the appropriate PropertyChangeEvent to any registered PropertyChangeListeners.

**Parameters:**propertyName - the property whose value has changedoldValue - the property's previous valuenewValue - the property's new value**Since:** 1.4

### firePropertyChange

protected void **firePropertyChange**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 int oldValue,  
 int newValue)

Support for reporting bound property changes for integer properties. This method can be called when a bound property has changed and it will send the appropriate PropertyChangeEvent to any registered PropertyChangeListeners.

**Parameters:**propertyName - the property whose value has changedoldValue - the property's previous valuenewValue - the property's new value**Since:** 1.4

### firePropertyChange

public void **firePropertyChange**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 byte oldValue,  
 byte newValue)

Reports a bound property change.

**Parameters:**propertyName - the programmatic name of the property that was changedoldValue - the old value of the property (as a byte)newValue - the new value of the property (as a byte)**Since:** 1.5 **See Also:**[firePropertyChange(java.lang.String, java.lang.Object, java.lang.Object)](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object))

### firePropertyChange

public void **firePropertyChange**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 char oldValue,  
 char newValue)

Reports a bound property change.

**Parameters:**propertyName - the programmatic name of the property that was changedoldValue - the old value of the property (as a char)newValue - the new value of the property (as a char)**Since:** 1.5 **See Also:**[firePropertyChange(java.lang.String, java.lang.Object, java.lang.Object)](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object))

### firePropertyChange

public void **firePropertyChange**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 short oldValue,  
 short newValue)

Reports a bound property change.

**Parameters:**propertyName - the programmatic name of the property that was changedoldValue - the old value of the property (as a short)newValue - the old value of the property (as a short)**Since:** 1.5 **See Also:**[firePropertyChange(java.lang.String, java.lang.Object, java.lang.Object)](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object))

### firePropertyChange

public void **firePropertyChange**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 long oldValue,  
 long newValue)

Reports a bound property change.

**Parameters:**propertyName - the programmatic name of the property that was changedoldValue - the old value of the property (as a long)newValue - the new value of the property (as a long)**Since:** 1.5 **See Also:**[firePropertyChange(java.lang.String, java.lang.Object, java.lang.Object)](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object))

### firePropertyChange

public void **firePropertyChange**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 float oldValue,  
 float newValue)

Reports a bound property change.

**Parameters:**propertyName - the programmatic name of the property that was changedoldValue - the old value of the property (as a float)newValue - the new value of the property (as a float)**Since:** 1.5 **See Also:**[firePropertyChange(java.lang.String, java.lang.Object, java.lang.Object)](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object))

### firePropertyChange

public void **firePropertyChange**([String](http://docs.google.com/java/lang/String.html) propertyName,  
 double oldValue,  
 double newValue)

Reports a bound property change.

**Parameters:**propertyName - the programmatic name of the property that was changedoldValue - the old value of the property (as a double)newValue - the new value of the property (as a double)**Since:** 1.5 **See Also:**[firePropertyChange(java.lang.String, java.lang.Object, java.lang.Object)](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object))

### setComponentOrientation

public void **setComponentOrientation**([ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html) o)

Sets the language-sensitive orientation that is to be used to order the elements or text within this component. Language-sensitive LayoutManager and Component subclasses will use this property to determine how to lay out and draw components.

At construction time, a component's orientation is set to ComponentOrientation.UNKNOWN, indicating that it has not been specified explicitly. The UNKNOWN orientation behaves the same as ComponentOrientation.LEFT\_TO\_RIGHT.

To set the orientation of a single component, use this method. To set the orientation of an entire component hierarchy, use [applyComponentOrientation](http://docs.google.com/java/awt/Component.html#applyComponentOrientation(java.awt.ComponentOrientation)).

**See Also:**[ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html)

### getComponentOrientation

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

Retrieves the language-sensitive orientation that is to be used to order the elements or text within this component. LayoutManager and Component subclasses that wish to respect orientation should call this method to get the component's orientation before performing layout or drawing.

**See Also:**[ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html)

### applyComponentOrientation

public void **applyComponentOrientation**([ComponentOrientation](http://docs.google.com/java/awt/ComponentOrientation.html) orientation)

Sets the ComponentOrientation property of this component and all components contained within it.

**Parameters:**orientation - the new component orientation of this component and the components contained within it. **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if orientation is null.**Since:** 1.4 **See Also:**[setComponentOrientation(java.awt.ComponentOrientation)](http://docs.google.com/java/awt/Component.html#setComponentOrientation(java.awt.ComponentOrientation)), [getComponentOrientation()](http://docs.google.com/java/awt/Component.html#getComponentOrientation())

### getAccessibleContext

public [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) **getAccessibleContext**()

Gets the AccessibleContext associated with this Component. The method implemented by this base class returns null. Classes that extend Component should implement this method to return the AccessibleContext associated with the subclass.

**Returns:**the AccessibleContext of this Component**Since:** 1.3

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Component.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/Color.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Component.AccessibleAWTComponent.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Component.html)    [**NO FRAMES**](http://docs.google.com/Component.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#2et92p0) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#2s8eyo1) | [CONSTR](#1ksv4uv) | [METHOD](#2jxsxqh) |

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

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

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