| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JDialog.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/JDesktopPane.AccessibleJDesktopPane.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JDialog.AccessibleJDialog.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JDialog.html)    [**NO FRAMES**](http://docs.google.com/JDialog.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#4d34og8) | [CONSTR](#lnxbz9) | [METHOD](#35nkun2) | DETAIL: [FIELD](#1y810tw) | [CONSTR](#3whwml4) | [METHOD](#1v1yuxt) |

## **javax.swing**

Class JDialog

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

**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 **JDialog**extends [Dialog](http://docs.google.com/java/awt/Dialog.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)

The main class for creating a dialog window. You can use this class to create a custom dialog, or invoke the many class methods in [JOptionPane](http://docs.google.com/javax/swing/JOptionPane.html) to create a variety of standard dialogs. For information about creating dialogs, see *The Java Tutorial* section [How to Make Dialogs](http://java.sun.com/docs/books/tutorial/uiswing/components/dialog.html).

The JDialog component contains a JRootPane as its only child. The contentPane should be the parent of any children of the JDialog. As a convenience add and its variants, remove and setLayout have been overridden to forward to the contentPane as necessary. This means you can write:

dialog.add(child);

And the child will be added to the contentPane. The contentPane is always non-null. Attempting to set it to null generates an exception. The default contentPane has 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 JDialog.

Please see the JRootPane documentation for a complete description of the contentPane, glassPane, and layeredPane components.

In a multi-screen environment, you can create a JDialog on a different screen device than its owner. 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:**[JOptionPane](http://docs.google.com/javax/swing/JOptionPane.html), [JRootPane](http://docs.google.com/javax/swing/JRootPane.html), [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html)

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

| **Nested classes/interfaces inherited from class java.awt.**[**Dialog**](http://docs.google.com/java/awt/Dialog.html) |
| --- |
| [Dialog.AccessibleAWTDialog](http://docs.google.com/java/awt/Dialog.AccessibleAWTDialog.html), [Dialog.ModalExclusionType](http://docs.google.com/java/awt/Dialog.ModalExclusionType.html), [Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.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/JDialog.html#accessibleContext) |
| protected  [JRootPane](http://docs.google.com/javax/swing/JRootPane.html) | [**rootPane**](http://docs.google.com/javax/swing/JDialog.html#rootPane) |
| protected  boolean | [**rootPaneCheckingEnabled**](http://docs.google.com/javax/swing/JDialog.html#rootPaneCheckingEnabled)            If true then calls to add and setLayout will be forwarded to the contentPane. |

| **Fields inherited from class java.awt.**[**Dialog**](http://docs.google.com/java/awt/Dialog.html) |
| --- |
| [DEFAULT\_MODALITY\_TYPE](http://docs.google.com/java/awt/Dialog.html#DEFAULT_MODALITY_TYPE) |

| **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), [EXIT\_ON\_CLOSE](http://docs.google.com/javax/swing/WindowConstants.html#EXIT_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** | |
| --- | --- |
| [**JDialog**](http://docs.google.com/javax/swing/JDialog.html#JDialog())()            Creates a modeless dialog without a title and without a specified Frame owner. |
| [**JDialog**](http://docs.google.com/javax/swing/JDialog.html#JDialog(java.awt.Dialog))([Dialog](http://docs.google.com/java/awt/Dialog.html) owner)            Creates a modeless dialog without a title with the specified Dialog as its owner. |
| [**JDialog**](http://docs.google.com/javax/swing/JDialog.html#JDialog(java.awt.Dialog,%20boolean))([Dialog](http://docs.google.com/java/awt/Dialog.html) owner, boolean modal)            Creates a dialog with the specified owner Dialog and modality. |
| [**JDialog**](http://docs.google.com/javax/swing/JDialog.html#JDialog(java.awt.Dialog,%20java.lang.String))([Dialog](http://docs.google.com/java/awt/Dialog.html) owner, [String](http://docs.google.com/java/lang/String.html) title)            Creates a modeless dialog with the specified title and with the specified owner dialog. |
| [**JDialog**](http://docs.google.com/javax/swing/JDialog.html#JDialog(java.awt.Dialog,%20java.lang.String,%20boolean))([Dialog](http://docs.google.com/java/awt/Dialog.html) owner, [String](http://docs.google.com/java/lang/String.html) title, boolean modal)            Creates a dialog with the specified title, modality and the specified owner Dialog. |
| [**JDialog**](http://docs.google.com/javax/swing/JDialog.html#JDialog(java.awt.Dialog,%20java.lang.String,%20boolean,%20java.awt.GraphicsConfiguration))([Dialog](http://docs.google.com/java/awt/Dialog.html) owner, [String](http://docs.google.com/java/lang/String.html) title, boolean modal, [GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html) gc)            Creates a dialog with the specified title, owner Dialog, modality and GraphicsConfiguration. |
| [**JDialog**](http://docs.google.com/javax/swing/JDialog.html#JDialog(java.awt.Frame))([Frame](http://docs.google.com/java/awt/Frame.html) owner)            Creates a modeless dialog without a title with the specified Frame as its owner. |
| [**JDialog**](http://docs.google.com/javax/swing/JDialog.html#JDialog(java.awt.Frame,%20boolean))([Frame](http://docs.google.com/java/awt/Frame.html) owner, boolean modal)            Creates a dialog with the specified owner Frame, modality and an empty title. |
| [**JDialog**](http://docs.google.com/javax/swing/JDialog.html#JDialog(java.awt.Frame,%20java.lang.String))([Frame](http://docs.google.com/java/awt/Frame.html) owner, [String](http://docs.google.com/java/lang/String.html) title)            Creates a modeless dialog with the specified title and with the specified owner frame. |
| [**JDialog**](http://docs.google.com/javax/swing/JDialog.html#JDialog(java.awt.Frame,%20java.lang.String,%20boolean))([Frame](http://docs.google.com/java/awt/Frame.html) owner, [String](http://docs.google.com/java/lang/String.html) title, boolean modal)            Creates a dialog with the specified title, owner Frame and modality. |
| [**JDialog**](http://docs.google.com/javax/swing/JDialog.html#JDialog(java.awt.Frame,%20java.lang.String,%20boolean,%20java.awt.GraphicsConfiguration))([Frame](http://docs.google.com/java/awt/Frame.html) owner, [String](http://docs.google.com/java/lang/String.html) title, boolean modal, [GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html) gc)            Creates a dialog with the specified title, owner Frame, modality and GraphicsConfiguration. |
| [**JDialog**](http://docs.google.com/javax/swing/JDialog.html#JDialog(java.awt.Window))([Window](http://docs.google.com/java/awt/Window.html) owner)            Creates a modeless dialog with the specified owner Window and an empty title. |
| [**JDialog**](http://docs.google.com/javax/swing/JDialog.html#JDialog(java.awt.Window,%20java.awt.Dialog.ModalityType))([Window](http://docs.google.com/java/awt/Window.html) owner, [Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html) modalityType)            Creates a dialog with the specified owner Window, modality and an empty title. |
| [**JDialog**](http://docs.google.com/javax/swing/JDialog.html#JDialog(java.awt.Window,%20java.lang.String))([Window](http://docs.google.com/java/awt/Window.html) owner, [String](http://docs.google.com/java/lang/String.html) title)            Creates a modeless dialog with the specified title and owner Window. |
| [**JDialog**](http://docs.google.com/javax/swing/JDialog.html#JDialog(java.awt.Window,%20java.lang.String,%20java.awt.Dialog.ModalityType))([Window](http://docs.google.com/java/awt/Window.html) owner, [String](http://docs.google.com/java/lang/String.html) title, [Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html) modalityType)            Creates a dialog with the specified title, owner Window and modality. |
| [**JDialog**](http://docs.google.com/javax/swing/JDialog.html#JDialog(java.awt.Window,%20java.lang.String,%20java.awt.Dialog.ModalityType,%20java.awt.GraphicsConfiguration))([Window](http://docs.google.com/java/awt/Window.html) owner, [String](http://docs.google.com/java/lang/String.html) title, [Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html) modalityType, [GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html) gc)            Creates a dialog with the specified title, owner Window, modality and GraphicsConfiguration. |

| **Method Summary** | |
| --- | --- |
| protected  void | [**addImpl**](http://docs.google.com/javax/swing/JDialog.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/JDialog.html#createRootPane())()            Called by the constructor methods to create the default rootPane. |
| protected  void | [**dialogInit**](http://docs.google.com/javax/swing/JDialog.html#dialogInit())()            Called by the constructors to init the JDialog properly. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JDialog.html#getAccessibleContext())()            Gets the AccessibleContext associated with this JDialog. |
| [Container](http://docs.google.com/java/awt/Container.html) | [**getContentPane**](http://docs.google.com/javax/swing/JDialog.html#getContentPane())()            Returns the contentPane object for this dialog. |
| int | [**getDefaultCloseOperation**](http://docs.google.com/javax/swing/JDialog.html#getDefaultCloseOperation())()            Returns the operation which occurs when the user initiates a "close" on this dialog. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getGlassPane**](http://docs.google.com/javax/swing/JDialog.html#getGlassPane())()            Returns the glassPane object for this dialog. |
| [Graphics](http://docs.google.com/java/awt/Graphics.html) | [**getGraphics**](http://docs.google.com/javax/swing/JDialog.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/JDialog.html#getJMenuBar())()            Returns the menubar set on this dialog. |
| [JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html) | [**getLayeredPane**](http://docs.google.com/javax/swing/JDialog.html#getLayeredPane())()            Returns the layeredPane object for this dialog. |
| [JRootPane](http://docs.google.com/javax/swing/JRootPane.html) | [**getRootPane**](http://docs.google.com/javax/swing/JDialog.html#getRootPane())()            Returns the rootPane object for this dialog. |
| [TransferHandler](http://docs.google.com/javax/swing/TransferHandler.html) | [**getTransferHandler**](http://docs.google.com/javax/swing/JDialog.html#getTransferHandler())()            Gets the transferHandler property. |
| static boolean | [**isDefaultLookAndFeelDecorated**](http://docs.google.com/javax/swing/JDialog.html#isDefaultLookAndFeelDecorated())()            Returns true if newly created JDialogs should have their Window decorations provided by the current look and feel. |
| protected  boolean | [**isRootPaneCheckingEnabled**](http://docs.google.com/javax/swing/JDialog.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/JDialog.html#paramString())()            Returns a string representation of this JDialog. |
| protected  void | [**processWindowEvent**](http://docs.google.com/javax/swing/JDialog.html#processWindowEvent(java.awt.event.WindowEvent))([WindowEvent](http://docs.google.com/java/awt/event/WindowEvent.html) e)            Handles window events depending on the state of the defaultCloseOperation property. |
| void | [**remove**](http://docs.google.com/javax/swing/JDialog.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/JDialog.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/JDialog.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/JDialog.html#setDefaultCloseOperation(int))(int operation)            Sets the operation that will happen by default when the user initiates a "close" on this dialog. |
| static void | [**setDefaultLookAndFeelDecorated**](http://docs.google.com/javax/swing/JDialog.html#setDefaultLookAndFeelDecorated(boolean))(boolean defaultLookAndFeelDecorated)            Provides a hint as to whether or not newly created JDialogs 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/JDialog.html#setGlassPane(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) glassPane)            Sets the glassPane property. |
| void | [**setJMenuBar**](http://docs.google.com/javax/swing/JDialog.html#setJMenuBar(javax.swing.JMenuBar))([JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) menu)            Sets the menubar for this dialog. |
| void | [**setLayeredPane**](http://docs.google.com/javax/swing/JDialog.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/JDialog.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/JDialog.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/JDialog.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/JDialog.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/JDialog.html#update(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Calls paint(g). |

| **Methods inherited from class java.awt.**[**Dialog**](http://docs.google.com/java/awt/Dialog.html) |
| --- |
| [addNotify](http://docs.google.com/java/awt/Dialog.html#addNotify()), [getModalityType](http://docs.google.com/java/awt/Dialog.html#getModalityType()), [getTitle](http://docs.google.com/java/awt/Dialog.html#getTitle()), [hide](http://docs.google.com/java/awt/Dialog.html#hide()), [isModal](http://docs.google.com/java/awt/Dialog.html#isModal()), [isResizable](http://docs.google.com/java/awt/Dialog.html#isResizable()), [isUndecorated](http://docs.google.com/java/awt/Dialog.html#isUndecorated()), [setModal](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [setModalityType](http://docs.google.com/java/awt/Dialog.html#setModalityType(java.awt.Dialog.ModalityType)), [setResizable](http://docs.google.com/java/awt/Dialog.html#setResizable(boolean)), [setTitle](http://docs.google.com/java/awt/Dialog.html#setTitle(java.lang.String)), [setUndecorated](http://docs.google.com/java/awt/Dialog.html#setUndecorated(boolean)), [setVisible](http://docs.google.com/java/awt/Dialog.html#setVisible(boolean)), [show](http://docs.google.com/java/awt/Dialog.html#show()), [toBack](http://docs.google.com/java/awt/Dialog.html#toBack()) |

| **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()), [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)), [removeNotify](http://docs.google.com/java/awt/Window.html#removeNotify()), [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)), [setIconImage](http://docs.google.com/java/awt/Window.html#setIconImage(java.awt.Image)), [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)), [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)), [remove](http://docs.google.com/java/awt/Component.html#remove(java.awt.MenuComponent)), [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)) |

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

### rootPane

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

**See Also:**[getRootPane()](http://docs.google.com/javax/swing/JDialog.html#getRootPane()), [setRootPane(javax.swing.JRootPane)](http://docs.google.com/javax/swing/JDialog.html#setRootPane(javax.swing.JRootPane))

### 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 JDialog is constructed.

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

### accessibleContext

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

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

### JDialog

public **JDialog**()

Creates a modeless dialog without a title and without a specified Frame owner. A shared, hidden frame will be set as the owner of the dialog.

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

NOTE: This constructor does not allow you to create an unowned JDialog. To create an unowned JDialog you must use either the JDialog(Window) or JDialog(Dialog) constructor with an argument of null.

**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()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JDialog

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

Creates a modeless dialog without a title with the specified Frame as its owner. If owner is null, a shared, hidden frame will be set as the owner of the dialog.

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

NOTE: This constructor does not allow you to create an unowned JDialog. To create an unowned JDialog you must use either the JDialog(Window) or JDialog(Dialog) constructor with an argument of null.

**Parameters:**owner - the Frame from which the dialog is displayed **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()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JDialog

public **JDialog**([Frame](http://docs.google.com/java/awt/Frame.html) owner,  
 boolean modal)

Creates a dialog with the specified owner Frame, modality and an empty title. If owner is null, a shared, hidden frame will be set as the owner of the dialog.

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

NOTE: This constructor does not allow you to create an unowned JDialog. To create an unowned JDialog you must use either the JDialog(Window) or JDialog(Dialog) constructor with an argument of null.

**Parameters:**owner - the Frame from which the dialog is displayedmodal - specifies whether dialog blocks user input to other top-level windows when shown. If true, the modality type property is set to DEFAULT\_MODALITY\_TYPE, otherwise the dialog is modeless. **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()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JDialog

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

Creates a modeless dialog with the specified title and with the specified owner frame. If owner is null, a shared, hidden frame will be set as the owner of the dialog.

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

NOTE: This constructor does not allow you to create an unowned JDialog. To create an unowned JDialog you must use either the JDialog(Window) or JDialog(Dialog) constructor with an argument of null.

**Parameters:**owner - the Frame from which the dialog is displayedtitle - the String to display in the dialog's title bar **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()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JDialog

public **JDialog**([Frame](http://docs.google.com/java/awt/Frame.html) owner,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 boolean modal)

Creates a dialog with the specified title, owner Frame and modality. If owner is null, a shared, hidden frame will be set as the owner of this dialog.

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

NOTE: Any popup components (JComboBox, JPopupMenu, JMenuBar) created within a modal dialog will be forced to be lightweight.

NOTE: This constructor does not allow you to create an unowned JDialog. To create an unowned JDialog you must use either the JDialog(Window) or JDialog(Dialog) constructor with an argument of null.

**Parameters:**owner - the Frame from which the dialog is displayedtitle - the String to display in the dialog's title barmodal - specifies whether dialog blocks user input to other top-level windows when shown. If true, the modality type property is set to DEFAULT\_MODALITY\_TYPE otherwise the dialog is modeless **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true.**See Also:**[Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html), [Dialog.ModalityType.MODELESS](http://docs.google.com/java/awt/Dialog.ModalityType.html#MODELESS), [Dialog.DEFAULT\_MODALITY\_TYPE](http://docs.google.com/java/awt/Dialog.html#DEFAULT_MODALITY_TYPE), [Dialog.setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [Dialog.setModalityType(java.awt.Dialog.ModalityType)](http://docs.google.com/java/awt/Dialog.html#setModalityType(java.awt.Dialog.ModalityType)), [GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JDialog

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

Creates a dialog with the specified title, owner Frame, modality and GraphicsConfiguration. If owner is null, a shared, hidden frame will be set as the owner of this dialog.

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

NOTE: Any popup components (JComboBox, JPopupMenu, JMenuBar) created within a modal dialog will be forced to be lightweight.

NOTE: This constructor does not allow you to create an unowned JDialog. To create an unowned JDialog you must use either the JDialog(Window) or JDialog(Dialog) constructor with an argument of null.

**Parameters:**owner - the Frame from which the dialog is displayedtitle - the String to display in the dialog's title barmodal - specifies whether dialog blocks user input to other top-level windows when shown. If true, the modality type property is set to DEFAULT\_MODALITY\_TYPE, otherwise the dialog is modeless.gc - the GraphicsConfiguration of the target screen device. If gc is null, the same GraphicsConfiguration as the owning Frame is used. **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true.**Since:** 1.4 **See Also:**[Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html), [Dialog.ModalityType.MODELESS](http://docs.google.com/java/awt/Dialog.ModalityType.html#MODELESS), [Dialog.DEFAULT\_MODALITY\_TYPE](http://docs.google.com/java/awt/Dialog.html#DEFAULT_MODALITY_TYPE), [Dialog.setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [Dialog.setModalityType(java.awt.Dialog.ModalityType)](http://docs.google.com/java/awt/Dialog.html#setModalityType(java.awt.Dialog.ModalityType)), [GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JDialog

public **JDialog**([Dialog](http://docs.google.com/java/awt/Dialog.html) owner)

Creates a modeless dialog without a title with the specified Dialog as its owner.

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

**Parameters:**owner - the owner Dialog from which the dialog is displayed or null if this dialog has no owner **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()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JDialog

public **JDialog**([Dialog](http://docs.google.com/java/awt/Dialog.html) owner,  
 boolean modal)

Creates a dialog with the specified owner Dialog and modality.

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

**Parameters:**owner - the owner Dialog from which the dialog is displayed or null if this dialog has no ownermodal - specifies whether dialog blocks user input to other top-level windows when shown. If true, the modality type property is set to DEFAULT\_MODALITY\_TYPE, otherwise the dialog is modeless. **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true.**See Also:**[Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html), [Dialog.ModalityType.MODELESS](http://docs.google.com/java/awt/Dialog.ModalityType.html#MODELESS), [Dialog.DEFAULT\_MODALITY\_TYPE](http://docs.google.com/java/awt/Dialog.html#DEFAULT_MODALITY_TYPE), [Dialog.setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [Dialog.setModalityType(java.awt.Dialog.ModalityType)](http://docs.google.com/java/awt/Dialog.html#setModalityType(java.awt.Dialog.ModalityType)), [GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JDialog

public **JDialog**([Dialog](http://docs.google.com/java/awt/Dialog.html) owner,  
 [String](http://docs.google.com/java/lang/String.html) title)

Creates a modeless dialog with the specified title and with the specified owner dialog.

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

**Parameters:**owner - the owner Dialog from which the dialog is displayed or null if this dialog has no ownertitle - the String to display in the dialog's title bar **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()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JDialog

public **JDialog**([Dialog](http://docs.google.com/java/awt/Dialog.html) owner,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 boolean modal)

Creates a dialog with the specified title, modality and the specified owner Dialog.

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

**Parameters:**owner - the owner Dialog from which the dialog is displayed or null if this dialog has no ownertitle - the String to display in the dialog's title barmodal - specifies whether dialog blocks user input to other top-level windows when shown. If true, the modality type property is set to DEFAULT\_MODALITY\_TYPE, otherwise the dialog is modeless **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true.**See Also:**[Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html), [Dialog.ModalityType.MODELESS](http://docs.google.com/java/awt/Dialog.ModalityType.html#MODELESS), [Dialog.DEFAULT\_MODALITY\_TYPE](http://docs.google.com/java/awt/Dialog.html#DEFAULT_MODALITY_TYPE), [Dialog.setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [Dialog.setModalityType(java.awt.Dialog.ModalityType)](http://docs.google.com/java/awt/Dialog.html#setModalityType(java.awt.Dialog.ModalityType)), [GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JDialog

public **JDialog**([Dialog](http://docs.google.com/java/awt/Dialog.html) owner,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 boolean modal,  
 [GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html) gc)

Creates a dialog with the specified title, owner Dialog, modality and GraphicsConfiguration.

NOTE: Any popup components (JComboBox, JPopupMenu, JMenuBar) created within a modal dialog will be forced to be lightweight.

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

**Parameters:**owner - the owner Dialog from which the dialog is displayed or null if this dialog has no ownertitle - the String to display in the dialog's title barmodal - specifies whether dialog blocks user input to other top-level windows when shown. If true, the modality type property is set to DEFAULT\_MODALITY\_TYPE, otherwise the dialog is modelessgc - the GraphicsConfiguration of the target screen device. If gc is null, the same GraphicsConfiguration as the owning Dialog is used. **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true.**Since:** 1.4 **See Also:**[Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html), [Dialog.ModalityType.MODELESS](http://docs.google.com/java/awt/Dialog.ModalityType.html#MODELESS), [Dialog.DEFAULT\_MODALITY\_TYPE](http://docs.google.com/java/awt/Dialog.html#DEFAULT_MODALITY_TYPE), [Dialog.setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [Dialog.setModalityType(java.awt.Dialog.ModalityType)](http://docs.google.com/java/awt/Dialog.html#setModalityType(java.awt.Dialog.ModalityType)), [GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JDialog

public **JDialog**([Window](http://docs.google.com/java/awt/Window.html) owner)

Creates a modeless dialog with the specified owner Window and an empty title.

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

**Parameters:**owner - the Window from which the dialog is displayed or null if this dialog has no owner **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true**Since:** 1.6 **See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JDialog

public **JDialog**([Window](http://docs.google.com/java/awt/Window.html) owner,  
 [Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html) modalityType)

Creates a dialog with the specified owner Window, modality and an empty title.

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

**Parameters:**owner - the Window from which the dialog is displayed or null if this dialog has no ownermodalityType - specifies whether dialog blocks input to other windows when shown. null value and unsupported modality types are equivalent to MODELESS **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true**Since:** 1.6 **See Also:**[Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html), [Dialog.setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [Dialog.setModalityType(java.awt.Dialog.ModalityType)](http://docs.google.com/java/awt/Dialog.html#setModalityType(java.awt.Dialog.ModalityType)), [GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JDialog

public **JDialog**([Window](http://docs.google.com/java/awt/Window.html) owner,  
 [String](http://docs.google.com/java/lang/String.html) title)

Creates a modeless dialog with the specified title and owner Window.

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

**Parameters:**owner - the Window from which the dialog is displayed or null if this dialog has no ownertitle - the String to display in the dialog's title bar or null if the dialog has no title **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true**Since:** 1.6 **See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JDialog

public **JDialog**([Window](http://docs.google.com/java/awt/Window.html) owner,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 [Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html) modalityType)

Creates a dialog with the specified title, owner Window and modality.

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

**Parameters:**owner - the Window from which the dialog is displayed or null if this dialog has no ownertitle - the String to display in the dialog's title bar or null if the dialog has no titlemodalityType - specifies whether dialog blocks input to other windows when shown. null value and unsupported modality types are equivalent to MODELESS **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true**Since:** 1.6 **See Also:**[Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html), [Dialog.setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [Dialog.setModalityType(java.awt.Dialog.ModalityType)](http://docs.google.com/java/awt/Dialog.html#setModalityType(java.awt.Dialog.ModalityType)), [GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

### JDialog

public **JDialog**([Window](http://docs.google.com/java/awt/Window.html) owner,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 [Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html) modalityType,  
 [GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html) gc)

Creates a dialog with the specified title, owner Window, modality and GraphicsConfiguration.

NOTE: Any popup components (JComboBox, JPopupMenu, JMenuBar) created within a modal dialog will be forced to be lightweight.

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

**Parameters:**owner - the Window from which the dialog is displayed or null if this dialog has no ownertitle - the String to display in the dialog's title bar or null if the dialog has no titlemodalityType - specifies whether dialog blocks input to other windows when shown. null value and unsupported modality types are equivalent to MODELESSgc - the GraphicsConfiguration of the target screen device; if null, the GraphicsConfiguration from the owning window is used; if owner is also null, the system default GraphicsConfiguration is assumed **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true**Since:** 1.6 **See Also:**[Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html), [Dialog.setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [Dialog.setModalityType(java.awt.Dialog.ModalityType)](http://docs.google.com/java/awt/Dialog.html#setModalityType(java.awt.Dialog.ModalityType)), [GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [JComponent.getDefaultLocale()](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale())

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

### dialogInit

protected void **dialogInit**()

Called by the constructors to init the JDialog 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)

Handles window events depending on the state 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/JDialog.html#setDefaultCloseOperation(int))

### setDefaultCloseOperation

public void **setDefaultCloseOperation**(int operation)

Sets the operation that will happen by default when the user initiates a "close" on this dialog. 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 dialog after invoking any registered WindowListener objects.
* DISPOSE\_ON\_CLOSE (defined in WindowConstants): Automatically hide and dispose the dialog after invoking any registered WindowListener objects.

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 dialog **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if defaultCloseOperation value isn't one of the above valid values**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/JDialog.html#getDefaultCloseOperation()), [WindowConstants](http://docs.google.com/javax/swing/WindowConstants.html)

### getDefaultCloseOperation

public int **getDefaultCloseOperation**()

Returns the operation which occurs when the user initiates a "close" on this dialog.

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

### update

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

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) menu)

Sets the menubar for this dialog.

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

### getJMenuBar

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

Returns the menubar set on this dialog.

**See Also:**[setJMenuBar(javax.swing.JMenuBar)](http://docs.google.com/javax/swing/JDialog.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/JDialog.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [setLayout(java.awt.LayoutManager)](http://docs.google.com/javax/swing/JDialog.html#setLayout(java.awt.LayoutManager)), [setRootPaneCheckingEnabled(boolean)](http://docs.google.com/javax/swing/JDialog.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 JDialog.**See Also:**[addImpl(java.awt.Component, java.lang.Object, int)](http://docs.google.com/javax/swing/JDialog.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [setLayout(java.awt.LayoutManager)](http://docs.google.com/javax/swing/JDialog.html#setLayout(java.awt.LayoutManager)), [isRootPaneCheckingEnabled()](http://docs.google.com/javax/swing/JDialog.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/JDialog.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 JDialog 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/JDialog.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 dialog.

**Specified by:**[getRootPane](http://docs.google.com/javax/swing/RootPaneContainer.html#getRootPane()) in interface [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html) **Returns:**this components single JRootPane child.**See Also:**[setRootPane(javax.swing.JRootPane)](http://docs.google.com/javax/swing/JDialog.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 dialog**See Also:**[getRootPane()](http://docs.google.com/javax/swing/JDialog.html#getRootPane())

### getContentPane

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

Returns the contentPane object for this dialog.

**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/JDialog.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 dialog **Throws:** [IllegalComponentStateException](http://docs.google.com/java/awt/IllegalComponentStateException.html) - (a runtime exception) if the content pane parameter is null**See Also:**[JRootPane](http://docs.google.com/javax/swing/JRootPane.html), [getContentPane()](http://docs.google.com/javax/swing/JDialog.html#getContentPane()), [RootPaneContainer.setContentPane(java.awt.Container)](http://docs.google.com/javax/swing/RootPaneContainer.html#setContentPane(java.awt.Container))

### getLayeredPane

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

Returns the layeredPane object for this dialog.

**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/JDialog.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 new layeredPane property **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/JDialog.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 dialog.

**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/JDialog.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 dialog**See Also:**[getGlassPane()](http://docs.google.com/javax/swing/JDialog.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 JDialogs 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 JDialogs will have their Window decorations provided by the current LookAndFeel. Otherwise, newly created JDialogs will have their Window decorations provided by the current window manager.

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

JDialog dialog = new JDialog();  
 dialog.setUndecorated(true);  
 dialog.getRootPane().setWindowDecorationStyle(JRootPane.PLAIN\_DIALOG);

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

### getAccessibleContext

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

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

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JDialog.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/JDesktopPane.AccessibleJDesktopPane.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JDialog.AccessibleJDialog.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JDialog.html)    [**NO FRAMES**](http://docs.google.com/JDialog.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#4d34og8) | [CONSTR](#lnxbz9) | [METHOD](#35nkun2) | DETAIL: [FIELD](#1y810tw) | [CONSTR](#3whwml4) | [METHOD](#1v1yuxt) |

[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).