java.awt

Class Component

```
java.lang.Object
  |
  +--java.awt.Component
```

All Implemented Interfaces:
>   ImageObserver, MenuContainer, Serializable

Direct Known Subclasses:
>   Box.Filler, Button, Canvas, Checkbox, Choice, Container, Label, List, Scrollbar, TextComponent

---

public abstract class **Component**
extends Object
implements ImageObserver, MenuContainer, Serializable

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.

See Also:
>   Serialized Form

Inner Class Summary

protected class	**Component.AccessibleAWTComponent** Inner class of Component used to provide default support for accessibility.

Field Summary

static float	**BOTTOM_ALIGNMENT** Ease-of-use constant for getAlignmentY.
static float	**CENTER_ALIGNMENT** Ease-of-use constant for getAlignmentY and getAlignmentX.
static float	**LEFT_ALIGNMENT** Ease-of-use constant for getAlignmentX.
static float	**RIGHT_ALIGNMENT** Ease-of-use constant for getAlignmentX.
static float	**TOP_ALIGNMENT** Ease-of-use constant for getAlignmentY().

2

Fields inherited from interface java.awt.image.ImageObserver	
ABORT, ALLBITS, ERROR, FRAMEBITS, HEIGHT, PROPERTIES, SOMEBITS, WIDTH	

Constructor Summary

protected	**Component**() Constructs a new component.

Method Summary

boolean	**action**(Event evt, Object what) **Deprecated.** *As of JDK version 1.1, should register this component as ActionListener on component which fires action events.*
void	**add**(PopupMenu popup) Adds the specified popup menu to the component.
void	**addComponentListener**(ComponentListener l) Adds the specified component listener to receive component events from this component.
void	**addFocusListener**(FocusListener l) Adds the specified focus listener to receive focus events from this component when this component gains input focus.
void	**addHierarchyBoundsListener**(HierarchyBoundsListener 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**(HierarchyListener 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**(InputMethodListener l) Adds the specified input method listener to receive input method events from this component.
void	**addKeyListener**(KeyListener l) Adds the specified key listener to receive key events from this component.
void	**addMouseListener**(MouseListener l) Adds the specified mouse listener to receive mouse events from this component.
void	**addMouseMotionListener**(MouseMotionListener l) Adds the specified mouse motion listener to receive mouse motion events from this component.
void	**addNotify**() Makes this Component displayable by connecting it to a native screen resource.
void	**addPropertyChangeListener**(PropertyChangeListener listener) Add a PropertyChangeListener to the listener list.
void	**addPropertyChangeListener**(String propertyName, PropertyChangeListener listener) Add a PropertyChangeListener for a specific property.
Rectangle	

	bounds() **Deprecated.** *As of JDK version 1.1, replaced by getBounds().*
int	**checkImage**(Image image, ImageObserver observer) Returns the status of the construction of a screen representation of the specified image.
int	**checkImage**(Image image, int width, int height, ImageObserver observer) Returns the status of the construction of a screen representation of the specified image.
protected AWTEvent	**coalesceEvents**(AWTEvent existingEvent, AWTEvent newEvent) Potentially coalesce an event being posted with an existing event.
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.
boolean	**contains**(Point 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	**createImage**(ImageProducer producer) Creates an image from the specified image producer.
Image	**createImage**(int width, int height) Creates an off-screen drawable image to be used for double buffering.
void	**deliverEvent**(Event e) **Deprecated.** *As of JDK version 1.1, replaced by dispatchEvent(AWTEvent e).*
void	**disable**() **Deprecated.** *As of JDK version 1.1, replaced by setEnabled(boolean).*
protected void	**disableEvents**(long eventsToDisable) Disables the events defined by the specified event mask parameter from being delivered to this component.
void	**dispatchEvent**(AWTEvent e) Dispatches an event to this component or one of its sub components.
void	**doLayout**() Prompts the layout manager to lay out this component.
void	**enable**() **Deprecated.** *As of JDK version 1.1, replaced by setEnabled(boolean).*
void	**enable**(boolean b) **Deprecated.** *As of JDK version 1.1, replaced by setEnabled(boolean).*
protected void	**enableEvents**(long eventsToEnable) Enables the events defined by the specified event mask parameter to be delivered to this component.
void	**enableInputMethods**(boolean enable) Enables or disables input method support for this component.
protected void	**firePropertyChange**(String propertyName, Object oldValue, Object newValue) Support for reporting bound property changes.
AccessibleContext	**getAccessibleContext**() Get the AccessibleContext associated with this Component
float	

	getAlignmentX() Returns the alignment along the x axis.
float	**getAlignmentY**() Returns the alignment along the y axis.
Color	**getBackground**() Gets the background color of this component.
Rectangle	**getBounds**() Gets the bounds of this component in the form of a Rectangle object.
Rectangle	**getBounds**(Rectangle rv) Store the bounds of this component into "return value" **rv** and return **rv**.
ColorModel	**getColorModel**() Gets the instance of ColorModel used to display the component on the output device.
Component	**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.
Component	**getComponentAt**(Point p) Returns the component or subcomponent that contains the specified point.
ComponentOrientation	**getComponentOrientation**() Retrieve the language-sensitive orientation that is to be used to order the elements or text within this component.
Cursor	**getCursor**() Gets the cursor set in the component.
DropTarget	**getDropTarget**() Get the DropTarget associated with this Component
Font	**getFont**() Gets the font of this component.
FontMetrics	**getFontMetrics**(Font font) Gets the font metrics for the specified font.
Color	**getForeground**() Gets the foreground color of this component.
Graphics	**getGraphics**() Creates a graphics context for this component.
GraphicsConfiguration	**getGraphicsConfiguration**() Get the GraphicsConfiguration associated with this Component.
int	**getHeight**() Return the current height of this component.
InputContext	**getInputContext**() Gets the input context used by this component for handling the communication with input methods when text is entered in this component.
InputMethodRequests	**getInputMethodRequests**() Gets the input method request handler which supports requests from input methods for this component.
EventListener[]	**getListeners**(Class listenerType) Return an array of all the listeners that were added to the Component with

	addXXXListener(), where XXX is the name of the `listenerType` argument.
Locale	**getLocale**() Gets the locale of this component.
Point	**getLocation**() Gets the location of this component in the form of a point specifying the component's top-left corner.
Point	**getLocation**(Point rv) Store the x,y origin of this component into "return value" **rv** and return **rv**.
Point	**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	**getMaximumSize**() Gets the maximum size of this component.
Dimension	**getMinimumSize**() Gets the mininimum size of this component.
String	**getName**() Gets the name of the component.
Container	**getParent**() Gets the parent of this component.
java.awt.peer.ComponentPeer	**getPeer**() **Deprecated.** *As of JDK version 1.1, programs should not directly manipulate peers. replaced by* `boolean isDisplayable()`.
Dimension	**getPreferredSize**() Gets the preferred size of this component.
Dimension	**getSize**() Returns the size of this component in the form of a `Dimension` object.
Dimension	**getSize**(Dimension rv) Store the width/height of this component into "return value" **rv** and return **rv**.
Toolkit	**getToolkit**() Gets the toolkit of this component.
Object	**getTreeLock**() Gets the locking object for AWT component-tree and layout Gets this component's locking object (the object that owns the thread sychronization monitor) for AWT component-tree and layout operations.
int	**getWidth**() Return the current width of this component.
int	**getX**() Return the current x coordinate of the components origin.
int	**getY**() Return the current y coordinate of the components origin.
boolean	**gotFocus**(Event evt, Object what) **Deprecated.** *As of JDK version 1.1, replaced by processFocusEvent(FocusEvent).*
boolean	**handleEvent**(Event evt) **Deprecated.** *As of JDK version 1.1 replaced by processEvent(AWTEvent).*

DOCKET ALARM

Explore Litigation Insights

Docket Alarm provides insights to develop a more informed litigation strategy and the peace of mind of knowing you're on top of things.

Real-Time Litigation Alerts

Keep your litigation team up-to-date with **real-time alerts** and advanced team management tools built for the enterprise, all while greatly reducing PACER spend.

Our comprehensive service means we can handle Federal, State, and Administrative courts across the country.

Advanced Docket Research

With over 230 million records, Docket Alarm's cloud-native docket research platform finds what other services can't. Coverage includes Federal, State, plus PTAB, TTAB, ITC and NLRB decisions, all in one place.

Identify arguments that have been successful in the past with full text, pinpoint searching. Link to case law cited within any court document via Fastcase.

Analytics At Your Fingertips

Learn what happened the last time a particular judge, opposing counsel or company faced cases similar to yours.

Advanced out-of-the-box PTAB and TTAB analytics are always at your fingertips.

API

Docket Alarm offers a powerful API (application programming interface) to developers that want to integrate case filings into their apps.

LAW FIRMS

Build custom dashboards for your attorneys and clients with live data direct from the court.

Automate many repetitive legal tasks like conflict checks, document management, and marketing.

FINANCIAL INSTITUTIONS

Litigation and bankruptcy checks for companies and debtors.

E-DISCOVERY AND LEGAL VENDORS

Sync your system to PACER to automate legal marketing.

fastcase
Smarter legal research.