| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JFrame.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatterFactory.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JFrame.AccessibleJFrame.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JFrame.html)    [**NO FRAMES**](http://docs.google.com/JFrame.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#4d34og8) | [CONSTR](#lnxbz9) | [METHOD](#35nkun2) | DETAIL: [FIELD](#4i7ojhp) | [CONSTR](#qsh70q) | [METHOD](#147n2zr) |

## **javax.swing**

Class JFrame

[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](http://docs.google.com/java/awt/Frame.html)  
 **javax.swing.JFrame**

**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), [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html), [WindowConstants](http://docs.google.com/javax/swing/WindowConstants.html)

public class **JFrame**extends [Frame](http://docs.google.com/java/awt/Frame.html)implements [WindowConstants](http://docs.google.com/javax/swing/WindowConstants.html), [Accessible](http://docs.google.com/javax/accessibility/Accessible.html), [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html)

An extended version of java.awt.Frame that adds support for the JFC/Swing component architecture. You can find task-oriented documentation about using JFrame in *The Java Tutorial*, in the section [How to Make Frames](http://java.sun.com/docs/books/tutorial/uiswing/components/frame.html).

The JFrame class is slightly incompatible with Frame. Like all other JFC/Swing top-level containers, a JFrame contains a JRootPane as its only child. The **content pane** provided by the root pane should, as a rule, contain all the non-menu components displayed by the JFrame. This is different from the AWT Frame case. As a conveniance add and its variants, remove and setLayout have been overridden to forward to the contentPane as necessary. This means you can write:

frame.add(child);

And the child will be added to the contentPane. The content pane will always be non-null. Attempting to set it to null will cause the JFrame to throw an exception. The default content pane will have a BorderLayout manager set on it. Refer to [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html) for details on adding, removing and setting the LayoutManager of a JFrame.

Unlike a Frame, a JFrame has some notion of how to respond when the user attempts to close the window. The default behavior is to simply hide the JFrame when the user closes the window. To change the default behavior, you invoke the method [setDefaultCloseOperation(int)](http://docs.google.com/javax/swing/JFrame.html#setDefaultCloseOperation(int)). To make the JFrame behave the same as a Frame instance, use setDefaultCloseOperation(WindowConstants.DO\_NOTHING\_ON\_CLOSE).

For more information on content panes and other features that root panes provide, see [Using Top-Level Containers](http://java.sun.com/docs/books/tutorial/uiswing/components/toplevel.html) in *The Java Tutorial*.

In a multi-screen environment, you can create a JFrame on a different screen device. See [Frame](http://docs.google.com/java/awt/Frame.html) for more information.

**Warning:** Swing is not thread safe. For more information see [Swing's Threading Policy](http://docs.google.com/package-summary.html#threading).

**Warning:** Serialized objects of this class will not be compatible with future Swing releases. The current serialization support is appropriate for short term storage or RMI between applications running the same version of Swing. As of 1.4, support for long term storage of all JavaBeansTM has been added to the java.beans package. Please see [XMLEncoder](http://docs.google.com/java/beans/XMLEncoder.html).

**See Also:**[JRootPane](http://docs.google.com/javax/swing/JRootPane.html), [setDefaultCloseOperation(int)](http://docs.google.com/javax/swing/JFrame.html#setDefaultCloseOperation(int)), [WindowListener.windowClosing(java.awt.event.WindowEvent)](http://docs.google.com/java/awt/event/WindowListener.html#windowClosing(java.awt.event.WindowEvent)), [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html)

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**JFrame.AccessibleJFrame**](http://docs.google.com/javax/swing/JFrame.AccessibleJFrame.html)            This class implements accessibility support for the JFrame class. |

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

| **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** | |
| --- | --- |
| protected  [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**accessibleContext**](http://docs.google.com/javax/swing/JFrame.html#accessibleContext)            The accessible context property. |
| static int | [**EXIT\_ON\_CLOSE**](http://docs.google.com/javax/swing/JFrame.html#EXIT_ON_CLOSE)            The exit application default window close operation. |
| protected  [JRootPane](http://docs.google.com/javax/swing/JRootPane.html) | [**rootPane**](http://docs.google.com/javax/swing/JFrame.html#rootPane)            The JRootPane instance that manages the contentPane and optional menuBar for this frame, as well as the glassPane. |
| protected  boolean | [**rootPaneCheckingEnabled**](http://docs.google.com/javax/swing/JFrame.html#rootPaneCheckingEnabled)            If true then calls to add and setLayout will be forwarded to the contentPane. |

| **Fields inherited from class java.awt.**[**Frame**](http://docs.google.com/java/awt/Frame.html) |
| --- |
| [CROSSHAIR\_CURSOR](http://docs.google.com/java/awt/Frame.html#CROSSHAIR_CURSOR), [DEFAULT\_CURSOR](http://docs.google.com/java/awt/Frame.html#DEFAULT_CURSOR), [E\_RESIZE\_CURSOR](http://docs.google.com/java/awt/Frame.html#E_RESIZE_CURSOR), [HAND\_CURSOR](http://docs.google.com/java/awt/Frame.html#HAND_CURSOR), [ICONIFIED](http://docs.google.com/java/awt/Frame.html#ICONIFIED), [MAXIMIZED\_BOTH](http://docs.google.com/java/awt/Frame.html#MAXIMIZED_BOTH), [MAXIMIZED\_HORIZ](http://docs.google.com/java/awt/Frame.html#MAXIMIZED_HORIZ), [MAXIMIZED\_VERT](http://docs.google.com/java/awt/Frame.html#MAXIMIZED_VERT), [MOVE\_CURSOR](http://docs.google.com/java/awt/Frame.html#MOVE_CURSOR), [N\_RESIZE\_CURSOR](http://docs.google.com/java/awt/Frame.html#N_RESIZE_CURSOR), [NE\_RESIZE\_CURSOR](http://docs.google.com/java/awt/Frame.html#NE_RESIZE_CURSOR), [NORMAL](http://docs.google.com/java/awt/Frame.html#NORMAL), [NW\_RESIZE\_CURSOR](http://docs.google.com/java/awt/Frame.html#NW_RESIZE_CURSOR), [S\_RESIZE\_CURSOR](http://docs.google.com/java/awt/Frame.html#S_RESIZE_CURSOR), [SE\_RESIZE\_CURSOR](http://docs.google.com/java/awt/Frame.html#SE_RESIZE_CURSOR), [SW\_RESIZE\_CURSOR](http://docs.google.com/java/awt/Frame.html#SW_RESIZE_CURSOR), [TEXT\_CURSOR](http://docs.google.com/java/awt/Frame.html#TEXT_CURSOR), [W\_RESIZE\_CURSOR](http://docs.google.com/java/awt/Frame.html#W_RESIZE_CURSOR), [WAIT\_CURSOR](http://docs.google.com/java/awt/Frame.html#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 javax.swing.**[**WindowConstants**](http://docs.google.com/javax/swing/WindowConstants.html) |
| --- |
| [DISPOSE\_ON\_CLOSE](http://docs.google.com/javax/swing/WindowConstants.html#DISPOSE_ON_CLOSE), [DO\_NOTHING\_ON\_CLOSE](http://docs.google.com/javax/swing/WindowConstants.html#DO_NOTHING_ON_CLOSE), [HIDE\_ON\_CLOSE](http://docs.google.com/javax/swing/WindowConstants.html#HIDE_ON_CLOSE) |

| **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** | |
| --- | --- |
| [**JFrame**](http://docs.google.com/javax/swing/JFrame.html#JFrame())()            Constructs a new frame that is initially invisible. |
| [**JFrame**](http://docs.google.com/javax/swing/JFrame.html#JFrame(java.awt.GraphicsConfiguration))([GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html) gc)            Creates a Frame in the specified GraphicsConfiguration of a screen device and a blank title. |
| [**JFrame**](http://docs.google.com/javax/swing/JFrame.html#JFrame(java.lang.String))([String](http://docs.google.com/java/lang/String.html) title)            Creates a new, initially invisible Frame with the specified title. |
| [**JFrame**](http://docs.google.com/javax/swing/JFrame.html#JFrame(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)            Creates a JFrame with the specified title and the specified GraphicsConfiguration of a screen device. |

| **Method Summary** | |
| --- | --- |
| protected  void | [**addImpl**](http://docs.google.com/javax/swing/JFrame.html#addImpl(java.awt.Component,%20java.lang.Object,%20int))([Component](http://docs.google.com/java/awt/Component.html) comp, [Object](http://docs.google.com/java/lang/Object.html) constraints, int index)            Adds the specified child Component. |
| protected  [JRootPane](http://docs.google.com/javax/swing/JRootPane.html) | [**createRootPane**](http://docs.google.com/javax/swing/JFrame.html#createRootPane())()            Called by the constructor methods to create the default rootPane. |
| protected  void | [**frameInit**](http://docs.google.com/javax/swing/JFrame.html#frameInit())()            Called by the constructors to init the JFrame properly. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JFrame.html#getAccessibleContext())()            Gets the AccessibleContext associated with this JFrame. |
| [Container](http://docs.google.com/java/awt/Container.html) | [**getContentPane**](http://docs.google.com/javax/swing/JFrame.html#getContentPane())()            Returns the contentPane object for this frame. |
| int | [**getDefaultCloseOperation**](http://docs.google.com/javax/swing/JFrame.html#getDefaultCloseOperation())()            Returns the operation that occurs when the user initiates a "close" on this frame. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getGlassPane**](http://docs.google.com/javax/swing/JFrame.html#getGlassPane())()            Returns the glassPane object for this frame. |
| [Graphics](http://docs.google.com/java/awt/Graphics.html) | [**getGraphics**](http://docs.google.com/javax/swing/JFrame.html#getGraphics())()            Creates a graphics context for this component. |
| [JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) | [**getJMenuBar**](http://docs.google.com/javax/swing/JFrame.html#getJMenuBar())()            Returns the menubar set on this frame. |
| [JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html) | [**getLayeredPane**](http://docs.google.com/javax/swing/JFrame.html#getLayeredPane())()            Returns the layeredPane object for this frame. |
| [JRootPane](http://docs.google.com/javax/swing/JRootPane.html) | [**getRootPane**](http://docs.google.com/javax/swing/JFrame.html#getRootPane())()            Returns the rootPane object for this frame. |
| [TransferHandler](http://docs.google.com/javax/swing/TransferHandler.html) | [**getTransferHandler**](http://docs.google.com/javax/swing/JFrame.html#getTransferHandler())()            Gets the transferHandler property. |
| static boolean | [**isDefaultLookAndFeelDecorated**](http://docs.google.com/javax/swing/JFrame.html#isDefaultLookAndFeelDecorated())()            Returns true if newly created JFrames should have their Window decorations provided by the current look and feel. |
| protected  boolean | [**isRootPaneCheckingEnabled**](http://docs.google.com/javax/swing/JFrame.html#isRootPaneCheckingEnabled())()            Returns whether calls to add and setLayout are forwarded to the contentPane. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/JFrame.html#paramString())()            Returns a string representation of this JFrame. |
| protected  void | [**processWindowEvent**](http://docs.google.com/javax/swing/JFrame.html#processWindowEvent(java.awt.event.WindowEvent))([WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html) e)            Processes window events occurring on this component. |
| void | [**remove**](http://docs.google.com/javax/swing/JFrame.html#remove(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) comp)            Removes the specified component from the container. |
| void | [**repaint**](http://docs.google.com/javax/swing/JFrame.html#repaint(long,%20int,%20int,%20int,%20int))(long time, int x, int y, int width, int height)            Repaints the specified rectangle of this component within time milliseconds. |
| void | [**setContentPane**](http://docs.google.com/javax/swing/JFrame.html#setContentPane(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) contentPane)            Sets the contentPane property. |
| void | [**setDefaultCloseOperation**](http://docs.google.com/javax/swing/JFrame.html#setDefaultCloseOperation(int))(int operation)            Sets the operation that will happen by default when the user initiates a "close" on this frame. |
| static void | [**setDefaultLookAndFeelDecorated**](http://docs.google.com/javax/swing/JFrame.html#setDefaultLookAndFeelDecorated(boolean))(boolean defaultLookAndFeelDecorated)            Provides a hint as to whether or not newly created JFrames should have their Window decorations (such as borders, widgets to close the window, title...) provided by the current look and feel. |
| void | [**setGlassPane**](http://docs.google.com/javax/swing/JFrame.html#setGlassPane(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) glassPane)            Sets the glassPane property. |
| void | [**setIconImage**](http://docs.google.com/javax/swing/JFrame.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 | [**setJMenuBar**](http://docs.google.com/javax/swing/JFrame.html#setJMenuBar(javax.swing.JMenuBar))([JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) menubar)            Sets the menubar for this frame. |
| void | [**setLayeredPane**](http://docs.google.com/javax/swing/JFrame.html#setLayeredPane(javax.swing.JLayeredPane))([JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html) layeredPane)            Sets the layeredPane property. |
| void | [**setLayout**](http://docs.google.com/javax/swing/JFrame.html#setLayout(java.awt.LayoutManager))([LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) manager)            Sets the LayoutManager. |
| protected  void | [**setRootPane**](http://docs.google.com/javax/swing/JFrame.html#setRootPane(javax.swing.JRootPane))([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) root)            Sets the rootPane property. |
| protected  void | [**setRootPaneCheckingEnabled**](http://docs.google.com/javax/swing/JFrame.html#setRootPaneCheckingEnabled(boolean))(boolean enabled)            Sets whether calls to add and setLayout are forwarded to the contentPane. |
| void | [**setTransferHandler**](http://docs.google.com/javax/swing/JFrame.html#setTransferHandler(javax.swing.TransferHandler))([TransferHandler](http://docs.google.com/javax/swing/TransferHandler.html) newHandler)            Sets the transferHandler property, which is a mechanism to support transfer of data into this component. |
| void | [**update**](http://docs.google.com/javax/swing/JFrame.html#update(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Just calls paint(g). |

| **Methods inherited from class java.awt.**[**Frame**](http://docs.google.com/java/awt/Frame.html) |
| --- |
| [addNotify](http://docs.google.com/java/awt/Frame.html#addNotify()), [getCursorType](http://docs.google.com/java/awt/Frame.html#getCursorType()), [getExtendedState](http://docs.google.com/java/awt/Frame.html#getExtendedState()), [getFrames](http://docs.google.com/java/awt/Frame.html#getFrames()), [getIconImage](http://docs.google.com/java/awt/Frame.html#getIconImage()), [getMaximizedBounds](http://docs.google.com/java/awt/Frame.html#getMaximizedBounds()), [getMenuBar](http://docs.google.com/java/awt/Frame.html#getMenuBar()), [getState](http://docs.google.com/java/awt/Frame.html#getState()), [getTitle](http://docs.google.com/java/awt/Frame.html#getTitle()), [isResizable](http://docs.google.com/java/awt/Frame.html#isResizable()), [isUndecorated](http://docs.google.com/java/awt/Frame.html#isUndecorated()), [remove](http://docs.google.com/java/awt/Frame.html#remove(java.awt.MenuComponent)), [removeNotify](http://docs.google.com/java/awt/Frame.html#removeNotify()), [setCursor](http://docs.google.com/java/awt/Frame.html#setCursor(int)), [setExtendedState](http://docs.google.com/java/awt/Frame.html#setExtendedState(int)), [setMaximizedBounds](http://docs.google.com/java/awt/Frame.html#setMaximizedBounds(java.awt.Rectangle)), [setMenuBar](http://docs.google.com/java/awt/Frame.html#setMenuBar(java.awt.MenuBar)), [setResizable](http://docs.google.com/java/awt/Frame.html#setResizable(boolean)), [setState](http://docs.google.com/java/awt/Frame.html#setState(int)), [setTitle](http://docs.google.com/java/awt/Frame.html#setTitle(java.lang.String)), [setUndecorated](http://docs.google.com/java/awt/Frame.html#setUndecorated(boolean)) |

| **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)), [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)), [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(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)), [transferFocusBackward](http://docs.google.com/java/awt/Container.html#transferFocusBackward()), [transferFocusDownCycle](http://docs.google.com/java/awt/Container.html#transferFocusDownCycle()), [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()), [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)), [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** |
| --- |

### EXIT\_ON\_CLOSE

public static final int **EXIT\_ON\_CLOSE**

The exit application default window close operation. If a window has this set as the close operation and is closed in an applet, a SecurityException may be thrown. It is recommended you only use this in an application.

**Since:** 1.3 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#javax.swing.JFrame.EXIT_ON_CLOSE)

### rootPane

protected [JRootPane](http://docs.google.com/javax/swing/JRootPane.html) **rootPane**

The JRootPane instance that manages the contentPane and optional menuBar for this frame, as well as the glassPane.

**See Also:**[JRootPane](http://docs.google.com/javax/swing/JRootPane.html), [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html)

### rootPaneCheckingEnabled

protected boolean **rootPaneCheckingEnabled**

If true then calls to add and setLayout will be forwarded to the contentPane. This is initially false, but is set to true when the JFrame is constructed.

**See Also:**[isRootPaneCheckingEnabled()](http://docs.google.com/javax/swing/JFrame.html#isRootPaneCheckingEnabled()), [setRootPaneCheckingEnabled(boolean)](http://docs.google.com/javax/swing/JFrame.html#setRootPaneCheckingEnabled(boolean)), [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html)

### accessibleContext

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

The accessible context property.

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

### JFrame

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

Constructs a new frame that is initially invisible.

This constructor sets the component's locale property to the value returned by JComponent.getDefaultLocale.

**Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if 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)), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JFrame

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

Creates a Frame in the specified GraphicsConfiguration of a screen device and a blank title.

This constructor sets the component's locale property to the value returned by JComponent.getDefaultLocale.

**Parameters:**gc - the GraphicsConfiguration that is used to construct the new Frame; 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. This exception is always thrown when GraphicsEnvironment.isHeadless() returns true.**Since:** 1.3 **See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JFrame

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

Creates a new, initially invisible Frame with the specified title.

This constructor sets the component's locale property to the value returned by JComponent.getDefaultLocale.

**Parameters:**title - the title for the frame **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if 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)), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JFrame

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

Creates a JFrame with the specified title and the specified GraphicsConfiguration of a screen device.

This constructor sets the component's locale property to the value returned by JComponent.getDefaultLocale.

**Parameters:**title - the title to be displayed in the frame's border. A null value is treated as an empty string, "".gc - the GraphicsConfiguration that is used to construct the new JFrame with; 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. This exception is always thrown when GraphicsEnvironment.isHeadless() returns true.**Since:** 1.3 **See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

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

### frameInit

protected void **frameInit**()

Called by the constructors to init the JFrame properly.

### createRootPane

protected [JRootPane](http://docs.google.com/javax/swing/JRootPane.html) **createRootPane**()

Called by the constructor methods to create the default rootPane.

### processWindowEvent

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

Processes window events occurring on this component. Hides the window or disposes of it, as specified by the setting of the defaultCloseOperation property.

**Overrides:**[processWindowEvent](http://docs.google.com/java/awt/Window.html#processWindowEvent(java.awt.event.WindowEvent)) in class [Window](http://docs.google.com/java/awt/Window.html) **Parameters:**e - the window event**See Also:**[setDefaultCloseOperation(int)](http://docs.google.com/javax/swing/JFrame.html#setDefaultCloseOperation(int)), [Window.processWindowEvent(java.awt.event.WindowEvent)](http://docs.google.com/java/awt/Window.html#processWindowEvent(java.awt.event.WindowEvent))

### setDefaultCloseOperation

public void **setDefaultCloseOperation**(int operation)

Sets the operation that will happen by default when the user initiates a "close" on this frame. You must specify one of the following choices:

* DO\_NOTHING\_ON\_CLOSE (defined in WindowConstants): Don't do anything; require the program to handle the operation in the windowClosing method of a registered WindowListener object.
* HIDE\_ON\_CLOSE (defined in WindowConstants): Automatically hide the frame after invoking any registered WindowListener objects.
* DISPOSE\_ON\_CLOSE (defined in WindowConstants): Automatically hide and dispose the frame after invoking any registered WindowListener objects.
* EXIT\_ON\_CLOSE (defined in JFrame): Exit the application using the System exit method. Use this only in applications.

The value is set to HIDE\_ON\_CLOSE by default. Changes to the value of this property cause the firing of a property change event, with property name "defaultCloseOperation".

**Note**: When the last displayable window within the Java virtual machine (VM) is disposed of, the VM may terminate. See  [AWT Threading Issues](http://docs.google.com/java/awt/doc-files/AWTThreadIssues.html) for more information.

**Parameters:**operation - the operation which should be performed when the user closes the frame **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if defaultCloseOperation value isn't one of the above valid values [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if EXIT\_ON\_CLOSE has been specified and the SecurityManager will not allow the caller to invoke System.exit**See Also:**[Window.addWindowListener(java.awt.event.WindowListener)](http://docs.google.com/java/awt/Window.html#addWindowListener(java.awt.event.WindowListener)), [getDefaultCloseOperation()](http://docs.google.com/javax/swing/JFrame.html#getDefaultCloseOperation()), [WindowConstants](http://docs.google.com/javax/swing/WindowConstants.html), [Runtime.exit(int)](http://docs.google.com/java/lang/Runtime.html#exit(int))

### getDefaultCloseOperation

public int **getDefaultCloseOperation**()

Returns the operation that occurs when the user initiates a "close" on this frame.

**Returns:**an integer indicating the window-close operation**See Also:**[setDefaultCloseOperation(int)](http://docs.google.com/javax/swing/JFrame.html#setDefaultCloseOperation(int))

### setTransferHandler

public void **setTransferHandler**([TransferHandler](http://docs.google.com/javax/swing/TransferHandler.html) newHandler)

Sets the transferHandler property, which is a mechanism to support transfer of data into this component. Use null if the component does not support data transfer operations.

If the system property suppressSwingDropSupport is false (the default) and the current drop target on this component is either null or not a user-set drop target, this method will change the drop target as follows: If newHandler is null it will clear the drop target. If not null it will install a new DropTarget.

Note: When used with JFrame, TransferHandler only provides data import capability, as the data export related methods are currently typed to JComponent.

Please see  [How to Use Drag and Drop and Data Transfer](http://java.sun.com/docs/books/tutorial/uiswing/misc/dnd.html), a section in *The Java Tutorial*, for more information.

**Parameters:**newHandler - the new TransferHandler**Since:** 1.6 **See Also:**[TransferHandler](http://docs.google.com/javax/swing/TransferHandler.html), [getTransferHandler()](http://docs.google.com/javax/swing/JFrame.html#getTransferHandler()), [Component.setDropTarget(java.awt.dnd.DropTarget)](http://docs.google.com/java/awt/Component.html#setDropTarget(java.awt.dnd.DropTarget))

### getTransferHandler

public [TransferHandler](http://docs.google.com/javax/swing/TransferHandler.html) **getTransferHandler**()

Gets the transferHandler property.

**Returns:**the value of the transferHandler property**Since:** 1.6 **See Also:**[TransferHandler](http://docs.google.com/javax/swing/TransferHandler.html), [setTransferHandler(javax.swing.TransferHandler)](http://docs.google.com/javax/swing/JFrame.html#setTransferHandler(javax.swing.TransferHandler))

### update

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

Just calls paint(g). This method was overridden to prevent an unnecessary call to clear the background.

**Overrides:**[update](http://docs.google.com/java/awt/Container.html#update(java.awt.Graphics)) in class [Container](http://docs.google.com/java/awt/Container.html) **Parameters:**g - the Graphics context in which to paint**See Also:**[Component.update(Graphics)](http://docs.google.com/java/awt/Component.html#update(java.awt.Graphics))

### setJMenuBar

public void **setJMenuBar**([JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) menubar)

Sets the menubar for this frame.

**Parameters:**menubar - the menubar being placed in the frame**See Also:**[getJMenuBar()](http://docs.google.com/javax/swing/JFrame.html#getJMenuBar())

### getJMenuBar

public [JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) **getJMenuBar**()

Returns the menubar set on this frame.

**Returns:**the menubar for this frame**See Also:**[setJMenuBar(javax.swing.JMenuBar)](http://docs.google.com/javax/swing/JFrame.html#setJMenuBar(javax.swing.JMenuBar))

### isRootPaneCheckingEnabled

protected boolean **isRootPaneCheckingEnabled**()

Returns whether calls to add and setLayout are forwarded to the contentPane.

**Returns:**true if add and setLayout are fowarded; false otherwise**See Also:**[addImpl(java.awt.Component, java.lang.Object, int)](http://docs.google.com/javax/swing/JFrame.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [setLayout(java.awt.LayoutManager)](http://docs.google.com/javax/swing/JFrame.html#setLayout(java.awt.LayoutManager)), [setRootPaneCheckingEnabled(boolean)](http://docs.google.com/javax/swing/JFrame.html#setRootPaneCheckingEnabled(boolean)), [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html)

### setRootPaneCheckingEnabled

protected void **setRootPaneCheckingEnabled**(boolean enabled)

Sets whether calls to add and setLayout are forwarded to the contentPane.

**Parameters:**enabled - true if add and setLayout are forwarded, false if they should operate directly on the JFrame.**See Also:**[addImpl(java.awt.Component, java.lang.Object, int)](http://docs.google.com/javax/swing/JFrame.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [setLayout(java.awt.LayoutManager)](http://docs.google.com/javax/swing/JFrame.html#setLayout(java.awt.LayoutManager)), [isRootPaneCheckingEnabled()](http://docs.google.com/javax/swing/JFrame.html#isRootPaneCheckingEnabled()), [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html)

### addImpl

protected void **addImpl**([Component](http://docs.google.com/java/awt/Component.html) comp,  
 [Object](http://docs.google.com/java/lang/Object.html) constraints,  
 int index)

Adds the specified child Component. This method is overridden to conditionally forward calls to the contentPane. By default, children are added to the contentPane instead of the frame, refer to [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html) for details.

**Overrides:**[addImpl](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)) in class [Container](http://docs.google.com/java/awt/Container.html) **Parameters:**comp - the component to be enhancedconstraints - the constraints to be respectedindex - the index **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if index is invalid [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if adding the container's parent to itself [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if adding a window to a container**See Also:**[setRootPaneCheckingEnabled(boolean)](http://docs.google.com/javax/swing/JFrame.html#setRootPaneCheckingEnabled(boolean)), [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html)

### remove

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

Removes the specified component from the container. If comp is not the rootPane, this will forward the call to the contentPane. This will do nothing if comp is not a child of the JFrame or contentPane.

**Overrides:**[remove](http://docs.google.com/java/awt/Container.html#remove(java.awt.Component)) in class [Container](http://docs.google.com/java/awt/Container.html) **Parameters:**comp - the component to be removed **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if comp is null**See Also:**[Container.add(java.awt.Component)](http://docs.google.com/java/awt/Container.html#add(java.awt.Component)), [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html)

### setLayout

public void **setLayout**([LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) manager)

Sets the LayoutManager. Overridden to conditionally forward the call to the contentPane. Refer to [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html) for more information.

**Overrides:**[setLayout](http://docs.google.com/java/awt/Container.html#setLayout(java.awt.LayoutManager)) in class [Container](http://docs.google.com/java/awt/Container.html) **Parameters:**manager - the LayoutManager**See Also:**[setRootPaneCheckingEnabled(boolean)](http://docs.google.com/javax/swing/JFrame.html#setRootPaneCheckingEnabled(boolean)), [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html)

### getRootPane

public [JRootPane](http://docs.google.com/javax/swing/JRootPane.html) **getRootPane**()

Returns the rootPane object for this frame.

**Specified by:**[getRootPane](http://docs.google.com/javax/swing/RootPaneContainer.html#getRootPane()) in interface [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html) **Returns:**the rootPane property**See Also:**[setRootPane(javax.swing.JRootPane)](http://docs.google.com/javax/swing/JFrame.html#setRootPane(javax.swing.JRootPane)), [RootPaneContainer.getRootPane()](http://docs.google.com/javax/swing/RootPaneContainer.html#getRootPane())

### setRootPane

protected void **setRootPane**([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) root)

Sets the rootPane property. This method is called by the constructor.

**Parameters:**root - the rootPane object for this frame**See Also:**[getRootPane()](http://docs.google.com/javax/swing/JFrame.html#getRootPane())

### 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/Frame.html#setIconImage(java.awt.Image)) in class [Frame](http://docs.google.com/java/awt/Frame.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())

### getContentPane

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

Returns the contentPane object for this frame.

**Specified by:**[getContentPane](http://docs.google.com/javax/swing/RootPaneContainer.html#getContentPane()) in interface [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html) **Returns:**the contentPane property**See Also:**[setContentPane(java.awt.Container)](http://docs.google.com/javax/swing/JFrame.html#setContentPane(java.awt.Container)), [RootPaneContainer.getContentPane()](http://docs.google.com/javax/swing/RootPaneContainer.html#getContentPane())

### setContentPane

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

Sets the contentPane property. This method is called by the constructor.

Swing's painting architecture requires an opaque JComponent in the containment hiearchy. This is typically provided by the content pane. If you replace the content pane it is recommended you replace it with an opaque JComponent.

**Specified by:**[setContentPane](http://docs.google.com/javax/swing/RootPaneContainer.html#setContentPane(java.awt.Container)) in interface [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html) **Parameters:**contentPane - the contentPane object for this frame **Throws:** [IllegalComponentStateException](http://docs.google.com/java/awt/IllegalComponentStateException.html) - (a runtime exception) if the content pane parameter is null**See Also:**[getContentPane()](http://docs.google.com/javax/swing/JFrame.html#getContentPane()), [RootPaneContainer.setContentPane(java.awt.Container)](http://docs.google.com/javax/swing/RootPaneContainer.html#setContentPane(java.awt.Container)), [JRootPane](http://docs.google.com/javax/swing/JRootPane.html)

### getLayeredPane

public [JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html) **getLayeredPane**()

Returns the layeredPane object for this frame.

**Specified by:**[getLayeredPane](http://docs.google.com/javax/swing/RootPaneContainer.html#getLayeredPane()) in interface [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html) **Returns:**the layeredPane property**See Also:**[setLayeredPane(javax.swing.JLayeredPane)](http://docs.google.com/javax/swing/JFrame.html#setLayeredPane(javax.swing.JLayeredPane)), [RootPaneContainer.getLayeredPane()](http://docs.google.com/javax/swing/RootPaneContainer.html#getLayeredPane())

### setLayeredPane

public void **setLayeredPane**([JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html) layeredPane)

Sets the layeredPane property. This method is called by the constructor.

**Specified by:**[setLayeredPane](http://docs.google.com/javax/swing/RootPaneContainer.html#setLayeredPane(javax.swing.JLayeredPane)) in interface [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html) **Parameters:**layeredPane - the layeredPane object for this frame **Throws:** [IllegalComponentStateException](http://docs.google.com/java/awt/IllegalComponentStateException.html) - (a runtime exception) if the layered pane parameter is null**See Also:**[getLayeredPane()](http://docs.google.com/javax/swing/JFrame.html#getLayeredPane()), [RootPaneContainer.setLayeredPane(javax.swing.JLayeredPane)](http://docs.google.com/javax/swing/RootPaneContainer.html#setLayeredPane(javax.swing.JLayeredPane))

### getGlassPane

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

Returns the glassPane object for this frame.

**Specified by:**[getGlassPane](http://docs.google.com/javax/swing/RootPaneContainer.html#getGlassPane()) in interface [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html) **Returns:**the glassPane property**See Also:**[setGlassPane(java.awt.Component)](http://docs.google.com/javax/swing/JFrame.html#setGlassPane(java.awt.Component)), [RootPaneContainer.getGlassPane()](http://docs.google.com/javax/swing/RootPaneContainer.html#getGlassPane())

### setGlassPane

public void **setGlassPane**([Component](http://docs.google.com/java/awt/Component.html) glassPane)

Sets the glassPane property. This method is called by the constructor.

**Specified by:**[setGlassPane](http://docs.google.com/javax/swing/RootPaneContainer.html#setGlassPane(java.awt.Component)) in interface [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html) **Parameters:**glassPane - the glassPane object for this frame**See Also:**[getGlassPane()](http://docs.google.com/javax/swing/JFrame.html#getGlassPane()), [RootPaneContainer.setGlassPane(java.awt.Component)](http://docs.google.com/javax/swing/RootPaneContainer.html#setGlassPane(java.awt.Component))

### 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.

**Overrides:**[getGraphics](http://docs.google.com/java/awt/Component.html#getGraphics()) in class [Component](http://docs.google.com/java/awt/Component.html) **Returns:**a graphics context for this component, or null if it has none**Since:** 1.6 **See Also:**[Component.paint(java.awt.Graphics)](http://docs.google.com/java/awt/Component.html#paint(java.awt.Graphics))

### repaint

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

Repaints the specified rectangle of this component within time milliseconds. Refer to RepaintManager for details on how the repaint is handled.

**Overrides:**[repaint](http://docs.google.com/java/awt/Component.html#repaint(long,%20int,%20int,%20int,%20int)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**time - maximum time in milliseconds before updatex - the *x* coordinatey - the *y* coordinatewidth - the widthheight - the height**Since:** 1.6 **See Also:**[RepaintManager](http://docs.google.com/javax/swing/RepaintManager.html)

### setDefaultLookAndFeelDecorated

public static void **setDefaultLookAndFeelDecorated**(boolean defaultLookAndFeelDecorated)

Provides a hint as to whether or not newly created JFrames should have their Window decorations (such as borders, widgets to close the window, title...) provided by the current look and feel. If defaultLookAndFeelDecorated is true, the current LookAndFeel supports providing window decorations, and the current window manager supports undecorated windows, then newly created JFrames will have their Window decorations provided by the current LookAndFeel. Otherwise, newly created JFrames will have their Window decorations provided by the current window manager.

You can get the same effect on a single JFrame by doing the following:

JFrame frame = new JFrame();  
 frame.setUndecorated(true);  
 frame.getRootPane().setWindowDecorationStyle(JRootPane.FRAME);

**Parameters:**defaultLookAndFeelDecorated - A hint as to whether or not current look and feel should provide window decorations**Since:** 1.4 **See Also:**[LookAndFeel.getSupportsWindowDecorations()](http://docs.google.com/javax/swing/LookAndFeel.html#getSupportsWindowDecorations())

### isDefaultLookAndFeelDecorated

public static boolean **isDefaultLookAndFeelDecorated**()

Returns true if newly created JFrames should have their Window decorations provided by the current look and feel. This is only a hint, as certain look and feels may not support this feature.

**Returns:**true if look and feel should provide Window decorations.**Since:** 1.4

### paramString

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

Returns a string representation of this JFrame. 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/Frame.html#paramString()) in class [Frame](http://docs.google.com/java/awt/Frame.html) **Returns:**a string representation of this JFrame

### getAccessibleContext

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

Gets the AccessibleContext associated with this JFrame. For JFrames, the AccessibleContext takes the form of an AccessibleJFrame. A new AccessibleJFrame 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/Frame.html#getAccessibleContext()) in class [Frame](http://docs.google.com/java/awt/Frame.html) **Returns:**an AccessibleJFrame that serves as the AccessibleContext of this JFrame

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JFrame.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/javax/swing/JFormattedTextField.AbstractFormatterFactory.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JFrame.AccessibleJFrame.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JFrame.html)    [**NO FRAMES**](http://docs.google.com/JFrame.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#4d34og8) | [CONSTR](#lnxbz9) | [METHOD](#35nkun2) | DETAIL: [FIELD](#4i7ojhp) | [CONSTR](#qsh70q) | [METHOD](#147n2zr) |

[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).