| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Dialog.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/Desktop.Action.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Dialog.AccessibleAWTDialog.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Dialog.html)    [**NO FRAMES**](http://docs.google.com/Dialog.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#3rdcrjn) | DETAIL: [FIELD](#44sinio) | [CONSTR](#z337ya) | [METHOD](#23ckvvd) |

## **java.awt**

Class Dialog

[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**

**All Implemented Interfaces:** [ImageObserver](http://docs.google.com/java/awt/image/ImageObserver.html), [MenuContainer](http://docs.google.com/java/awt/MenuContainer.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [Accessible](http://docs.google.com/javax/accessibility/Accessible.html) **Direct Known Subclasses:** [FileDialog](http://docs.google.com/java/awt/FileDialog.html), [JDialog](http://docs.google.com/javax/swing/JDialog.html)

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

A Dialog is a top-level window with a title and a border that is typically used to take some form of input from the user. The size of the dialog includes any area designated for the border. The dimensions of the border area can be obtained using the getInsets method, however, since these dimensions are platform-dependent, a valid insets value cannot be obtained until the dialog is made displayable by either calling pack or show. Since the border area is included in the overall size of the dialog, the border effectively obscures a portion of the dialog, constraining the area available for rendering and/or displaying subcomponents to the rectangle which has an upper-left corner location of (insets.left, insets.top), and has a size of width - (insets.left + insets.right) by height - (insets.top + insets.bottom).

The default layout for a dialog is BorderLayout.

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

A dialog may have another window as its owner when it's constructed. When the owner window of a visible dialog is minimized, the dialog will automatically be hidden from the user. When the owner window is subsequently restored, the dialog is made visible to the user again.

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

A dialog can be either modeless (the default) or modal. A modal dialog is one which blocks input to some other top-level windows in the application, except for any windows created with the dialog as their owner. See [AWT Modality](http://docs.google.com/doc-files/Modality.html) specification for details.

Dialogs are capable of generating the following WindowEvents: WindowOpened, WindowClosing, WindowClosed, WindowActivated, WindowDeactivated, WindowGainedFocus, WindowLostFocus.

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

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**Dialog.AccessibleAWTDialog**](http://docs.google.com/java/awt/Dialog.AccessibleAWTDialog.html)            This class implements accessibility support for the Dialog class. |
| static class | [**Dialog.ModalExclusionType**](http://docs.google.com/java/awt/Dialog.ModalExclusionType.html)            Any top-level window can be marked not to be blocked by modal dialogs. |
| static class | [**Dialog.ModalityType**](http://docs.google.com/java/awt/Dialog.ModalityType.html)            Modal dialogs block all input to some top-level windows. |

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

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

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

| **Field Summary** | |
| --- | --- |
| static [Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html) | [**DEFAULT\_MODALITY\_TYPE**](http://docs.google.com/java/awt/Dialog.html#DEFAULT_MODALITY_TYPE)            Default modality type for modal dialogs. |

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

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

| **Constructor Summary** | |
| --- | --- |
| [**Dialog**](http://docs.google.com/java/awt/Dialog.html#Dialog(java.awt.Dialog))([Dialog](http://docs.google.com/java/awt/Dialog.html) owner)            Constructs an initially invisible, modeless Dialog with the specified owner Dialog and an empty title. |
| [**Dialog**](http://docs.google.com/java/awt/Dialog.html#Dialog(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)            Constructs an initially invisible, modeless Dialog with the specified owner Dialog and title. |
| [**Dialog**](http://docs.google.com/java/awt/Dialog.html#Dialog(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)            Constructs an initially invisible Dialog with the specified owner Dialog, title, and modality. |
| [**Dialog**](http://docs.google.com/java/awt/Dialog.html#Dialog(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)            Constructs an initially invisible Dialog with the specified owner Dialog, title, modality and GraphicsConfiguration. |
| [**Dialog**](http://docs.google.com/java/awt/Dialog.html#Dialog(java.awt.Frame))([Frame](http://docs.google.com/java/awt/Frame.html) owner)            Constructs an initially invisible, modeless Dialog with the specified owner Frame and an empty title. |
| [**Dialog**](http://docs.google.com/java/awt/Dialog.html#Dialog(java.awt.Frame,%20boolean))([Frame](http://docs.google.com/java/awt/Frame.html) owner, boolean modal)            Constructs an initially invisible Dialog with the specified owner Frame and modality and an empty title. |
| [**Dialog**](http://docs.google.com/java/awt/Dialog.html#Dialog(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)            Constructs an initially invisible, modeless Dialog with the specified owner Frame and title. |
| [**Dialog**](http://docs.google.com/java/awt/Dialog.html#Dialog(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)            Constructs an initially invisible Dialog with the specified owner Frame, title and modality. |
| [**Dialog**](http://docs.google.com/java/awt/Dialog.html#Dialog(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)            Constructs an initially invisible Dialog with the specified owner Frame, title, modality, and GraphicsConfiguration. |
| [**Dialog**](http://docs.google.com/java/awt/Dialog.html#Dialog(java.awt.Window))([Window](http://docs.google.com/java/awt/Window.html) owner)            Constructs an initially invisible, modeless Dialog with the specified owner Window and an empty title. |
| [**Dialog**](http://docs.google.com/java/awt/Dialog.html#Dialog(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)            Constructs an initially invisible Dialog with the specified owner Window and modality and an empty title. |
| [**Dialog**](http://docs.google.com/java/awt/Dialog.html#Dialog(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)            Constructs an initially invisible, modeless Dialog with the specified owner Window and title. |
| [**Dialog**](http://docs.google.com/java/awt/Dialog.html#Dialog(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)            Constructs an initially invisible Dialog with the specified owner Window, title and modality. |
| [**Dialog**](http://docs.google.com/java/awt/Dialog.html#Dialog(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)            Constructs an initially invisible Dialog with the specified owner Window, title, modality and GraphicsConfiguration. |

| **Method Summary** | |
| --- | --- |
| void | [**addNotify**](http://docs.google.com/java/awt/Dialog.html#addNotify())()            Makes this Dialog displayable by connecting it to a native screen resource. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/java/awt/Dialog.html#getAccessibleContext())()            Gets the AccessibleContext associated with this Dialog. |
| [Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html) | [**getModalityType**](http://docs.google.com/java/awt/Dialog.html#getModalityType())()            Returns the modality type of this dialog. |
| [String](http://docs.google.com/java/lang/String.html) | [**getTitle**](http://docs.google.com/java/awt/Dialog.html#getTitle())()            Gets the title of the dialog. |
| void | [**hide**](http://docs.google.com/java/awt/Dialog.html#hide())()  **Deprecated.** *As of JDK version 1.5, replaced by* [*setVisible(boolean)*](http://docs.google.com/java/awt/Dialog.html#setVisible(boolean))*.* |
| boolean | [**isModal**](http://docs.google.com/java/awt/Dialog.html#isModal())()            Indicates whether the dialog is modal. |
| boolean | [**isResizable**](http://docs.google.com/java/awt/Dialog.html#isResizable())()            Indicates whether this dialog is resizable by the user. |
| boolean | [**isUndecorated**](http://docs.google.com/java/awt/Dialog.html#isUndecorated())()            Indicates whether this dialog is undecorated. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/java/awt/Dialog.html#paramString())()            Returns a string representing the state of this dialog. |
| void | [**setModal**](http://docs.google.com/java/awt/Dialog.html#setModal(boolean))(boolean modal)            Specifies whether this dialog should be modal. |
| void | [**setModalityType**](http://docs.google.com/java/awt/Dialog.html#setModalityType(java.awt.Dialog.ModalityType))([Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html) type)            Sets the modality type for this dialog. |
| void | [**setResizable**](http://docs.google.com/java/awt/Dialog.html#setResizable(boolean))(boolean resizable)            Sets whether this dialog is resizable by the user. |
| void | [**setTitle**](http://docs.google.com/java/awt/Dialog.html#setTitle(java.lang.String))([String](http://docs.google.com/java/lang/String.html) title)            Sets the title of the Dialog. |
| void | [**setUndecorated**](http://docs.google.com/java/awt/Dialog.html#setUndecorated(boolean))(boolean undecorated)            Disables or enables decorations for this dialog. |
| void | [**setVisible**](http://docs.google.com/java/awt/Dialog.html#setVisible(boolean))(boolean b)            Shows or hides this Dialog depending on the value of parameter b. |
| void | [**show**](http://docs.google.com/java/awt/Dialog.html#show())()  **Deprecated.** *As of JDK version 1.5, replaced by* [*setVisible(boolean)*](http://docs.google.com/java/awt/Dialog.html#setVisible(boolean))*.* |
| void | [**toBack**](http://docs.google.com/java/awt/Dialog.html#toBack())()            If this Window is visible, sends this Window to the back and may cause it to lose focus or activation if it is the focused or active Window. |

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

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

| **Methods inherited from class java.lang.**[**Object**](http://docs.google.com/java/lang/Object.html) |
| --- |
| [clone](http://docs.google.com/java/lang/Object.html#clone()), [equals](http://docs.google.com/java/lang/Object.html#equals(java.lang.Object)), [finalize](http://docs.google.com/java/lang/Object.html#finalize()), [getClass](http://docs.google.com/java/lang/Object.html#getClass()), [hashCode](http://docs.google.com/java/lang/Object.html#hashCode()), [notify](http://docs.google.com/java/lang/Object.html#notify()), [notifyAll](http://docs.google.com/java/lang/Object.html#notifyAll()), [wait](http://docs.google.com/java/lang/Object.html#wait()), [wait](http://docs.google.com/java/lang/Object.html#wait(long)), [wait](http://docs.google.com/java/lang/Object.html#wait(long,%20int)) |

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

### DEFAULT\_MODALITY\_TYPE

public static final [Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html) **DEFAULT\_MODALITY\_TYPE**

Default modality type for modal dialogs. The default modality type is APPLICATION\_MODAL. Calling the oldstyle setModal(true) is equal to setModalityType(DEFAULT\_MODALITY\_TYPE).

**Since:** 1.6 **See Also:**[Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html), [setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean))

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

### Dialog

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

Constructs an initially invisible, modeless Dialog with the specified owner Frame and an empty title.

**Parameters:**owner - the owner of the dialog or null if this dialog has no owner **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the owner's GraphicsConfiguration is not from a screen device [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [Component.setSize(int, int)](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [Component.setVisible(boolean)](http://docs.google.com/java/awt/Component.html#setVisible(boolean))

### Dialog

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

Constructs an initially invisible Dialog with the specified owner Frame and modality and an empty title.

**Parameters:**owner - the owner of the dialog or null if this dialog has no ownermodal - specifes whether dialog blocks user input to other top-level windows when shown. If false, the dialog is MODELESS; if true, the modality type property is set to DEFAULT\_MODALITY\_TYPE **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the owner's GraphicsConfiguration is not from a screen device [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when 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), [DEFAULT\_MODALITY\_TYPE](http://docs.google.com/java/awt/Dialog.html#DEFAULT_MODALITY_TYPE), [setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [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())

### Dialog

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

Constructs an initially invisible, modeless Dialog with the specified owner Frame and title.

**Parameters:**owner - the owner of the dialog or null if this dialog has no ownertitle - the title of the dialog or null if this dialog has no title **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the owner's GraphicsConfiguration is not from a screen device [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless()), [Component.setSize(int, int)](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [Component.setVisible(boolean)](http://docs.google.com/java/awt/Component.html#setVisible(boolean))

### Dialog

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

Constructs an initially invisible Dialog with the specified owner Frame, title and modality.

**Parameters:**owner - the owner of the dialog or null if this dialog has no ownertitle - the title of the dialog or null if this dialog has no titlemodal - specifes whether dialog blocks user input to other top-level windows when shown. If false, the dialog is MODELESS; if true, the modality type property is set to DEFAULT\_MODALITY\_TYPE **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the owner's GraphicsConfiguration is not from a screen device [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when 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), [DEFAULT\_MODALITY\_TYPE](http://docs.google.com/java/awt/Dialog.html#DEFAULT_MODALITY_TYPE), [setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [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()), [Component.setSize(int, int)](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [Component.setVisible(boolean)](http://docs.google.com/java/awt/Component.html#setVisible(boolean))

### Dialog

public **Dialog**([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)

Constructs an initially invisible Dialog with the specified owner Frame, title, modality, and GraphicsConfiguration.

**Parameters:**owner - the owner of the dialog or null if this dialog has no ownertitle - the title of the dialog or null if this dialog has no titlemodal - specifes whether dialog blocks user input to other top-level windows when shown. If false, the dialog is MODELESS; if true, the modality type property is set to DEFAULT\_MODALITY\_TYPEgc - the GraphicsConfiguration of the target screen device; if null, the default system GraphicsConfiguration is assumed **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if gc is not from a screen device [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true**Since:** 1.4 **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), [DEFAULT\_MODALITY\_TYPE](http://docs.google.com/java/awt/Dialog.html#DEFAULT_MODALITY_TYPE), [setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [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()), [Component.setSize(int, int)](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [Component.setVisible(boolean)](http://docs.google.com/java/awt/Component.html#setVisible(boolean))

### Dialog

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

Constructs an initially invisible, modeless Dialog with the specified owner Dialog and an empty title.

**Parameters:**owner - the owner of the dialog or null if this dialog has no owner **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the owner's GraphicsConfiguration is not from a screen device [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true**Since:** 1.2 **See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### Dialog

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

Constructs an initially invisible, modeless Dialog with the specified owner Dialog and title.

**Parameters:**owner - the owner of the dialog or null if this has no ownertitle - the title of the dialog or null if this dialog has no title **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the owner's GraphicsConfiguration is not from a screen device [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true**Since:** 1.2 **See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### Dialog

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

Constructs an initially invisible Dialog with the specified owner Dialog, title, and modality.

**Parameters:**owner - the owner of the dialog or null if this dialog has no ownertitle - the title of the dialog or null if this dialog has no titlemodal - specifes whether dialog blocks user input to other top-level windows when shown. If false, the dialog is MODELESS; if true, the modality type property is set to DEFAULT\_MODALITY\_TYPE **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the owner's GraphicsConfiguration is not from a screen device [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true**Since:** 1.2 **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), [DEFAULT\_MODALITY\_TYPE](http://docs.google.com/java/awt/Dialog.html#DEFAULT_MODALITY_TYPE), [setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [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())

### Dialog

public **Dialog**([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)

Constructs an initially invisible Dialog with the specified owner Dialog, title, modality and GraphicsConfiguration.

**Parameters:**owner - the owner of the dialog or null if this dialog has no ownertitle - the title of the dialog or null if this dialog has no titlemodal - specifes whether dialog blocks user input to other top-level windows when shown. If false, the dialog is MODELESS; if true, the modality type property is set to DEFAULT\_MODALITY\_TYPEgc - the GraphicsConfiguration of the target screen device; if null, the default system GraphicsConfiguration is assumed **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if gc is not from a screen device [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true**Since:** 1.4 **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), [DEFAULT\_MODALITY\_TYPE](http://docs.google.com/java/awt/Dialog.html#DEFAULT_MODALITY_TYPE), [setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [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()), [Component.setSize(int, int)](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [Component.setVisible(boolean)](http://docs.google.com/java/awt/Component.html#setVisible(boolean))

### Dialog

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

Constructs an initially invisible, modeless Dialog with the specified owner Window and an empty title.

**Parameters:**owner - the owner of the dialog. The owner must be an instance of [Dialog](http://docs.google.com/java/awt/Dialog.html), [Frame](http://docs.google.com/java/awt/Frame.html), any of their descendents or null **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the owner is not an instance of [Dialog](http://docs.google.com/java/awt/Dialog.html) or [Frame](http://docs.google.com/java/awt/Frame.html) [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the owner's GraphicsConfiguration is not from a screen device [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())

### Dialog

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

Constructs an initially invisible, modeless Dialog with the specified owner Window and title.

**Parameters:**owner - the owner of the dialog. The owner must be an instance of [Dialog](http://docs.google.com/java/awt/Dialog.html), [Frame](http://docs.google.com/java/awt/Frame.html), any of their descendents or nulltitle - the title of the dialog or null if this dialog has no title **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the owner is not an instance of [Dialog](http://docs.google.com/java/awt/Dialog.html) or [Frame](http://docs.google.com/java/awt/Frame.html) [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the owner's GraphicsConfiguration is not from a screen device [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())

### Dialog

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

Constructs an initially invisible Dialog with the specified owner Window and modality and an empty title.

**Parameters:**owner - the owner of the dialog. The owner must be an instance of [Dialog](http://docs.google.com/java/awt/Dialog.html), [Frame](http://docs.google.com/java/awt/Frame.html), any of their descendents or nullmodalityType - specifies whether dialog blocks input to other windows when shown. null value and unsupported modality types are equivalent to MODELESS **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the owner is not an instance of [Dialog](http://docs.google.com/java/awt/Dialog.html) or [Frame](http://docs.google.com/java/awt/Frame.html) [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the owner's GraphicsConfiguration is not from a screen device [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the calling thread does not have permission to create modal dialogs with the given modalityType**Since:** 1.6 **See Also:**[Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html), [setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [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()), [Toolkit.isModalityTypeSupported(java.awt.Dialog.ModalityType)](http://docs.google.com/java/awt/Toolkit.html#isModalityTypeSupported(java.awt.Dialog.ModalityType))

### Dialog

public **Dialog**([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)

Constructs an initially invisible Dialog with the specified owner Window, title and modality.

**Parameters:**owner - the owner of the dialog. The owner must be an instance of [Dialog](http://docs.google.com/java/awt/Dialog.html), [Frame](http://docs.google.com/java/awt/Frame.html), any of their descendents or nulltitle - the title of the dialog or null if this 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:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the owner is not an instance of [Dialog](http://docs.google.com/java/awt/Dialog.html) or [Frame](http://docs.google.com/java/awt/Frame.html) [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the owner's GraphicsConfiguration is not from a screen device [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the calling thread does not have permission to create modal dialogs with the given modalityType**Since:** 1.6 **See Also:**[Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html), [setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [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()), [Toolkit.isModalityTypeSupported(java.awt.Dialog.ModalityType)](http://docs.google.com/java/awt/Toolkit.html#isModalityTypeSupported(java.awt.Dialog.ModalityType))

### Dialog

public **Dialog**([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)

Constructs an initially invisible Dialog with the specified owner Window, title, modality and GraphicsConfiguration.

**Parameters:**owner - the owner of the dialog. The owner must be an instance of [Dialog](http://docs.google.com/java/awt/Dialog.html), [Frame](http://docs.google.com/java/awt/Frame.html), any of their descendents or nulltitle - the title of the dialog or null if this 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 default system GraphicsConfiguration is assumed **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the owner is not an instance of [Dialog](http://docs.google.com/java/awt/Dialog.html) or [Frame](http://docs.google.com/java/awt/Frame.html) [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if gc is not from a screen device [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - when GraphicsEnvironment.isHeadless() returns true [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the calling thread does not have permission to create modal dialogs with the given modalityType**Since:** 1.6 **See Also:**[Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html), [setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [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()), [Toolkit.isModalityTypeSupported(java.awt.Dialog.ModalityType)](http://docs.google.com/java/awt/Toolkit.html#isModalityTypeSupported(java.awt.Dialog.ModalityType))

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

### addNotify

public void **addNotify**()

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

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

### isModal

public boolean **isModal**()

Indicates whether the dialog is modal.

This method is obsolete and is kept for backwards compatiblity only. Use [getModalityType()](http://docs.google.com/java/awt/Dialog.html#getModalityType()) instead.

**Returns:**true if this dialog window is modal; false otherwise**See Also:**[DEFAULT\_MODALITY\_TYPE](http://docs.google.com/java/awt/Dialog.html#DEFAULT_MODALITY_TYPE), [Dialog.ModalityType.MODELESS](http://docs.google.com/java/awt/Dialog.ModalityType.html#MODELESS), [setModal(boolean)](http://docs.google.com/java/awt/Dialog.html#setModal(boolean)), [getModalityType()](http://docs.google.com/java/awt/Dialog.html#getModalityType()), [setModalityType(java.awt.Dialog.ModalityType)](http://docs.google.com/java/awt/Dialog.html#setModalityType(java.awt.Dialog.ModalityType))

### setModal

public void **setModal**(boolean modal)

Specifies whether this dialog should be modal.

This method is obsolete and is kept for backwards compatibility only. Use [setModalityType()](http://docs.google.com/java/awt/Dialog.html#setModalityType(java.awt.Dialog.ModalityType)) instead.

Note: changing modality of the visible dialog may have no effect until it is hidden and then shown again.

**Parameters:**modal - specifies whether dialog blocks input to other windows when shown; calling to setModal(true) is equivalent to setModalityType(Dialog.DEFAULT\_MODALITY\_TYPE), and calling to setModal(false) is equvivalent to setModalityType(Dialog.ModalityType.MODELESS)**Since:** 1.1 **See Also:**[DEFAULT\_MODALITY\_TYPE](http://docs.google.com/java/awt/Dialog.html#DEFAULT_MODALITY_TYPE), [Dialog.ModalityType.MODELESS](http://docs.google.com/java/awt/Dialog.ModalityType.html#MODELESS), [isModal()](http://docs.google.com/java/awt/Dialog.html#isModal()), [getModalityType()](http://docs.google.com/java/awt/Dialog.html#getModalityType()), [setModalityType(java.awt.Dialog.ModalityType)](http://docs.google.com/java/awt/Dialog.html#setModalityType(java.awt.Dialog.ModalityType))

### getModalityType

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

Returns the modality type of this dialog.

**Returns:**modality type of this dialog**Since:** 1.6 **See Also:**[setModalityType(java.awt.Dialog.ModalityType)](http://docs.google.com/java/awt/Dialog.html#setModalityType(java.awt.Dialog.ModalityType))

### setModalityType

public void **setModalityType**([Dialog.ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html) type)

Sets the modality type for this dialog. See [ModalityType](http://docs.google.com/java/awt/Dialog.ModalityType.html) for possible modality types.

If the given modality type is not supported, MODELESS is used. You may want to call getModalityType() after calling this method to ensure that the modality type has been set.

Note: changing modality of the visible dialog may have no effect until it is hidden and then shown again.

**Parameters:**type - specifies whether dialog blocks input to other windows when shown. null value and unsupported modality types are equivalent to MODELESS **Throws:** [SecurityException](http://docs.google.com/java/lang/SecurityException.html) - if the calling thread does not have permission to create modal dialogs with the given modalityType**Since:** 1.6 **See Also:**[getModalityType()](http://docs.google.com/java/awt/Dialog.html#getModalityType()), [Toolkit.isModalityTypeSupported(java.awt.Dialog.ModalityType)](http://docs.google.com/java/awt/Toolkit.html#isModalityTypeSupported(java.awt.Dialog.ModalityType))

### getTitle

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

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

**Returns:**the title of this dialog window. The title may be null.**See Also:**[setTitle(java.lang.String)](http://docs.google.com/java/awt/Dialog.html#setTitle(java.lang.String))

### setTitle

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

Sets the title of the Dialog.

**Parameters:**title - the title displayed in the dialog's border; a null value results in an empty title**See Also:**[getTitle()](http://docs.google.com/java/awt/Dialog.html#getTitle())

### setVisible

public void **setVisible**(boolean b)

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

**Overrides:**[setVisible](http://docs.google.com/java/awt/Window.html#setVisible(boolean)) in class [Window](http://docs.google.com/java/awt/Window.html) **Parameters:**b - if true, makes the Dialog visible, otherwise hides the Dialog. If the dialog and/or its owner are not yet displayable, both are made displayable. The dialog will be validated prior to being made visible. If false, hides the Dialog and then causes setVisible(true) to return if it is currently blocked.

**Notes for modal dialogs**.

* setVisible(true): If the dialog is not already visible, this call will not return until the dialog is hidden by calling setVisible(false) or dispose.
* setVisible(false): Hides the dialog and then returns on setVisible(true) if it is currently blocked.
* It is OK to call this method from the event dispatching thread because the toolkit ensures that other events are not blocked while this method is blocked.

**See Also:**[Window.setVisible(boolean)](http://docs.google.com/java/awt/Window.html#setVisible(boolean)), [Window.dispose()](http://docs.google.com/java/awt/Window.html#dispose()), [Component.isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable()), [Component.validate()](http://docs.google.com/java/awt/Component.html#validate()), [isModal()](http://docs.google.com/java/awt/Dialog.html#isModal())

### show

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

**Deprecated.** *As of JDK version 1.5, replaced by* [*setVisible(boolean)*](http://docs.google.com/java/awt/Dialog.html#setVisible(boolean))*.*

Makes the Dialog visible. If the dialog and/or its owner are not yet displayable, both are made displayable. The dialog will be validated prior to being made visible. If the dialog is already visible, this will bring the dialog to the front.

If the dialog is modal and is not already visible, this call will not return until the dialog is hidden by calling hide or dispose. It is permissible to show modal dialogs from the event dispatching thread because the toolkit will ensure that another event pump runs while the one which invoked this method is blocked.

**Overrides:**[show](http://docs.google.com/java/awt/Window.html#show()) in class [Window](http://docs.google.com/java/awt/Window.html) **See Also:**[Component.hide()](http://docs.google.com/java/awt/Component.html#hide()), [Component.isDisplayable()](http://docs.google.com/java/awt/Component.html#isDisplayable()), [Component.validate()](http://docs.google.com/java/awt/Component.html#validate()), [isModal()](http://docs.google.com/java/awt/Dialog.html#isModal()), [Window.setVisible(boolean)](http://docs.google.com/java/awt/Window.html#setVisible(boolean))

### hide

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

**Deprecated.** *As of JDK version 1.5, replaced by* [*setVisible(boolean)*](http://docs.google.com/java/awt/Dialog.html#setVisible(boolean))*.*

Hides the Dialog and then causes show to return if it is currently blocked.

**Overrides:**[hide](http://docs.google.com/java/awt/Window.html#hide()) in class [Window](http://docs.google.com/java/awt/Window.html) **See Also:**[Window.show()](http://docs.google.com/java/awt/Window.html#show()), [Window.dispose()](http://docs.google.com/java/awt/Window.html#dispose()), [Window.setVisible(boolean)](http://docs.google.com/java/awt/Window.html#setVisible(boolean))

### toBack

public void **toBack**()

If this Window is visible, sends this Window to the back and may cause it to lose focus or activation if it is the focused or active Window.

Places this Window at the bottom of the stacking order and shows it behind any other Windows in this VM. No action will take place is this Window is not visible. Some platforms do not allow Windows which are owned by other Windows to appear below their owners. Every attempt will be made to move this Window as low as possible in the stacking order; however, developers should not assume that this method will move this Window below all other windows in every situation.

Because of variations in native windowing systems, no guarantees about changes to the focused and active Windows can be made. Developers must never assume that this Window is no longer the focused or active Window until this Window receives a WINDOW\_LOST\_FOCUS or WINDOW\_DEACTIVATED event. On platforms where the top-most window is the focused window, this method will **probably** cause this Window to lose focus. In that case, the next highest, focusable Window in this VM will receive focus. On platforms where the stacking order does not typically affect the focused window, this method will **probably** leave the focused and active Windows unchanged.

If this dialog is modal and blocks some windows, then all of them are also sent to the back to keep them below the blocking dialog.

**Overrides:**[toBack](http://docs.google.com/java/awt/Window.html#toBack()) in class [Window](http://docs.google.com/java/awt/Window.html) **See Also:**[Window.toBack()](http://docs.google.com/java/awt/Window.html#toBack())

### isResizable

public boolean **isResizable**()

Indicates whether this dialog is resizable by the user. By default, all dialogs are initially resizable.

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

### setResizable

public void **setResizable**(boolean resizable)

Sets whether this dialog is resizable by the user.

**Parameters:**resizable - true if the user can resize this dialog; false otherwise.**See Also:**[isResizable()](http://docs.google.com/java/awt/Dialog.html#isResizable())

### setUndecorated

public void **setUndecorated**(boolean undecorated)

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

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

### isUndecorated

public boolean **isUndecorated**()

Indicates whether this dialog is undecorated. By default, all dialogs are initially decorated.

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

### paramString

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

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

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

### getAccessibleContext

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

Gets the AccessibleContext associated with this Dialog. For dialogs, the AccessibleContext takes the form of an AccessibleAWTDialog. A new AccessibleAWTDialog instance is created if necessary.

**Specified by:**[getAccessibleContext](http://docs.google.com/javax/accessibility/Accessible.html#getAccessibleContext()) in interface [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)**Overrides:**[getAccessibleContext](http://docs.google.com/java/awt/Window.html#getAccessibleContext()) in class [Window](http://docs.google.com/java/awt/Window.html) **Returns:**an AccessibleAWTDialog that serves as the AccessibleContext of this Dialog**Since:** 1.3

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/Dialog.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/Desktop.Action.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/Dialog.AccessibleAWTDialog.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/Dialog.html)    [**NO FRAMES**](http://docs.google.com/Dialog.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#17dp8vu) | [METHOD](#3rdcrjn) | DETAIL: [FIELD](#44sinio) | [CONSTR](#z337ya) | [METHOD](#23ckvvd) |

[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).