| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Frame.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/FontMetrics.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Frame.AccessibleAWTFrame.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Frame.html)    [**NO FRAMES**](http://docs.google.com/Frame.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#3rdcrjn) | DETAIL: [FIELD](#2jxsxqh) | [CONSTR](#41mghml) | [METHOD](#4f1mdlm) |

## **java.awt**

Class Frame

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

**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), [Accessible](http://docs.google.com/javax/accessibility/Accessible.html) **Direct Known Subclasses:** [JFrame](http://docs.google.com/javax/swing/JFrame.html)

public class **Frame**extends [Window](http://docs.google.com/java/awt/Window.html)implements [MenuContainer](http://docs.google.com/java/awt/MenuContainer.html)

A Frame is a top-level window with a title and a border.

The size of the frame includes any area designated for the border. The dimensions of the border area may be obtained using the getInsets method, however, since these dimensions are platform-dependent, a valid insets value cannot be obtained until the frame is made displayable by either calling pack or show. Since the border area is included in the overall size of the frame, the border effectively obscures a portion of the frame, constraining the area available for rendering and/or displaying subcomponents to the rectangle which has an upper-left corner location of (insets.left, insets.top), and has a size of width - (insets.left + insets.right) by height - (insets.top + insets.bottom).

The default layout for a frame is BorderLayout.

A frame may have its native decorations (i.e. Frame and Titlebar) turned off with setUndecorated. This can only be done while the frame is not [displayable](http://docs.google.com/java/awt/Component.html#isDisplayable()).

In a multi-screen environment, you can create a Frame on a different screen device by constructing the Frame with [Frame(GraphicsConfiguration)](http://docs.google.com/java/awt/Frame.html#Frame(java.awt.GraphicsConfiguration)) or [Frame(String title, GraphicsConfiguration)](http://docs.google.com/java/awt/Frame.html#Frame(java.lang.String,%20java.awt.GraphicsConfiguration)). The GraphicsConfiguration object is one of the GraphicsConfiguration objects of the target screen device.

In a virtual device multi-screen environment in which the desktop area could span multiple physical screen devices, the bounds of all configurations are relative to the virtual-coordinate system. The origin of the virtual-coordinate system is at the upper left-hand corner of the primary physical screen. Depending on the location of the primary screen in the virtual device, negative coordinates are possible, as shown in the following figure.



In such an environment, when calling setLocation, you must pass a virtual coordinate to this method. Similarly, calling getLocationOnScreen on a Frame returns virtual device coordinates. Call the getBounds method of a GraphicsConfiguration to find its origin in the virtual coordinate system.

The following code sets the location of the Frame at (10, 10) relative to the origin of the physical screen of the corresponding GraphicsConfiguration. If the bounds of the GraphicsConfiguration is not taken into account, the Frame location would be set at (10, 10) relative to the virtual-coordinate system and would appear on the primary physical screen, which might be different from the physical screen of the specified GraphicsConfiguration.

Frame f = new Frame(GraphicsConfiguration gc);  
 Rectangle bounds = gc.getBounds();  
 f.setLocation(10 + bounds.x, 10 + bounds.y);

Frames are capable of generating the following types of WindowEvents:

* WINDOW\_OPENED
* WINDOW\_CLOSING:  
  If the program doesn't explicitly hide or dispose the window while processing this event, the window close operation is canceled.
* WINDOW\_CLOSED
* WINDOW\_ICONIFIED
* WINDOW\_DEICONIFIED
* WINDOW\_ACTIVATED
* WINDOW\_DEACTIVATED
* WINDOW\_GAINED\_FOCUS
* WINDOW\_LOST\_FOCUS
* WINDOW\_STATE\_CHANGED

**Since:** JDK1.0 **See Also:**[WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html), [Window.addWindowListener(java.awt.event.WindowListener)](http://docs.google.com/java/awt/Window.html#addWindowListener(java.awt.event.WindowListener)), [Serialized Form](http://docs.google.com/serialized-form.html#java.awt.Frame)

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**Frame.AccessibleAWTFrame**](http://docs.google.com/java/awt/Frame.AccessibleAWTFrame.html)            This class implements accessibility support for the Frame class. |

| **Nested classes/interfaces inherited from class java.awt.**[**Window**](http://docs.google.com/java/awt/Window.html) |
| --- |
| [Window.AccessibleAWTWindow](http://docs.google.com/java/awt/Window.AccessibleAWTWindow.html) |

| **Nested classes/interfaces inherited from class java.awt.**[**Container**](http://docs.google.com/java/awt/Container.html) |
| --- |
| [Container.AccessibleAWTContainer](http://docs.google.com/java/awt/Container.AccessibleAWTContainer.html) |

| **Nested classes/interfaces inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [Component.AccessibleAWTComponent](http://docs.google.com/java/awt/Component.AccessibleAWTComponent.html), [Component.BaselineResizeBehavior](http://docs.google.com/java/awt/Component.BaselineResizeBehavior.html), [Component.BltBufferStrategy](http://docs.google.com/java/awt/Component.BltBufferStrategy.html), [Component.FlipBufferStrategy](http://docs.google.com/java/awt/Component.FlipBufferStrategy.html) |

| **Field Summary** | |
| --- | --- |
| static int | [**CROSSHAIR\_CURSOR**](http://docs.google.com/java/awt/Frame.html#CROSSHAIR_CURSOR)  **Deprecated.** *replaced by Cursor.CROSSHAIR\_CURSOR.* |
| static int | [**DEFAULT\_CURSOR**](http://docs.google.com/java/awt/Frame.html#DEFAULT_CURSOR)  **Deprecated.** *replaced by Cursor.DEFAULT\_CURSOR.* |
| static int | [**E\_RESIZE\_CURSOR**](http://docs.google.com/java/awt/Frame.html#E_RESIZE_CURSOR)  **Deprecated.** *replaced by Cursor.E\_RESIZE\_CURSOR.* |
| static int | [**HAND\_CURSOR**](http://docs.google.com/java/awt/Frame.html#HAND_CURSOR)  **Deprecated.** *replaced by Cursor.HAND\_CURSOR.* |
| static int | [**ICONIFIED**](http://docs.google.com/java/awt/Frame.html#ICONIFIED)            This state bit indicates that frame is iconified. |
| static int | [**MAXIMIZED\_BOTH**](http://docs.google.com/java/awt/Frame.html#MAXIMIZED_BOTH)            This state bit mask indicates that frame is fully maximized (that is both horizontally and vertically). |
| static int | [**MAXIMIZED\_HORIZ**](http://docs.google.com/java/awt/Frame.html#MAXIMIZED_HORIZ)            This state bit indicates that frame is maximized in the horizontal direction. |
| static int | [**MAXIMIZED\_VERT**](http://docs.google.com/java/awt/Frame.html#MAXIMIZED_VERT)            This state bit indicates that frame is maximized in the vertical direction. |
| static int | [**MOVE\_CURSOR**](http://docs.google.com/java/awt/Frame.html#MOVE_CURSOR)  **Deprecated.** *replaced by Cursor.MOVE\_CURSOR.* |
| static int | [**N\_RESIZE\_CURSOR**](http://docs.google.com/java/awt/Frame.html#N_RESIZE_CURSOR)  **Deprecated.** *replaced by Cursor.N\_RESIZE\_CURSOR.* |
| static int | [**NE\_RESIZE\_CURSOR**](http://docs.google.com/java/awt/Frame.html#NE_RESIZE_CURSOR)  **Deprecated.** *replaced by Cursor.NE\_RESIZE\_CURSOR.* |
| static int | [**NORMAL**](http://docs.google.com/java/awt/Frame.html#NORMAL)            Frame is in the "normal" state. |
| static int | [**NW\_RESIZE\_CURSOR**](http://docs.google.com/java/awt/Frame.html#NW_RESIZE_CURSOR)  **Deprecated.** *replaced by Cursor.NW\_RESIZE\_CURSOR.* |
| static int | [**S\_RESIZE\_CURSOR**](http://docs.google.com/java/awt/Frame.html#S_RESIZE_CURSOR)  **Deprecated.** *replaced by Cursor.S\_RESIZE\_CURSOR.* |
| static int | [**SE\_RESIZE\_CURSOR**](http://docs.google.com/java/awt/Frame.html#SE_RESIZE_CURSOR)  **Deprecated.** *replaced by Cursor.SE\_RESIZE\_CURSOR.* |
| static int | [**SW\_RESIZE\_CURSOR**](http://docs.google.com/java/awt/Frame.html#SW_RESIZE_CURSOR)  **Deprecated.** *replaced by Cursor.SW\_RESIZE\_CURSOR.* |
| static int | [**TEXT\_CURSOR**](http://docs.google.com/java/awt/Frame.html#TEXT_CURSOR)  **Deprecated.** *replaced by Cursor.TEXT\_CURSOR.* |
| static int | [**W\_RESIZE\_CURSOR**](http://docs.google.com/java/awt/Frame.html#W_RESIZE_CURSOR)  **Deprecated.** *replaced by Cursor.W\_RESIZE\_CURSOR.* |
| static int | [**WAIT\_CURSOR**](http://docs.google.com/java/awt/Frame.html#WAIT_CURSOR)  **Deprecated.** *replaced by Cursor.WAIT\_CURSOR.* |

| **Fields inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [BOTTOM\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#BOTTOM_ALIGNMENT), [CENTER\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#CENTER_ALIGNMENT), [LEFT\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#LEFT_ALIGNMENT), [RIGHT\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#RIGHT_ALIGNMENT), [TOP\_ALIGNMENT](http://docs.google.com/java/awt/Component.html#TOP_ALIGNMENT) |

| **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** | |
| --- | --- |
| [**Frame**](http://docs.google.com/java/awt/Frame.html#Frame())()            Constructs a new instance of Frame that is initially invisible. |
| [**Frame**](http://docs.google.com/java/awt/Frame.html#Frame(java.awt.GraphicsConfiguration))([GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html) gc)            Constructs a new, initially invisible Frame with the specified GraphicsConfiguration. |
| [**Frame**](http://docs.google.com/java/awt/Frame.html#Frame(java.lang.String))([String](http://docs.google.com/java/lang/String.html) title)            Constructs a new, initially invisible Frame object with the specified title. |
| [**Frame**](http://docs.google.com/java/awt/Frame.html#Frame(java.lang.String,%20java.awt.GraphicsConfiguration))([String](http://docs.google.com/java/lang/String.html) title, [GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html) gc)            Constructs a new, initially invisible Frame object with the specified title and a GraphicsConfiguration. |

| **Method Summary** | |
| --- | --- |
| void | [**addNotify**](http://docs.google.com/java/awt/Frame.html#addNotify())()            Makes this Frame displayable by connecting it to a native screen resource. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/java/awt/Frame.html#getAccessibleContext())()            Gets the AccessibleContext associated with this Frame. |
| int | [**getCursorType**](http://docs.google.com/java/awt/Frame.html#getCursorType())()  **Deprecated.** *As of JDK version 1.1, replaced by Component.getCursor().* |
| int | [**getExtendedState**](http://docs.google.com/java/awt/Frame.html#getExtendedState())()            Gets the state of this frame. |
| static [Frame](http://docs.google.com/java/awt/Frame.html)[] | [**getFrames**](http://docs.google.com/java/awt/Frame.html#getFrames())()            Returns an array of all Frames created by this application. |
| [Image](http://docs.google.com/java/awt/Image.html) | [**getIconImage**](http://docs.google.com/java/awt/Frame.html#getIconImage())()            Returns the image to be displayed as the icon for this frame. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getMaximizedBounds**](http://docs.google.com/java/awt/Frame.html#getMaximizedBounds())()            Gets maximized bounds for this frame. |
| [MenuBar](http://docs.google.com/java/awt/MenuBar.html) | [**getMenuBar**](http://docs.google.com/java/awt/Frame.html#getMenuBar())()            Gets the menu bar for this frame. |
| int | [**getState**](http://docs.google.com/java/awt/Frame.html#getState())()            Gets the state of this frame (obsolete). |
| [String](http://docs.google.com/java/lang/String.html) | [**getTitle**](http://docs.google.com/java/awt/Frame.html#getTitle())()            Gets the title of the frame. |
| boolean | [**isResizable**](http://docs.google.com/java/awt/Frame.html#isResizable())()            Indicates whether this frame is resizable by the user. |
| boolean | [**isUndecorated**](http://docs.google.com/java/awt/Frame.html#isUndecorated())()            Indicates whether this frame is undecorated. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/java/awt/Frame.html#paramString())()            Returns a string representing the state of this Frame. |
| void | [**remove**](http://docs.google.com/java/awt/Frame.html#remove(java.awt.MenuComponent))([MenuComponent](http://docs.google.com/java/awt/MenuComponent.html) m)            Removes the specified menu bar from this frame. |
| void | [**removeNotify**](http://docs.google.com/java/awt/Frame.html#removeNotify())()            Makes this Frame undisplayable by removing its connection to its native screen resource. |
| void | [**setCursor**](http://docs.google.com/java/awt/Frame.html#setCursor(int))(int cursorType)  **Deprecated.** *As of JDK version 1.1, replaced by Component.setCursor(Cursor).* |
| void | [**setExtendedState**](http://docs.google.com/java/awt/Frame.html#setExtendedState(int))(int state)            Sets the state of this frame. |
| void | [**setIconImage**](http://docs.google.com/java/awt/Frame.html#setIconImage(java.awt.Image))([Image](http://docs.google.com/java/awt/Image.html) image)            Sets the image to be displayed as the icon for this window. |
| void | [**setMaximizedBounds**](http://docs.google.com/java/awt/Frame.html#setMaximizedBounds(java.awt.Rectangle))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) bounds)            Sets the maximized bounds for this frame. |
| void | [**setMenuBar**](http://docs.google.com/java/awt/Frame.html#setMenuBar(java.awt.MenuBar))([MenuBar](http://docs.google.com/java/awt/MenuBar.html) mb)            Sets the menu bar for this frame to the specified menu bar. |
| void | [**setResizable**](http://docs.google.com/java/awt/Frame.html#setResizable(boolean))(boolean resizable)            Sets whether this frame is resizable by the user. |
| void | [**setState**](http://docs.google.com/java/awt/Frame.html#setState(int))(int state)            Sets the state of this frame (obsolete). |
| void | [**setTitle**](http://docs.google.com/java/awt/Frame.html#setTitle(java.lang.String))([String](http://docs.google.com/java/lang/String.html) title)            Sets the title for this frame to the specified string. |
| void | [**setUndecorated**](http://docs.google.com/java/awt/Frame.html#setUndecorated(boolean))(boolean undecorated)            Disables or enables decorations for this frame. |

| **Methods inherited from class java.awt.**[**Window**](http://docs.google.com/java/awt/Window.html) |
| --- |
| [addPropertyChangeListener](http://docs.google.com/java/awt/Window.html#addPropertyChangeListener(java.beans.PropertyChangeListener)), [addPropertyChangeListener](http://docs.google.com/java/awt/Window.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [addWindowFocusListener](http://docs.google.com/java/awt/Window.html#addWindowFocusListener(java.awt.event.WindowFocusListener)), [addWindowListener](http://docs.google.com/java/awt/Window.html#addWindowListener(java.awt.event.WindowListener)), [addWindowStateListener](http://docs.google.com/java/awt/Window.html#addWindowStateListener(java.awt.event.WindowStateListener)), [applyResourceBundle](http://docs.google.com/java/awt/Window.html#applyResourceBundle(java.util.ResourceBundle)), [applyResourceBundle](http://docs.google.com/java/awt/Window.html#applyResourceBundle(java.lang.String)), [createBufferStrategy](http://docs.google.com/java/awt/Window.html#createBufferStrategy(int)), [createBufferStrategy](http://docs.google.com/java/awt/Window.html#createBufferStrategy(int,%20java.awt.BufferCapabilities)), [dispose](http://docs.google.com/java/awt/Window.html#dispose()), [getBufferStrategy](http://docs.google.com/java/awt/Window.html#getBufferStrategy()), [getFocusableWindowState](http://docs.google.com/java/awt/Window.html#getFocusableWindowState()), [getFocusCycleRootAncestor](http://docs.google.com/java/awt/Window.html#getFocusCycleRootAncestor()), [getFocusOwner](http://docs.google.com/java/awt/Window.html#getFocusOwner()), [getFocusTraversalKeys](http://docs.google.com/java/awt/Window.html#getFocusTraversalKeys(int)), [getGraphicsConfiguration](http://docs.google.com/java/awt/Window.html#getGraphicsConfiguration()), [getIconImages](http://docs.google.com/java/awt/Window.html#getIconImages()), [getInputContext](http://docs.google.com/java/awt/Window.html#getInputContext()), [getListeners](http://docs.google.com/java/awt/Window.html#getListeners(java.lang.Class)), [getLocale](http://docs.google.com/java/awt/Window.html#getLocale()), [getModalExclusionType](http://docs.google.com/java/awt/Window.html#getModalExclusionType()), [getMostRecentFocusOwner](http://docs.google.com/java/awt/Window.html#getMostRecentFocusOwner()), [getOwnedWindows](http://docs.google.com/java/awt/Window.html#getOwnedWindows()), [getOwner](http://docs.google.com/java/awt/Window.html#getOwner()), [getOwnerlessWindows](http://docs.google.com/java/awt/Window.html#getOwnerlessWindows()), [getToolkit](http://docs.google.com/java/awt/Window.html#getToolkit()), [getWarningString](http://docs.google.com/java/awt/Window.html#getWarningString()), [getWindowFocusListeners](http://docs.google.com/java/awt/Window.html#getWindowFocusListeners()), [getWindowListeners](http://docs.google.com/java/awt/Window.html#getWindowListeners()), [getWindows](http://docs.google.com/java/awt/Window.html#getWindows()), [getWindowStateListeners](http://docs.google.com/java/awt/Window.html#getWindowStateListeners()), [hide](http://docs.google.com/java/awt/Window.html#hide()), [isActive](http://docs.google.com/java/awt/Window.html#isActive()), [isAlwaysOnTop](http://docs.google.com/java/awt/Window.html#isAlwaysOnTop()), [isAlwaysOnTopSupported](http://docs.google.com/java/awt/Window.html#isAlwaysOnTopSupported()), [isFocusableWindow](http://docs.google.com/java/awt/Window.html#isFocusableWindow()), [isFocusCycleRoot](http://docs.google.com/java/awt/Window.html#isFocusCycleRoot()), [isFocused](http://docs.google.com/java/awt/Window.html#isFocused()), [isLocationByPlatform](http://docs.google.com/java/awt/Window.html#isLocationByPlatform()), [isShowing](http://docs.google.com/java/awt/Window.html#isShowing()), [pack](http://docs.google.com/java/awt/Window.html#pack()), [postEvent](http://docs.google.com/java/awt/Window.html#postEvent(java.awt.Event)), [processEvent](http://docs.google.com/java/awt/Window.html#processEvent(java.awt.AWTEvent)), [processWindowEvent](http://docs.google.com/java/awt/Window.html#processWindowEvent(java.awt.event.WindowEvent)), [processWindowFocusEvent](http://docs.google.com/java/awt/Window.html#processWindowFocusEvent(java.awt.event.WindowEvent)), [processWindowStateEvent](http://docs.google.com/java/awt/Window.html#processWindowStateEvent(java.awt.event.WindowEvent)), [removeWindowFocusListener](http://docs.google.com/java/awt/Window.html#removeWindowFocusListener(java.awt.event.WindowFocusListener)), [removeWindowListener](http://docs.google.com/java/awt/Window.html#removeWindowListener(java.awt.event.WindowListener)), [removeWindowStateListener](http://docs.google.com/java/awt/Window.html#removeWindowStateListener(java.awt.event.WindowStateListener)), [reshape](http://docs.google.com/java/awt/Window.html#reshape(int,%20int,%20int,%20int)), [setAlwaysOnTop](http://docs.google.com/java/awt/Window.html#setAlwaysOnTop(boolean)), [setBounds](http://docs.google.com/java/awt/Window.html#setBounds(int,%20int,%20int,%20int)), [setBounds](http://docs.google.com/java/awt/Window.html#setBounds(java.awt.Rectangle)), [setCursor](http://docs.google.com/java/awt/Window.html#setCursor(java.awt.Cursor)), [setFocusableWindowState](http://docs.google.com/java/awt/Window.html#setFocusableWindowState(boolean)), [setFocusCycleRoot](http://docs.google.com/java/awt/Window.html#setFocusCycleRoot(boolean)), [setIconImages](http://docs.google.com/java/awt/Window.html#setIconImages(java.util.List)), [setLocationByPlatform](http://docs.google.com/java/awt/Window.html#setLocationByPlatform(boolean)), [setLocationRelativeTo](http://docs.google.com/java/awt/Window.html#setLocationRelativeTo(java.awt.Component)), [setMinimumSize](http://docs.google.com/java/awt/Window.html#setMinimumSize(java.awt.Dimension)), [setModalExclusionType](http://docs.google.com/java/awt/Window.html#setModalExclusionType(java.awt.Dialog.ModalExclusionType)), [setSize](http://docs.google.com/java/awt/Window.html#setSize(java.awt.Dimension)), [setSize](http://docs.google.com/java/awt/Window.html#setSize(int,%20int)), [setVisible](http://docs.google.com/java/awt/Window.html#setVisible(boolean)), [show](http://docs.google.com/java/awt/Window.html#show()), [toBack](http://docs.google.com/java/awt/Window.html#toBack()), [toFront](http://docs.google.com/java/awt/Window.html#toFront()) |

| **Methods inherited from class java.awt.**[**Container**](http://docs.google.com/java/awt/Container.html) |
| --- |
| [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20int)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Container.html#add(java.awt.Component,%20java.lang.Object,%20int)), [add](http://docs.google.com/java/awt/Container.html#add(java.lang.String,%20java.awt.Component)), [addContainerListener](http://docs.google.com/java/awt/Container.html#addContainerListener(java.awt.event.ContainerListener)), [addImpl](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [applyComponentOrientation](http://docs.google.com/java/awt/Container.html#applyComponentOrientation(java.awt.ComponentOrientation)), [areFocusTraversalKeysSet](http://docs.google.com/java/awt/Container.html#areFocusTraversalKeysSet(int)), [countComponents](http://docs.google.com/java/awt/Container.html#countComponents()), [deliverEvent](http://docs.google.com/java/awt/Container.html#deliverEvent(java.awt.Event)), [doLayout](http://docs.google.com/java/awt/Container.html#doLayout()), [findComponentAt](http://docs.google.com/java/awt/Container.html#findComponentAt(int,%20int)), [findComponentAt](http://docs.google.com/java/awt/Container.html#findComponentAt(java.awt.Point)), [getAlignmentX](http://docs.google.com/java/awt/Container.html#getAlignmentX()), [getAlignmentY](http://docs.google.com/java/awt/Container.html#getAlignmentY()), [getComponent](http://docs.google.com/java/awt/Container.html#getComponent(int)), [getComponentAt](http://docs.google.com/java/awt/Container.html#getComponentAt(int,%20int)), [getComponentAt](http://docs.google.com/java/awt/Container.html#getComponentAt(java.awt.Point)), [getComponentCount](http://docs.google.com/java/awt/Container.html#getComponentCount()), [getComponents](http://docs.google.com/java/awt/Container.html#getComponents()), [getComponentZOrder](http://docs.google.com/java/awt/Container.html#getComponentZOrder(java.awt.Component)), [getContainerListeners](http://docs.google.com/java/awt/Container.html#getContainerListeners()), [getFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#getFocusTraversalPolicy()), [getInsets](http://docs.google.com/java/awt/Container.html#getInsets()), [getLayout](http://docs.google.com/java/awt/Container.html#getLayout()), [getMaximumSize](http://docs.google.com/java/awt/Container.html#getMaximumSize()), [getMinimumSize](http://docs.google.com/java/awt/Container.html#getMinimumSize()), [getMousePosition](http://docs.google.com/java/awt/Container.html#getMousePosition(boolean)), [getPreferredSize](http://docs.google.com/java/awt/Container.html#getPreferredSize()), [insets](http://docs.google.com/java/awt/Container.html#insets()), [invalidate](http://docs.google.com/java/awt/Container.html#invalidate()), [isAncestorOf](http://docs.google.com/java/awt/Container.html#isAncestorOf(java.awt.Component)), [isFocusCycleRoot](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot(java.awt.Container)), [isFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicyProvider()), [isFocusTraversalPolicySet](http://docs.google.com/java/awt/Container.html#isFocusTraversalPolicySet()), [layout](http://docs.google.com/java/awt/Container.html#layout()), [list](http://docs.google.com/java/awt/Container.html#list(java.io.PrintStream,%20int)), [list](http://docs.google.com/java/awt/Container.html#list(java.io.PrintWriter,%20int)), [locate](http://docs.google.com/java/awt/Container.html#locate(int,%20int)), [minimumSize](http://docs.google.com/java/awt/Container.html#minimumSize()), [paint](http://docs.google.com/java/awt/Container.html#paint(java.awt.Graphics)), [paintComponents](http://docs.google.com/java/awt/Container.html#paintComponents(java.awt.Graphics)), [preferredSize](http://docs.google.com/java/awt/Container.html#preferredSize()), [print](http://docs.google.com/java/awt/Container.html#print(java.awt.Graphics)), [printComponents](http://docs.google.com/java/awt/Container.html#printComponents(java.awt.Graphics)), [processContainerEvent](http://docs.google.com/java/awt/Container.html#processContainerEvent(java.awt.event.ContainerEvent)), [remove](http://docs.google.com/java/awt/Container.html#remove(java.awt.Component)), [remove](http://docs.google.com/java/awt/Container.html#remove(int)), [removeAll](http://docs.google.com/java/awt/Container.html#removeAll()), [removeContainerListener](http://docs.google.com/java/awt/Container.html#removeContainerListener(java.awt.event.ContainerListener)), [setComponentZOrder](http://docs.google.com/java/awt/Container.html#setComponentZOrder(java.awt.Component,%20int)), [setFocusTraversalKeys](http://docs.google.com/java/awt/Container.html#setFocusTraversalKeys(int,%20java.util.Set)), [setFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [setFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicyProvider(boolean)), [setFont](http://docs.google.com/java/awt/Container.html#setFont(java.awt.Font)), [setLayout](http://docs.google.com/java/awt/Container.html#setLayout(java.awt.LayoutManager)), [transferFocusBackward](http://docs.google.com/java/awt/Container.html#transferFocusBackward()), [transferFocusDownCycle](http://docs.google.com/java/awt/Container.html#transferFocusDownCycle()), [update](http://docs.google.com/java/awt/Container.html#update(java.awt.Graphics)), [validate](http://docs.google.com/java/awt/Container.html#validate()), [validateTree](http://docs.google.com/java/awt/Container.html#validateTree()) |

| **Methods inherited from class java.awt.**[**Component**](http://docs.google.com/java/awt/Component.html) |
| --- |
| [action](http://docs.google.com/java/awt/Component.html#action(java.awt.Event,%20java.lang.Object)), [add](http://docs.google.com/java/awt/Component.html#add(java.awt.PopupMenu)), [addComponentListener](http://docs.google.com/java/awt/Component.html#addComponentListener(java.awt.event.ComponentListener)), [addFocusListener](http://docs.google.com/java/awt/Component.html#addFocusListener(java.awt.event.FocusListener)), [addHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#addHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [addHierarchyListener](http://docs.google.com/java/awt/Component.html#addHierarchyListener(java.awt.event.HierarchyListener)), [addInputMethodListener](http://docs.google.com/java/awt/Component.html#addInputMethodListener(java.awt.event.InputMethodListener)), [addKeyListener](http://docs.google.com/java/awt/Component.html#addKeyListener(java.awt.event.KeyListener)), [addMouseListener](http://docs.google.com/java/awt/Component.html#addMouseListener(java.awt.event.MouseListener)), [addMouseMotionListener](http://docs.google.com/java/awt/Component.html#addMouseMotionListener(java.awt.event.MouseMotionListener)), [addMouseWheelListener](http://docs.google.com/java/awt/Component.html#addMouseWheelListener(java.awt.event.MouseWheelListener)), [bounds](http://docs.google.com/java/awt/Component.html#bounds()), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20java.awt.image.ImageObserver)), [checkImage](http://docs.google.com/java/awt/Component.html#checkImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [coalesceEvents](http://docs.google.com/java/awt/Component.html#coalesceEvents(java.awt.AWTEvent,%20java.awt.AWTEvent)), [contains](http://docs.google.com/java/awt/Component.html#contains(int,%20int)), [contains](http://docs.google.com/java/awt/Component.html#contains(java.awt.Point)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(java.awt.image.ImageProducer)), [createImage](http://docs.google.com/java/awt/Component.html#createImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int)), [createVolatileImage](http://docs.google.com/java/awt/Component.html#createVolatileImage(int,%20int,%20java.awt.ImageCapabilities)), [disable](http://docs.google.com/java/awt/Component.html#disable()), [disableEvents](http://docs.google.com/java/awt/Component.html#disableEvents(long)), [dispatchEvent](http://docs.google.com/java/awt/Component.html#dispatchEvent(java.awt.AWTEvent)), [enable](http://docs.google.com/java/awt/Component.html#enable()), [enable](http://docs.google.com/java/awt/Component.html#enable(boolean)), [enableEvents](http://docs.google.com/java/awt/Component.html#enableEvents(long)), [enableInputMethods](http://docs.google.com/java/awt/Component.html#enableInputMethods(boolean)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20boolean,%20boolean)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20byte,%20byte)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20char,%20char)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20double,%20double)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20float,%20float)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20int,%20int)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20long,%20long)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)), [firePropertyChange](http://docs.google.com/java/awt/Component.html#firePropertyChange(java.lang.String,%20short,%20short)), [getBackground](http://docs.google.com/java/awt/Component.html#getBackground()), [getBaseline](http://docs.google.com/java/awt/Component.html#getBaseline(int,%20int)), [getBaselineResizeBehavior](http://docs.google.com/java/awt/Component.html#getBaselineResizeBehavior()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds(java.awt.Rectangle)), [getColorModel](http://docs.google.com/java/awt/Component.html#getColorModel()), [getComponentListeners](http://docs.google.com/java/awt/Component.html#getComponentListeners()), [getComponentOrientation](http://docs.google.com/java/awt/Component.html#getComponentOrientation()), [getCursor](http://docs.google.com/java/awt/Component.html#getCursor()), [getDropTarget](http://docs.google.com/java/awt/Component.html#getDropTarget()), [getFocusListeners](http://docs.google.com/java/awt/Component.html#getFocusListeners()), [getFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#getFocusTraversalKeysEnabled()), [getFont](http://docs.google.com/java/awt/Component.html#getFont()), [getFontMetrics](http://docs.google.com/java/awt/Component.html#getFontMetrics(java.awt.Font)), [getForeground](http://docs.google.com/java/awt/Component.html#getForeground()), [getGraphics](http://docs.google.com/java/awt/Component.html#getGraphics()), [getHeight](http://docs.google.com/java/awt/Component.html#getHeight()), [getHierarchyBoundsListeners](http://docs.google.com/java/awt/Component.html#getHierarchyBoundsListeners()), [getHierarchyListeners](http://docs.google.com/java/awt/Component.html#getHierarchyListeners()), [getIgnoreRepaint](http://docs.google.com/java/awt/Component.html#getIgnoreRepaint()), [getInputMethodListeners](http://docs.google.com/java/awt/Component.html#getInputMethodListeners()), [getInputMethodRequests](http://docs.google.com/java/awt/Component.html#getInputMethodRequests()), [getKeyListeners](http://docs.google.com/java/awt/Component.html#getKeyListeners()), [getLocation](http://docs.google.com/java/awt/Component.html#getLocation()), [getLocation](http://docs.google.com/java/awt/Component.html#getLocation(java.awt.Point)), [getLocationOnScreen](http://docs.google.com/java/awt/Component.html#getLocationOnScreen()), [getMouseListeners](http://docs.google.com/java/awt/Component.html#getMouseListeners()), [getMouseMotionListeners](http://docs.google.com/java/awt/Component.html#getMouseMotionListeners()), [getMousePosition](http://docs.google.com/java/awt/Component.html#getMousePosition()), [getMouseWheelListeners](http://docs.google.com/java/awt/Component.html#getMouseWheelListeners()), [getName](http://docs.google.com/java/awt/Component.html#getName()), [getParent](http://docs.google.com/java/awt/Component.html#getParent()), [getPeer](http://docs.google.com/java/awt/Component.html#getPeer()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners()), [getPropertyChangeListeners](http://docs.google.com/java/awt/Component.html#getPropertyChangeListeners(java.lang.String)), [getSize](http://docs.google.com/java/awt/Component.html#getSize()), [getSize](http://docs.google.com/java/awt/Component.html#getSize(java.awt.Dimension)), [getTreeLock](http://docs.google.com/java/awt/Component.html#getTreeLock()), [getWidth](http://docs.google.com/java/awt/Component.html#getWidth()), [getX](http://docs.google.com/java/awt/Component.html#getX()), [getY](http://docs.google.com/java/awt/Component.html#getY()), [gotFocus](http://docs.google.com/java/awt/Component.html#gotFocus(java.awt.Event,%20java.lang.Object)), [handleEvent](http://docs.google.com/java/awt/Component.html#handleEvent(java.awt.Event)), [hasFocus](http://docs.google.com/java/awt/Component.html#hasFocus()), [imageUpdate](http://docs.google.com/java/awt/Component.html#imageUpdate(java.awt.Image,%20int,%20int,%20int,%20int,%20int)), [inside](http://docs.google.com/java/awt/Component.html#inside(int,%20int)), [isBackgroundSet](http://docs.google.com/java/awt/Component.html#isBackgroundSet()), [isCursorSet](http://docs.google.com/java/awt/Component.html#isCursorSet()), [isDisplayable](http://docs.google.com/java/awt/Component.html#isDisplayable()), [isDoubleBuffered](http://docs.google.com/java/awt/Component.html#isDoubleBuffered()), [isEnabled](http://docs.google.com/java/awt/Component.html#isEnabled()), [isFocusable](http://docs.google.com/java/awt/Component.html#isFocusable()), [isFocusOwner](http://docs.google.com/java/awt/Component.html#isFocusOwner()), [isFocusTraversable](http://docs.google.com/java/awt/Component.html#isFocusTraversable()), [isFontSet](http://docs.google.com/java/awt/Component.html#isFontSet()), [isForegroundSet](http://docs.google.com/java/awt/Component.html#isForegroundSet()), [isLightweight](http://docs.google.com/java/awt/Component.html#isLightweight()), [isMaximumSizeSet](http://docs.google.com/java/awt/Component.html#isMaximumSizeSet()), [isMinimumSizeSet](http://docs.google.com/java/awt/Component.html#isMinimumSizeSet()), [isOpaque](http://docs.google.com/java/awt/Component.html#isOpaque()), [isPreferredSizeSet](http://docs.google.com/java/awt/Component.html#isPreferredSizeSet()), [isValid](http://docs.google.com/java/awt/Component.html#isValid()), [isVisible](http://docs.google.com/java/awt/Component.html#isVisible()), [keyDown](http://docs.google.com/java/awt/Component.html#keyDown(java.awt.Event,%20int)), [keyUp](http://docs.google.com/java/awt/Component.html#keyUp(java.awt.Event,%20int)), [list](http://docs.google.com/java/awt/Component.html#list()), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintStream)), [list](http://docs.google.com/java/awt/Component.html#list(java.io.PrintWriter)), [location](http://docs.google.com/java/awt/Component.html#location()), [lostFocus](http://docs.google.com/java/awt/Component.html#lostFocus(java.awt.Event,%20java.lang.Object)), [mouseDown](http://docs.google.com/java/awt/Component.html#mouseDown(java.awt.Event,%20int,%20int)), [mouseDrag](http://docs.google.com/java/awt/Component.html#mouseDrag(java.awt.Event,%20int,%20int)), [mouseEnter](http://docs.google.com/java/awt/Component.html#mouseEnter(java.awt.Event,%20int,%20int)), [mouseExit](http://docs.google.com/java/awt/Component.html#mouseExit(java.awt.Event,%20int,%20int)), [mouseMove](http://docs.google.com/java/awt/Component.html#mouseMove(java.awt.Event,%20int,%20int)), [mouseUp](http://docs.google.com/java/awt/Component.html#mouseUp(java.awt.Event,%20int,%20int)), [move](http://docs.google.com/java/awt/Component.html#move(int,%20int)), [nextFocus](http://docs.google.com/java/awt/Component.html#nextFocus()), [paintAll](http://docs.google.com/java/awt/Component.html#paintAll(java.awt.Graphics)), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20java.awt.image.ImageObserver)), [prepareImage](http://docs.google.com/java/awt/Component.html#prepareImage(java.awt.Image,%20int,%20int,%20java.awt.image.ImageObserver)), [printAll](http://docs.google.com/java/awt/Component.html#printAll(java.awt.Graphics)), [processComponentEvent](http://docs.google.com/java/awt/Component.html#processComponentEvent(java.awt.event.ComponentEvent)), [processFocusEvent](http://docs.google.com/java/awt/Component.html#processFocusEvent(java.awt.event.FocusEvent)), [processHierarchyBoundsEvent](http://docs.google.com/java/awt/Component.html#processHierarchyBoundsEvent(java.awt.event.HierarchyEvent)), [processHierarchyEvent](http://docs.google.com/java/awt/Component.html#processHierarchyEvent(java.awt.event.HierarchyEvent)), [processInputMethodEvent](http://docs.google.com/java/awt/Component.html#processInputMethodEvent(java.awt.event.InputMethodEvent)), [processKeyEvent](http://docs.google.com/java/awt/Component.html#processKeyEvent(java.awt.event.KeyEvent)), [processMouseEvent](http://docs.google.com/java/awt/Component.html#processMouseEvent(java.awt.event.MouseEvent)), [processMouseMotionEvent](http://docs.google.com/java/awt/Component.html#processMouseMotionEvent(java.awt.event.MouseEvent)), [processMouseWheelEvent](http://docs.google.com/java/awt/Component.html#processMouseWheelEvent(java.awt.event.MouseWheelEvent)), [removeComponentListener](http://docs.google.com/java/awt/Component.html#removeComponentListener(java.awt.event.ComponentListener)), [removeFocusListener](http://docs.google.com/java/awt/Component.html#removeFocusListener(java.awt.event.FocusListener)), [removeHierarchyBoundsListener](http://docs.google.com/java/awt/Component.html#removeHierarchyBoundsListener(java.awt.event.HierarchyBoundsListener)), [removeHierarchyListener](http://docs.google.com/java/awt/Component.html#removeHierarchyListener(java.awt.event.HierarchyListener)), [removeInputMethodListener](http://docs.google.com/java/awt/Component.html#removeInputMethodListener(java.awt.event.InputMethodListener)), [removeKeyListener](http://docs.google.com/java/awt/Component.html#removeKeyListener(java.awt.event.KeyListener)), [removeMouseListener](http://docs.google.com/java/awt/Component.html#removeMouseListener(java.awt.event.MouseListener)), [removeMouseMotionListener](http://docs.google.com/java/awt/Component.html#removeMouseMotionListener(java.awt.event.MouseMotionListener)), [removeMouseWheelListener](http://docs.google.com/java/awt/Component.html#removeMouseWheelListener(java.awt.event.MouseWheelListener)), [removePropertyChangeListener](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.beans.PropertyChangeListener)), [removePropertyChangeListener](http://docs.google.com/java/awt/Component.html#removePropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [repaint](http://docs.google.com/java/awt/Component.html#repaint()), [repaint](http://docs.google.com/java/awt/Component.html#repaint(int,%20int,%20int,%20int)), [repaint](http://docs.google.com/java/awt/Component.html#repaint(long)), [repaint](http://docs.google.com/java/awt/Component.html#repaint(long,%20int,%20int,%20int,%20int)), [requestFocus](http://docs.google.com/java/awt/Component.html#requestFocus()), [requestFocus](http://docs.google.com/java/awt/Component.html#requestFocus(boolean)), [requestFocusInWindow](http://docs.google.com/java/awt/Component.html#requestFocusInWindow()), [requestFocusInWindow](http://docs.google.com/java/awt/Component.html#requestFocusInWindow(boolean)), [resize](http://docs.google.com/java/awt/Component.html#resize(java.awt.Dimension)), [resize](http://docs.google.com/java/awt/Component.html#resize(int,%20int)), [setBackground](http://docs.google.com/java/awt/Component.html#setBackground(java.awt.Color)), [setComponentOrientation](http://docs.google.com/java/awt/Component.html#setComponentOrientation(java.awt.ComponentOrientation)), [setDropTarget](http://docs.google.com/java/awt/Component.html#setDropTarget(java.awt.dnd.DropTarget)), [setEnabled](http://docs.google.com/java/awt/Component.html#setEnabled(boolean)), [setFocusable](http://docs.google.com/java/awt/Component.html#setFocusable(boolean)), [setFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeysEnabled(boolean)), [setForeground](http://docs.google.com/java/awt/Component.html#setForeground(java.awt.Color)), [setIgnoreRepaint](http://docs.google.com/java/awt/Component.html#setIgnoreRepaint(boolean)), [setLocale](http://docs.google.com/java/awt/Component.html#setLocale(java.util.Locale)), [setLocation](http://docs.google.com/java/awt/Component.html#setLocation(int,%20int)), [setLocation](http://docs.google.com/java/awt/Component.html#setLocation(java.awt.Point)), [setMaximumSize](http://docs.google.com/java/awt/Component.html#setMaximumSize(java.awt.Dimension)), [setName](http://docs.google.com/java/awt/Component.html#setName(java.lang.String)), [setPreferredSize](http://docs.google.com/java/awt/Component.html#setPreferredSize(java.awt.Dimension)), [show](http://docs.google.com/java/awt/Component.html#show(boolean)), [size](http://docs.google.com/java/awt/Component.html#size()), [toString](http://docs.google.com/java/awt/Component.html#toString()), [transferFocus](http://docs.google.com/java/awt/Component.html#transferFocus()), [transferFocusUpCycle](http://docs.google.com/java/awt/Component.html#transferFocusUpCycle()) |

| **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)) |

| **Methods inherited from interface java.awt.**[**MenuContainer**](http://docs.google.com/java/awt/MenuContainer.html) |
| --- |
| [getFont](http://docs.google.com/java/awt/MenuContainer.html#getFont()), [postEvent](http://docs.google.com/java/awt/MenuContainer.html#postEvent(java.awt.Event)) |

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

### DEFAULT\_CURSOR

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public static final int **DEFAULT\_CURSOR**

**Deprecated.** *replaced by Cursor.DEFAULT\_CURSOR.* **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.DEFAULT_CURSOR)

### CROSSHAIR\_CURSOR

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public static final int **CROSSHAIR\_CURSOR**

**Deprecated.** *replaced by Cursor.CROSSHAIR\_CURSOR.* **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.CROSSHAIR_CURSOR)

### TEXT\_CURSOR

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public static final int **TEXT\_CURSOR**

**Deprecated.** *replaced by Cursor.TEXT\_CURSOR.* **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.TEXT_CURSOR)

### WAIT\_CURSOR

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public static final int **WAIT\_CURSOR**

**Deprecated.** *replaced by Cursor.WAIT\_CURSOR.* **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.WAIT_CURSOR)

### SW\_RESIZE\_CURSOR

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public static final int **SW\_RESIZE\_CURSOR**

**Deprecated.** *replaced by Cursor.SW\_RESIZE\_CURSOR.* **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.SW_RESIZE_CURSOR)

### SE\_RESIZE\_CURSOR

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public static final int **SE\_RESIZE\_CURSOR**

**Deprecated.** *replaced by Cursor.SE\_RESIZE\_CURSOR.* **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.SE_RESIZE_CURSOR)

### NW\_RESIZE\_CURSOR

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public static final int **NW\_RESIZE\_CURSOR**

**Deprecated.** *replaced by Cursor.NW\_RESIZE\_CURSOR.* **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.NW_RESIZE_CURSOR)

### NE\_RESIZE\_CURSOR

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public static final int **NE\_RESIZE\_CURSOR**

**Deprecated.** *replaced by Cursor.NE\_RESIZE\_CURSOR.* **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.NE_RESIZE_CURSOR)

### N\_RESIZE\_CURSOR

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public static final int **N\_RESIZE\_CURSOR**

**Deprecated.** *replaced by Cursor.N\_RESIZE\_CURSOR.* **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.N_RESIZE_CURSOR)

### S\_RESIZE\_CURSOR

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public static final int **S\_RESIZE\_CURSOR**

**Deprecated.** *replaced by Cursor.S\_RESIZE\_CURSOR.* **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.S_RESIZE_CURSOR)

### W\_RESIZE\_CURSOR

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public static final int **W\_RESIZE\_CURSOR**

**Deprecated.** *replaced by Cursor.W\_RESIZE\_CURSOR.* **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.W_RESIZE_CURSOR)

### E\_RESIZE\_CURSOR

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public static final int **E\_RESIZE\_CURSOR**

**Deprecated.** *replaced by Cursor.E\_RESIZE\_CURSOR.* **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.E_RESIZE_CURSOR)

### HAND\_CURSOR

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public static final int **HAND\_CURSOR**

**Deprecated.** *replaced by Cursor.HAND\_CURSOR.* **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.HAND_CURSOR)

### MOVE\_CURSOR

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public static final int **MOVE\_CURSOR**

**Deprecated.** *replaced by Cursor.MOVE\_CURSOR.* **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.MOVE_CURSOR)

### NORMAL

public static final int **NORMAL**

Frame is in the "normal" state. This symbolic constant names a frame state with all state bits cleared.

**See Also:**[setExtendedState(int)](http://docs.google.com/java/awt/Frame.html#setExtendedState(int)), [getExtendedState()](http://docs.google.com/java/awt/Frame.html#getExtendedState()), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.NORMAL)

### ICONIFIED

public static final int **ICONIFIED**

This state bit indicates that frame is iconified.

**See Also:**[setExtendedState(int)](http://docs.google.com/java/awt/Frame.html#setExtendedState(int)), [getExtendedState()](http://docs.google.com/java/awt/Frame.html#getExtendedState()), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.ICONIFIED)

### MAXIMIZED\_HORIZ

public static final int **MAXIMIZED\_HORIZ**

This state bit indicates that frame is maximized in the horizontal direction.

**Since:** 1.4 **See Also:**[setExtendedState(int)](http://docs.google.com/java/awt/Frame.html#setExtendedState(int)), [getExtendedState()](http://docs.google.com/java/awt/Frame.html#getExtendedState()), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.MAXIMIZED_HORIZ)

### MAXIMIZED\_VERT

public static final int **MAXIMIZED\_VERT**

This state bit indicates that frame is maximized in the vertical direction.

**Since:** 1.4 **See Also:**[setExtendedState(int)](http://docs.google.com/java/awt/Frame.html#setExtendedState(int)), [getExtendedState()](http://docs.google.com/java/awt/Frame.html#getExtendedState()), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.MAXIMIZED_VERT)

### MAXIMIZED\_BOTH

public static final int **MAXIMIZED\_BOTH**

This state bit mask indicates that frame is fully maximized (that is both horizontally and vertically). It is just a convenience alias for MAXIMIZED\_VERT | MAXIMIZED\_HORIZ.

Note that the correct test for frame being fully maximized is

(state & Frame.MAXIMIZED\_BOTH) == Frame.MAXIMIZED\_BOTH

To test is frame is maximized in *some* direction use

(state & Frame.MAXIMIZED\_BOTH) != 0

**Since:** 1.4 **See Also:**[setExtendedState(int)](http://docs.google.com/java/awt/Frame.html#setExtendedState(int)), [getExtendedState()](http://docs.google.com/java/awt/Frame.html#getExtendedState()), [Constant Field Values](http://docs.google.com/constant-values.html#java.awt.Frame.MAXIMIZED_BOTH)

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

### Frame

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

Constructs a new instance of Frame that is initially invisible. The title of the Frame is empty.

**Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [Component.setSize(int, int)](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [Component.setVisible(boolean)](http://docs.google.com/java/awt/Component.html#setVisible(boolean))

### Frame

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

Constructs a new, initially invisible Frame with the specified GraphicsConfiguration.

**Parameters:**gc - the GraphicsConfiguration of the target screen device. If gc is null, the system default GraphicsConfiguration is assumed. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if gc is not from a screen device. [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true**Since:** 1.3 **See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### Frame

public **Frame**([String](http://docs.google.com/java/lang/String.html) title)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Constructs a new, initially invisible Frame object with the specified title.

**Parameters:**title - the title to be displayed in the frame's border. A null value is treated as an empty string, "". **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [Component.setSize(int, int)](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [Component.setVisible(boolean)](http://docs.google.com/java/awt/Component.html#setVisible(boolean)), [GraphicsConfiguration.getBounds()](http://docs.google.com/java/awt/GraphicsConfiguration.html#getBounds())

### Frame

public **Frame**([String](http://docs.google.com/java/lang/String.html) title,  
 [GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html) gc)

Constructs a new, initially invisible Frame object with the specified title and a GraphicsConfiguration.

**Parameters:**title - the title to be displayed in the frame's border. A null value is treated as an empty string, "".gc - the GraphicsConfiguration of the target screen device. If gc is null, the system default GraphicsConfiguration is assumed. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if gc is not from a screen device. [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true**Since:** 1.3 **See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [Component.setSize(int, int)](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [Component.setVisible(boolean)](http://docs.google.com/java/awt/Component.html#setVisible(boolean)), [GraphicsConfiguration.getBounds()](http://docs.google.com/java/awt/GraphicsConfiguration.html#getBounds())

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

### addNotify

public void **addNotify**()

Makes this Frame displayable by connecting it to a native screen resource. Making a frame displayable will cause any of its children to be made displayable. This method is called internally by the toolkit and should not be called directly by programs.

**Overrides:**[addNotify](http://docs.google.com/java/awt/Window.html#addNotify()) in class [Window](http://docs.google.com/java/awt/Window.html) **See Also:**[Component.isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable()), [removeNotify()](http://docs.google.com/java/awt/Frame.html#removeNotify())

### getTitle

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

Gets the title of the frame. The title is displayed in the frame's border.

**Returns:**the title of this frame, or an empty string ("") if this frame doesn't have a title.**See Also:**[setTitle(String)](http://docs.google.com/java/awt/Frame.html#setTitle(java.lang.String))

### setTitle

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

Sets the title for this frame to the specified string.

**Parameters:**title - the title to be displayed in the frame's border. A null value is treated as an empty string, "".**See Also:**[getTitle()](http://docs.google.com/java/awt/Frame.html#getTitle())

### getIconImage

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

Returns the image to be displayed as the icon for this frame.

This method is obsolete and kept for backward compatibility only. Use [Window.getIconImages()](http://docs.google.com/java/awt/Window.html#getIconImages()) instead.

If a list of several images was specified as a Window's icon, this method will return the first item of the list.

**Returns:**the icon image for this frame, or null if this frame doesn't have an icon image.**See Also:**[setIconImage(Image)](http://docs.google.com/java/awt/Frame.html#setIconImage(java.awt.Image)), [Window.getIconImages()](http://docs.google.com/java/awt/Window.html#getIconImages()), [Window.setIconImages(java.util.List)](http://docs.google.com/java/awt/Window.html#setIconImages(java.util.List))

### setIconImage

public void **setIconImage**([Image](http://docs.google.com/java/awt/Image.html) image)

Sets the image to be displayed as the icon for this window.

This method can be used instead of [setIconImages()](http://docs.google.com/java/awt/Window.html#setIconImages(java.util.List)) to specify a single image as a window's icon.

The following statement:

setIconImage(image);

is equivalent to:

ArrayList imageList = new ArrayList();  
 imageList.add(image);  
 setIconImages(imageList);

Note : Native windowing systems may use different images of differing dimensions to represent a window, depending on the context (e.g. window decoration, window list, taskbar, etc.). They could also use just a single image for all contexts or no image at all.

**Overrides:**[setIconImage](http://docs.google.com/java/awt/Window.html#setIconImage(java.awt.Image)) in class [Window](http://docs.google.com/java/awt/Window.html) **Parameters:**image - the icon image to be displayed.**See Also:**[Window.setIconImages(java.util.List)](http://docs.google.com/java/awt/Window.html#setIconImages(java.util.List)), [Window.getIconImages()](http://docs.google.com/java/awt/Window.html#getIconImages())

### getMenuBar

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

Gets the menu bar for this frame.

**Returns:**the menu bar for this frame, or null if this frame doesn't have a menu bar.**See Also:**[setMenuBar(MenuBar)](http://docs.google.com/java/awt/Frame.html#setMenuBar(java.awt.MenuBar))

### setMenuBar

public void **setMenuBar**([MenuBar](http://docs.google.com/java/awt/MenuBar.html) mb)

Sets the menu bar for this frame to the specified menu bar.

**Parameters:**mb - the menu bar being set. If this parameter is null then any existing menu bar on this frame is removed.**See Also:**[getMenuBar()](http://docs.google.com/java/awt/Frame.html#getMenuBar())

### isResizable

public boolean **isResizable**()

Indicates whether this frame is resizable by the user. By default, all frames are initially resizable.

**Returns:**true if the user can resize this frame; false otherwise.**See Also:**[setResizable(boolean)](http://docs.google.com/java/awt/Frame.html#setResizable(boolean))

### setResizable

public void **setResizable**(boolean resizable)

Sets whether this frame is resizable by the user.

**Parameters:**resizable - true if this frame is resizable; false otherwise.**See Also:**[isResizable()](http://docs.google.com/java/awt/Frame.html#isResizable())

### setState

public void **setState**(int state)

Sets the state of this frame (obsolete).

In older versions of JDK a frame state could only be NORMAL or ICONIFIED. Since JDK 1.4 set of supported frame states is expanded and frame state is represented as a bitwise mask.

For compatibility with old programs this method still accepts Frame.NORMAL and Frame.ICONIFIED but it only changes the iconic state of the frame, other aspects of frame state are not affected by this method.

**Parameters:**state - either Frame.NORMAL or Frame.ICONIFIED.**See Also:**[getState()](http://docs.google.com/java/awt/Frame.html#getState()), [setExtendedState(int)](http://docs.google.com/java/awt/Frame.html#setExtendedState(int))

### setExtendedState

public void **setExtendedState**(int state)

Sets the state of this frame. The state is represented as a bitwise mask.

* NORMAL  
  Indicates that no state bits are set.
* ICONIFIED
* MAXIMIZED\_HORIZ
* MAXIMIZED\_VERT
* MAXIMIZED\_BOTH  
  Concatenates MAXIMIZED\_HORIZ and MAXIMIZED\_VERT.

Note that if the state is not supported on a given platform, nothing will happen. The application may determine if a specific state is available via the java.awt.Toolkit#isFrameStateSupported(int state) method.

**Parameters:**state - a bitwise mask of frame state constants**Since:** 1.4 **See Also:**[getExtendedState()](http://docs.google.com/java/awt/Frame.html#getExtendedState()), [Toolkit.isFrameStateSupported(int)](http://docs.google.com/java/awt/Toolkit.html#isFrameStateSupported(int))

### getState

public int **getState**()

Gets the state of this frame (obsolete).

In older versions of JDK a frame state could only be NORMAL or ICONIFIED. Since JDK 1.4 set of supported frame states is expanded and frame state is represented as a bitwise mask.

For compatibility with old programs this method still returns Frame.NORMAL and Frame.ICONIFIED but it only reports the iconic state of the frame, other aspects of frame state are not reported by this method.

**Returns:**Frame.NORMAL or Frame.ICONIFIED.**See Also:**[setState(int)](http://docs.google.com/java/awt/Frame.html#setState(int)), [getExtendedState()](http://docs.google.com/java/awt/Frame.html#getExtendedState())

### getExtendedState

public int **getExtendedState**()

Gets the state of this frame. The state is represented as a bitwise mask.

* NORMAL  
  Indicates that no state bits are set.
* ICONIFIED
* MAXIMIZED\_HORIZ
* MAXIMIZED\_VERT
* MAXIMIZED\_BOTH  
  Concatenates MAXIMIZED\_HORIZ and MAXIMIZED\_VERT.

**Returns:**a bitwise mask of frame state constants**Since:** 1.4 **See Also:**[setExtendedState(int)](http://docs.google.com/java/awt/Frame.html#setExtendedState(int))

### setMaximizedBounds

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

Sets the maximized bounds for this frame.

When a frame is in maximized state the system supplies some defaults bounds. This method allows some or all of those system supplied values to be overridden.

If bounds is null, accept bounds supplied by the system. If non-null you can override some of the system supplied values while accepting others by setting those fields you want to accept from system to Integer.MAX\_VALUE.

On some systems only the size portion of the bounds is taken into account.

**Parameters:**bounds - bounds for the maximized state**Since:** 1.4 **See Also:**[getMaximizedBounds()](http://docs.google.com/java/awt/Frame.html#getMaximizedBounds())

### getMaximizedBounds

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

Gets maximized bounds for this frame. Some fields may contain Integer.MAX\_VALUE to indicate that system supplied values for this field must be used.

**Returns:**maximized bounds for this frame; may be null**Since:** 1.4 **See Also:**[setMaximizedBounds(Rectangle)](http://docs.google.com/java/awt/Frame.html#setMaximizedBounds(java.awt.Rectangle))

### setUndecorated

public void **setUndecorated**(boolean undecorated)

Disables or enables decorations for this frame. This method can only be called while the frame is not displayable.

**Parameters:**undecorated - true if no frame decorations are to be enabled; false if frame decorations are to be enabled. **Throws:** IllegalComponentStateException - if the frame is displayable.**Since:** 1.4 **See Also:**[isUndecorated()](http://docs.google.com/java/awt/Frame.html#isUndecorated()), [Component.isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable()), [JFrame.setDefaultLookAndFeelDecorated(boolean)](http://docs.google.com/javax/swing/JFrame.html#setDefaultLookAndFeelDecorated(boolean))

### isUndecorated

public boolean **isUndecorated**()

Indicates whether this frame is undecorated. By default, all frames are initially decorated.

**Returns:**true if frame is undecorated; false otherwise.**Since:** 1.4 **See Also:**[setUndecorated(boolean)](http://docs.google.com/java/awt/Frame.html#setUndecorated(boolean))

### remove

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

Removes the specified menu bar from this frame.

**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)**Overrides:**[remove](http://docs.google.com/java/awt/Component.html#remove(java.awt.MenuComponent)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**m - the menu component to remove. If m is null, then no action is taken**See Also:**[Component.add(PopupMenu)](http://docs.google.com/java/awt/Component.html#add(java.awt.PopupMenu))

### removeNotify

public void **removeNotify**()

Makes this Frame undisplayable by removing its connection to its native screen resource. Making a Frame undisplayable will cause any of its children to be made undisplayable. This method is called by the toolkit internally and should not be called directly by programs.

**Overrides:**[removeNotify](http://docs.google.com/java/awt/Window.html#removeNotify()) in class [Window](http://docs.google.com/java/awt/Window.html) **See Also:**[Component.isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable()), [addNotify()](http://docs.google.com/java/awt/Frame.html#addNotify())

### paramString

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

Returns a string representing the state of this Frame. 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.

**Overrides:**[paramString](http://docs.google.com/java/awt/Container.html#paramString()) in class [Container](http://docs.google.com/java/awt/Container.html) **Returns:**the parameter string of this frame

### setCursor

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

**Deprecated.** *As of JDK version 1.1, replaced by Component.setCursor(Cursor).*

### getCursorType

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

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

### getFrames

public static [Frame](http://docs.google.com/java/awt/Frame.html)[] **getFrames**()

Returns an array of all Frames created by this application. If called from an applet, the array includes only the Frames accessible by that applet.

**Warning:** this method may return system created frames, such as a shared, hidden frame which is used by Swing. Applications should not assume the existence of these frames, nor should an application assume anything about these frames such as component positions, LayoutManagers or serialization.

**Note**: To obtain a list of all ownerless windows, including ownerless Dialogs (introduced in release 1.6), use [Window.getOwnerlessWindows](http://docs.google.com/java/awt/Window.html#getOwnerlessWindows()).

**Since:** 1.2 **See Also:**[Window.getWindows(sun.awt.AppContext)](http://docs.google.com/java/awt/Window.html#getWindows(sun.awt.AppContext)), [Window.getOwnerlessWindows()](http://docs.google.com/java/awt/Window.html#getOwnerlessWindows())

### getAccessibleContext

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

Gets the AccessibleContext associated with this Frame. For frames, the AccessibleContext takes the form of an AccessibleAWTFrame. A new AccessibleAWTFrame instance is created if necessary.

**Specified by:**[getAccessibleContext](http://docs.google.com/javax/accessibility/Accessible.html#getAccessibleContext()) in interface [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)**Overrides:**[getAccessibleContext](http://docs.google.com/java/awt/Window.html#getAccessibleContext()) in class [Window](http://docs.google.com/java/awt/Window.html) **Returns:**an AccessibleAWTFrame that serves as the AccessibleContext of this Frame**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/Frame.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/FontMetrics.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Frame.AccessibleAWTFrame.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Frame.html)    [**NO FRAMES**](http://docs.google.com/Frame.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#3rdcrjn) | DETAIL: [FIELD](#2jxsxqh) | [CONSTR](#41mghml) | [METHOD](#4f1mdlm) |

[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).