| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JInternalFrame.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/JFrame.AccessibleJFrame.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JInternalFrame.AccessibleJInternalFrame.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JInternalFrame.html)    [**NO FRAMES**](http://docs.google.com/JInternalFrame.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#lnxbz9) | DETAIL: [FIELD](#z337ya) | [CONSTR](#2u6wntf) | [METHOD](#46r0co2) |

## **javax.swing**

Class JInternalFrame

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

**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 **JInternalFrame**extends [JComponent](http://docs.google.com/javax/swing/JComponent.html)implements [Accessible](http://docs.google.com/javax/accessibility/Accessible.html), [WindowConstants](http://docs.google.com/javax/swing/WindowConstants.html), [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html)

A lightweight object that provides many of the features of a native frame, including dragging, closing, becoming an icon, resizing, title display, and support for a menu bar. For task-oriented documentation and examples of using internal frames, see [How to Use Internal Frames](http://java.sun.com/docs/books/tutorial/uiswing/components/internalframe.html), a section in *The Java Tutorial*.

Generally, you add JInternalFrames to a JDesktopPane. The UI delegates the look-and-feel-specific actions to the DesktopManager object maintained by the JDesktopPane.

The JInternalFrame content pane is where you add child components. As a conveniance add and its variants, remove and setLayout have been overridden to forward to the contentPane as necessary. This means you can write:

internalFrame.add(child);

And the child will be added to the contentPane. The content pane is actually managed by an instance of JRootPane, which also manages a layout pane, glass pane, and optional menu bar for the internal frame. Please see the JRootPane documentation for a complete description of these components. Refer to [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html) for details on adding, removing and setting the LayoutManager of a JInternalFrame.

**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:**[InternalFrameEvent](http://docs.google.com/javax/swing/event/InternalFrameEvent.html), [JDesktopPane](http://docs.google.com/javax/swing/JDesktopPane.html), [DesktopManager](http://docs.google.com/javax/swing/DesktopManager.html), [JInternalFrame.JDesktopIcon](http://docs.google.com/javax/swing/JInternalFrame.JDesktopIcon.html), [JRootPane](http://docs.google.com/javax/swing/JRootPane.html), [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html)

| **Nested Class Summary** | |
| --- | --- |
| protected  class | [**JInternalFrame.AccessibleJInternalFrame**](http://docs.google.com/javax/swing/JInternalFrame.AccessibleJInternalFrame.html)            This class implements accessibility support for the JInternalFrame class. |
| static class | [**JInternalFrame.JDesktopIcon**](http://docs.google.com/javax/swing/JInternalFrame.JDesktopIcon.html)            This component represents an iconified version of a JInternalFrame. |

| **Nested classes/interfaces inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [JComponent.AccessibleJComponent](http://docs.google.com/javax/swing/JComponent.AccessibleJComponent.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  boolean | [**closable**](http://docs.google.com/javax/swing/JInternalFrame.html#closable)            The frame can be closed. |
| static [String](http://docs.google.com/java/lang/String.html) | [**CONTENT\_PANE\_PROPERTY**](http://docs.google.com/javax/swing/JInternalFrame.html#CONTENT_PANE_PROPERTY)            Bound property name. |
| protected  [JInternalFrame.JDesktopIcon](http://docs.google.com/javax/swing/JInternalFrame.JDesktopIcon.html) | [**desktopIcon**](http://docs.google.com/javax/swing/JInternalFrame.html#desktopIcon)            The icon that is displayed when this internal frame is iconified. |
| static [String](http://docs.google.com/java/lang/String.html) | [**FRAME\_ICON\_PROPERTY**](http://docs.google.com/javax/swing/JInternalFrame.html#FRAME_ICON_PROPERTY)            Bound property name. |
| protected  [Icon](http://docs.google.com/javax/swing/Icon.html) | [**frameIcon**](http://docs.google.com/javax/swing/JInternalFrame.html#frameIcon)            The icon shown in the top-left corner of this internal frame. |
| static [String](http://docs.google.com/java/lang/String.html) | [**GLASS\_PANE\_PROPERTY**](http://docs.google.com/javax/swing/JInternalFrame.html#GLASS_PANE_PROPERTY)            Bound property name. |
| protected  boolean | [**iconable**](http://docs.google.com/javax/swing/JInternalFrame.html#iconable)            The frame can "iconified" (shrunk down and displayed as an icon-image). |
| static [String](http://docs.google.com/java/lang/String.html) | [**IS\_CLOSED\_PROPERTY**](http://docs.google.com/javax/swing/JInternalFrame.html#IS_CLOSED_PROPERTY)            Constrained property name indicating that the internal frame is closed. |
| static [String](http://docs.google.com/java/lang/String.html) | [**IS\_ICON\_PROPERTY**](http://docs.google.com/javax/swing/JInternalFrame.html#IS_ICON_PROPERTY)            Constrained property name indicating that the internal frame is iconified. |
| static [String](http://docs.google.com/java/lang/String.html) | [**IS\_MAXIMUM\_PROPERTY**](http://docs.google.com/javax/swing/JInternalFrame.html#IS_MAXIMUM_PROPERTY)            Constrained property name indicating that the internal frame is maximized. |
| static [String](http://docs.google.com/java/lang/String.html) | [**IS\_SELECTED\_PROPERTY**](http://docs.google.com/javax/swing/JInternalFrame.html#IS_SELECTED_PROPERTY)            Constrained property name indicated that this frame has selected status. |
| protected  boolean | [**isClosed**](http://docs.google.com/javax/swing/JInternalFrame.html#isClosed)            The frame has been closed. |
| protected  boolean | [**isIcon**](http://docs.google.com/javax/swing/JInternalFrame.html#isIcon)            The frame has been iconified. |
| protected  boolean | [**isMaximum**](http://docs.google.com/javax/swing/JInternalFrame.html#isMaximum)            The frame has been expanded to its maximum size. |
| protected  boolean | [**isSelected**](http://docs.google.com/javax/swing/JInternalFrame.html#isSelected)            The frame is currently selected. |
| static [String](http://docs.google.com/java/lang/String.html) | [**LAYERED\_PANE\_PROPERTY**](http://docs.google.com/javax/swing/JInternalFrame.html#LAYERED_PANE_PROPERTY)            Bound property name. |
| protected  boolean | [**maximizable**](http://docs.google.com/javax/swing/JInternalFrame.html#maximizable)            The frame can be expanded to the size of the desktop pane. |
| static [String](http://docs.google.com/java/lang/String.html) | [**MENU\_BAR\_PROPERTY**](http://docs.google.com/javax/swing/JInternalFrame.html#MENU_BAR_PROPERTY)            Bound property name. |
| protected  boolean | [**resizable**](http://docs.google.com/javax/swing/JInternalFrame.html#resizable)            The frame's size can be changed. |
| static [String](http://docs.google.com/java/lang/String.html) | [**ROOT\_PANE\_PROPERTY**](http://docs.google.com/javax/swing/JInternalFrame.html#ROOT_PANE_PROPERTY)            Bound property name. |
| protected  [JRootPane](http://docs.google.com/javax/swing/JRootPane.html) | [**rootPane**](http://docs.google.com/javax/swing/JInternalFrame.html#rootPane)            The JRootPane instance that manages the content pane and optional menu bar for this internal frame, as well as the glass pane. |
| protected  boolean | [**rootPaneCheckingEnabled**](http://docs.google.com/javax/swing/JInternalFrame.html#rootPaneCheckingEnabled)            If true then calls to add and setLayout will be forwarded to the contentPane. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**title**](http://docs.google.com/javax/swing/JInternalFrame.html#title)            The title displayed in this internal frame's title bar. |
| static [String](http://docs.google.com/java/lang/String.html) | [**TITLE\_PROPERTY**](http://docs.google.com/javax/swing/JInternalFrame.html#TITLE_PROPERTY)            Bound property name. |

| **Fields inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [accessibleContext](http://docs.google.com/javax/swing/JComponent.html#accessibleContext), [listenerList](http://docs.google.com/javax/swing/JComponent.html#listenerList), [TOOL\_TIP\_TEXT\_KEY](http://docs.google.com/javax/swing/JComponent.html#TOOL_TIP_TEXT_KEY), [ui](http://docs.google.com/javax/swing/JComponent.html#ui), [UNDEFINED\_CONDITION](http://docs.google.com/javax/swing/JComponent.html#UNDEFINED_CONDITION), [WHEN\_ANCESTOR\_OF\_FOCUSED\_COMPONENT](http://docs.google.com/javax/swing/JComponent.html#WHEN_ANCESTOR_OF_FOCUSED_COMPONENT), [WHEN\_FOCUSED](http://docs.google.com/javax/swing/JComponent.html#WHEN_FOCUSED), [WHEN\_IN\_FOCUSED\_WINDOW](http://docs.google.com/javax/swing/JComponent.html#WHEN_IN_FOCUSED_WINDOW) |

| **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** | |
| --- | --- |
| [**JInternalFrame**](http://docs.google.com/javax/swing/JInternalFrame.html#JInternalFrame())()            Creates a non-resizable, non-closable, non-maximizable, non-iconifiable JInternalFrame with no title. |
| [**JInternalFrame**](http://docs.google.com/javax/swing/JInternalFrame.html#JInternalFrame(java.lang.String))([String](http://docs.google.com/java/lang/String.html) title)            Creates a non-resizable, non-closable, non-maximizable, non-iconifiable JInternalFrame with the specified title. |
| [**JInternalFrame**](http://docs.google.com/javax/swing/JInternalFrame.html#JInternalFrame(java.lang.String,%20boolean))([String](http://docs.google.com/java/lang/String.html) title, boolean resizable)            Creates a non-closable, non-maximizable, non-iconifiable JInternalFrame with the specified title and resizability. |
| [**JInternalFrame**](http://docs.google.com/javax/swing/JInternalFrame.html#JInternalFrame(java.lang.String,%20boolean,%20boolean))([String](http://docs.google.com/java/lang/String.html) title, boolean resizable, boolean closable)            Creates a non-maximizable, non-iconifiable JInternalFrame with the specified title, resizability, and closability. |
| [**JInternalFrame**](http://docs.google.com/javax/swing/JInternalFrame.html#JInternalFrame(java.lang.String,%20boolean,%20boolean,%20boolean))([String](http://docs.google.com/java/lang/String.html) title, boolean resizable, boolean closable, boolean maximizable)            Creates a non-iconifiable JInternalFrame with the specified title, resizability, closability, and maximizability. |
| [**JInternalFrame**](http://docs.google.com/javax/swing/JInternalFrame.html#JInternalFrame(java.lang.String,%20boolean,%20boolean,%20boolean,%20boolean))([String](http://docs.google.com/java/lang/String.html) title, boolean resizable, boolean closable, boolean maximizable, boolean iconifiable)            Creates a JInternalFrame with the specified title, resizability, closability, maximizability, and iconifiability. |

| **Method Summary** | |
| --- | --- |
| protected  void | [**addImpl**](http://docs.google.com/javax/swing/JInternalFrame.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. |
| void | [**addInternalFrameListener**](http://docs.google.com/javax/swing/JInternalFrame.html#addInternalFrameListener(javax.swing.event.InternalFrameListener))([InternalFrameListener](http://docs.google.com/javax/swing/event/InternalFrameListener.html) l)            Adds the specified listener to receive internal frame events from this internal frame. |
| protected  [JRootPane](http://docs.google.com/javax/swing/JRootPane.html) | [**createRootPane**](http://docs.google.com/javax/swing/JInternalFrame.html#createRootPane())()            Called by the constructor to set up the JRootPane. |
| void | [**dispose**](http://docs.google.com/javax/swing/JInternalFrame.html#dispose())()            Makes this internal frame invisible, unselected, and closed. |
| void | [**doDefaultCloseAction**](http://docs.google.com/javax/swing/JInternalFrame.html#doDefaultCloseAction())()            Fires an INTERNAL\_FRAME\_CLOSING event and then performs the action specified by the internal frame's default close operation. |
| protected  void | [**fireInternalFrameEvent**](http://docs.google.com/javax/swing/JInternalFrame.html#fireInternalFrameEvent(int))(int id)            Fires an internal frame event. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JInternalFrame.html#getAccessibleContext())()            Gets the AccessibleContext associated with this JInternalFrame. |
| [Container](http://docs.google.com/java/awt/Container.html) | [**getContentPane**](http://docs.google.com/javax/swing/JInternalFrame.html#getContentPane())()            Returns the content pane for this internal frame. |
| int | [**getDefaultCloseOperation**](http://docs.google.com/javax/swing/JInternalFrame.html#getDefaultCloseOperation())()            Returns the default operation that occurs when the user initiates a "close" on this internal frame. |
| [JInternalFrame.JDesktopIcon](http://docs.google.com/javax/swing/JInternalFrame.JDesktopIcon.html) | [**getDesktopIcon**](http://docs.google.com/javax/swing/JInternalFrame.html#getDesktopIcon())()            Returns the JDesktopIcon used when this JInternalFrame is iconified. |
| [JDesktopPane](http://docs.google.com/javax/swing/JDesktopPane.html) | [**getDesktopPane**](http://docs.google.com/javax/swing/JInternalFrame.html#getDesktopPane())()            Convenience method that searches the ancestor hierarchy for a JDesktop instance. |
| [Container](http://docs.google.com/java/awt/Container.html) | [**getFocusCycleRootAncestor**](http://docs.google.com/javax/swing/JInternalFrame.html#getFocusCycleRootAncestor())()            Always returns null because JInternalFrames must always be roots of a focus traversal cycle. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getFocusOwner**](http://docs.google.com/javax/swing/JInternalFrame.html#getFocusOwner())()            If this JInternalFrame is active, returns the child that has focus. |
| [Icon](http://docs.google.com/javax/swing/Icon.html) | [**getFrameIcon**](http://docs.google.com/javax/swing/JInternalFrame.html#getFrameIcon())()            Returns the image displayed in the title bar of this internal frame (usually in the top-left corner). |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getGlassPane**](http://docs.google.com/javax/swing/JInternalFrame.html#getGlassPane())()            Returns the glass pane for this internal frame. |
| [InternalFrameListener](http://docs.google.com/javax/swing/event/InternalFrameListener.html)[] | [**getInternalFrameListeners**](http://docs.google.com/javax/swing/JInternalFrame.html#getInternalFrameListeners())()            Returns an array of all the InternalFrameListeners added to this JInternalFrame with addInternalFrameListener. |
| [JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) | [**getJMenuBar**](http://docs.google.com/javax/swing/JInternalFrame.html#getJMenuBar())()            Returns the current JMenuBar for this JInternalFrame, or null if no menu bar has been set. |
| [Cursor](http://docs.google.com/java/awt/Cursor.html) | [**getLastCursor**](http://docs.google.com/javax/swing/JInternalFrame.html#getLastCursor())()            Returns the last Cursor that was set by the setCursor method that is not a resizable Cursor. |
| int | [**getLayer**](http://docs.google.com/javax/swing/JInternalFrame.html#getLayer())()            Convenience method for getting the layer attribute of this component. |
| [JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html) | [**getLayeredPane**](http://docs.google.com/javax/swing/JInternalFrame.html#getLayeredPane())()            Returns the layered pane for this internal frame. |
| [JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) | [**getMenuBar**](http://docs.google.com/javax/swing/JInternalFrame.html#getMenuBar())()  **Deprecated.** *As of Swing version 1.0.3, replaced by getJMenuBar().* |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getMostRecentFocusOwner**](http://docs.google.com/javax/swing/JInternalFrame.html#getMostRecentFocusOwner())()            Returns the child component of this JInternalFrame that will receive the focus when this JInternalFrame is selected. |
| [Rectangle](http://docs.google.com/java/awt/Rectangle.html) | [**getNormalBounds**](http://docs.google.com/javax/swing/JInternalFrame.html#getNormalBounds())()            If the JInternalFrame is not in maximized state, returns getBounds(); otherwise, returns the bounds that the JInternalFrame would be restored to. |
| [JRootPane](http://docs.google.com/javax/swing/JRootPane.html) | [**getRootPane**](http://docs.google.com/javax/swing/JInternalFrame.html#getRootPane())()            Returns the rootPane object for this internal frame. |
| [String](http://docs.google.com/java/lang/String.html) | [**getTitle**](http://docs.google.com/javax/swing/JInternalFrame.html#getTitle())()            Returns the title of the JInternalFrame. |
| [InternalFrameUI](http://docs.google.com/javax/swing/plaf/InternalFrameUI.html) | [**getUI**](http://docs.google.com/javax/swing/JInternalFrame.html#getUI())()            Returns the look-and-feel object that renders this component. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUIClassID**](http://docs.google.com/javax/swing/JInternalFrame.html#getUIClassID())()            Returns the name of the look-and-feel class that renders this component. |
| [String](http://docs.google.com/java/lang/String.html) | [**getWarningString**](http://docs.google.com/javax/swing/JInternalFrame.html#getWarningString())()            Gets the warning string that is displayed with this internal frame. |
| void | [**hide**](http://docs.google.com/javax/swing/JInternalFrame.html#hide())() |
| boolean | [**isClosable**](http://docs.google.com/javax/swing/JInternalFrame.html#isClosable())()            Returns whether this JInternalFrame can be closed by some user action. |
| boolean | [**isClosed**](http://docs.google.com/javax/swing/JInternalFrame.html#isClosed())()            Returns whether this JInternalFrame is currently closed. |
| boolean | [**isFocusCycleRoot**](http://docs.google.com/javax/swing/JInternalFrame.html#isFocusCycleRoot())()            Always returns true because all JInternalFrames must be roots of a focus traversal cycle. |
| boolean | [**isIcon**](http://docs.google.com/javax/swing/JInternalFrame.html#isIcon())()            Returns whether the JInternalFrame is currently iconified. |
| boolean | [**isIconifiable**](http://docs.google.com/javax/swing/JInternalFrame.html#isIconifiable())()            Gets the iconable property, which by default is false. |
| boolean | [**isMaximizable**](http://docs.google.com/javax/swing/JInternalFrame.html#isMaximizable())()            Gets the value of the maximizable property. |
| boolean | [**isMaximum**](http://docs.google.com/javax/swing/JInternalFrame.html#isMaximum())()            Returns whether the JInternalFrame is currently maximized. |
| boolean | [**isResizable**](http://docs.google.com/javax/swing/JInternalFrame.html#isResizable())()            Returns whether the JInternalFrame can be resized by some user action. |
| protected  boolean | [**isRootPaneCheckingEnabled**](http://docs.google.com/javax/swing/JInternalFrame.html#isRootPaneCheckingEnabled())()            Returns whether calls to add and setLayout are forwarded to the contentPane. |
| boolean | [**isSelected**](http://docs.google.com/javax/swing/JInternalFrame.html#isSelected())()            Returns whether the JInternalFrame is the currently "selected" or active frame. |
| void | [**moveToBack**](http://docs.google.com/javax/swing/JInternalFrame.html#moveToBack())()            Convenience method that moves this component to position -1 if its parent is a JLayeredPane. |
| void | [**moveToFront**](http://docs.google.com/javax/swing/JInternalFrame.html#moveToFront())()            Convenience method that moves this component to position 0 if its parent is a JLayeredPane. |
| void | [**pack**](http://docs.google.com/javax/swing/JInternalFrame.html#pack())()            Causes subcomponents of this JInternalFrame to be laid out at their preferred size. |
| protected  void | [**paintComponent**](http://docs.google.com/javax/swing/JInternalFrame.html#paintComponent(java.awt.Graphics))([Graphics](http://docs.google.com/java/awt/Graphics.html) g)            Overridden to allow optimized painting when the internal frame is being dragged. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/JInternalFrame.html#paramString())()            Returns a string representation of this JInternalFrame. |
| void | [**remove**](http://docs.google.com/javax/swing/JInternalFrame.html#remove(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) comp)            Removes the specified component from the container. |
| void | [**removeInternalFrameListener**](http://docs.google.com/javax/swing/JInternalFrame.html#removeInternalFrameListener(javax.swing.event.InternalFrameListener))([InternalFrameListener](http://docs.google.com/javax/swing/event/InternalFrameListener.html) l)            Removes the specified internal frame listener so that it no longer receives internal frame events from this internal frame. |
| void | [**reshape**](http://docs.google.com/javax/swing/JInternalFrame.html#reshape(int,%20int,%20int,%20int))(int x, int y, int width, int height)            Moves and resizes this component. |
| void | [**restoreSubcomponentFocus**](http://docs.google.com/javax/swing/JInternalFrame.html#restoreSubcomponentFocus())()            Requests the internal frame to restore focus to the last subcomponent that had focus. |
| void | [**setClosable**](http://docs.google.com/javax/swing/JInternalFrame.html#setClosable(boolean))(boolean b)            Sets whether this JInternalFrame can be closed by some user action. |
| void | [**setClosed**](http://docs.google.com/javax/swing/JInternalFrame.html#setClosed(boolean))(boolean b)            Closes this internal frame if the argument is true. |
| void | [**setContentPane**](http://docs.google.com/javax/swing/JInternalFrame.html#setContentPane(java.awt.Container))([Container](http://docs.google.com/java/awt/Container.html) c)            Sets this JInternalFrame's contentPane property. |
| void | [**setCursor**](http://docs.google.com/javax/swing/JInternalFrame.html#setCursor(java.awt.Cursor))([Cursor](http://docs.google.com/java/awt/Cursor.html) cursor)            Sets the cursor image to the specified cursor. |
| void | [**setDefaultCloseOperation**](http://docs.google.com/javax/swing/JInternalFrame.html#setDefaultCloseOperation(int))(int operation)            Sets the operation that will happen by default when the user initiates a "close" on this internal frame. |
| void | [**setDesktopIcon**](http://docs.google.com/javax/swing/JInternalFrame.html#setDesktopIcon(javax.swing.JInternalFrame.JDesktopIcon))([JInternalFrame.JDesktopIcon](http://docs.google.com/javax/swing/JInternalFrame.JDesktopIcon.html) d)            Sets the JDesktopIcon associated with this JInternalFrame. |
| void | [**setFocusCycleRoot**](http://docs.google.com/javax/swing/JInternalFrame.html#setFocusCycleRoot(boolean))(boolean focusCycleRoot)            Does nothing because JInternalFrames must always be roots of a focus traversal cycle. |
| void | [**setFrameIcon**](http://docs.google.com/javax/swing/JInternalFrame.html#setFrameIcon(javax.swing.Icon))([Icon](http://docs.google.com/javax/swing/Icon.html) icon)            Sets an image to be displayed in the titlebar of this internal frame (usually in the top-left corner). |
| void | [**setGlassPane**](http://docs.google.com/javax/swing/JInternalFrame.html#setGlassPane(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) glass)            Sets this JInternalFrame's glassPane property. |
| void | [**setIcon**](http://docs.google.com/javax/swing/JInternalFrame.html#setIcon(boolean))(boolean b)            Iconifies or de-iconifies this internal frame, if the look and feel supports iconification. |
| void | [**setIconifiable**](http://docs.google.com/javax/swing/JInternalFrame.html#setIconifiable(boolean))(boolean b)            Sets the iconable property, which must be true for the user to be able to make the JInternalFrame an icon. |
| void | [**setJMenuBar**](http://docs.google.com/javax/swing/JInternalFrame.html#setJMenuBar(javax.swing.JMenuBar))([JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) m)            Sets the menuBar property for this JInternalFrame. |
| void | [**setLayer**](http://docs.google.com/javax/swing/JInternalFrame.html#setLayer(int))(int layer)            Convenience method for setting the layer attribute of this component. |
| void | [**setLayer**](http://docs.google.com/javax/swing/JInternalFrame.html#setLayer(java.lang.Integer))([Integer](http://docs.google.com/java/lang/Integer.html) layer)            Convenience method for setting the layer attribute of this component. |
| void | [**setLayeredPane**](http://docs.google.com/javax/swing/JInternalFrame.html#setLayeredPane(javax.swing.JLayeredPane))([JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html) layered)            Sets this JInternalFrame's layeredPane property. |
| void | [**setLayout**](http://docs.google.com/javax/swing/JInternalFrame.html#setLayout(java.awt.LayoutManager))([LayoutManager](http://docs.google.com/java/awt/LayoutManager.html) manager)            Ensures that, by default, the layout of this component cannot be set. |
| void | [**setMaximizable**](http://docs.google.com/javax/swing/JInternalFrame.html#setMaximizable(boolean))(boolean b)            Sets the maximizable property, which determines whether the JInternalFrame can be maximized by some user action. |
| void | [**setMaximum**](http://docs.google.com/javax/swing/JInternalFrame.html#setMaximum(boolean))(boolean b)            Maximizes and restores this internal frame. |
| void | [**setMenuBar**](http://docs.google.com/javax/swing/JInternalFrame.html#setMenuBar(javax.swing.JMenuBar))([JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) m)  **Deprecated.** *As of Swing version 1.0.3 replaced by setJMenuBar(JMenuBar m).* |
| void | [**setNormalBounds**](http://docs.google.com/javax/swing/JInternalFrame.html#setNormalBounds(java.awt.Rectangle))([Rectangle](http://docs.google.com/java/awt/Rectangle.html) r)            Sets the normal bounds for this internal frame, the bounds that this internal frame would be restored to from its maximized state. |
| void | [**setResizable**](http://docs.google.com/javax/swing/JInternalFrame.html#setResizable(boolean))(boolean b)            Sets whether the JInternalFrame can be resized by some user action. |
| protected  void | [**setRootPane**](http://docs.google.com/javax/swing/JInternalFrame.html#setRootPane(javax.swing.JRootPane))([JRootPane](http://docs.google.com/javax/swing/JRootPane.html) root)            Sets the rootPane property for this JInternalFrame. |
| protected  void | [**setRootPaneCheckingEnabled**](http://docs.google.com/javax/swing/JInternalFrame.html#setRootPaneCheckingEnabled(boolean))(boolean enabled)            Sets whether calls to add and setLayout are forwarded to the contentPane. |
| void | [**setSelected**](http://docs.google.com/javax/swing/JInternalFrame.html#setSelected(boolean))(boolean selected)            Selects or deselects the internal frame if it's showing. |
| void | [**setTitle**](http://docs.google.com/javax/swing/JInternalFrame.html#setTitle(java.lang.String))([String](http://docs.google.com/java/lang/String.html) title)            Sets the JInternalFrame title. |
| void | [**setUI**](http://docs.google.com/javax/swing/JInternalFrame.html#setUI(javax.swing.plaf.InternalFrameUI))([InternalFrameUI](http://docs.google.com/javax/swing/plaf/InternalFrameUI.html) ui)            Sets the UI delegate for this JInternalFrame. |
| void | [**show**](http://docs.google.com/javax/swing/JInternalFrame.html#show())()            If the internal frame is not visible, brings the internal frame to the front, makes it visible, and attempts to select it. |
| void | [**toBack**](http://docs.google.com/javax/swing/JInternalFrame.html#toBack())()            Sends this internal frame to the back. |
| void | [**toFront**](http://docs.google.com/javax/swing/JInternalFrame.html#toFront())()            Brings this internal frame to the front. |
| void | [**updateUI**](http://docs.google.com/javax/swing/JInternalFrame.html#updateUI())()            Notification from the UIManager that the look and feel has changed. |

| **Methods inherited from class javax.swing.**[**JComponent**](http://docs.google.com/javax/swing/JComponent.html) |
| --- |
| [addAncestorListener](http://docs.google.com/javax/swing/JComponent.html#addAncestorListener(javax.swing.event.AncestorListener)), [addNotify](http://docs.google.com/javax/swing/JComponent.html#addNotify()), [addVetoableChangeListener](http://docs.google.com/javax/swing/JComponent.html#addVetoableChangeListener(java.beans.VetoableChangeListener)), [computeVisibleRect](http://docs.google.com/javax/swing/JComponent.html#computeVisibleRect(java.awt.Rectangle)), [contains](http://docs.google.com/javax/swing/JComponent.html#contains(int,%20int)), [createToolTip](http://docs.google.com/javax/swing/JComponent.html#createToolTip()), [disable](http://docs.google.com/javax/swing/JComponent.html#disable()), [enable](http://docs.google.com/javax/swing/JComponent.html#enable()), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20boolean,%20boolean)), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20char,%20char)), [firePropertyChange](http://docs.google.com/javax/swing/JComponent.html#firePropertyChange(java.lang.String,%20int,%20int)), [fireVetoableChange](http://docs.google.com/javax/swing/JComponent.html#fireVetoableChange(java.lang.String,%20java.lang.Object,%20java.lang.Object)), [getActionForKeyStroke](http://docs.google.com/javax/swing/JComponent.html#getActionForKeyStroke(javax.swing.KeyStroke)), [getActionMap](http://docs.google.com/javax/swing/JComponent.html#getActionMap()), [getAlignmentX](http://docs.google.com/javax/swing/JComponent.html#getAlignmentX()), [getAlignmentY](http://docs.google.com/javax/swing/JComponent.html#getAlignmentY()), [getAncestorListeners](http://docs.google.com/javax/swing/JComponent.html#getAncestorListeners()), [getAutoscrolls](http://docs.google.com/javax/swing/JComponent.html#getAutoscrolls()), [getBaseline](http://docs.google.com/javax/swing/JComponent.html#getBaseline(int,%20int)), [getBaselineResizeBehavior](http://docs.google.com/javax/swing/JComponent.html#getBaselineResizeBehavior()), [getBorder](http://docs.google.com/javax/swing/JComponent.html#getBorder()), [getBounds](http://docs.google.com/javax/swing/JComponent.html#getBounds(java.awt.Rectangle)), [getClientProperty](http://docs.google.com/javax/swing/JComponent.html#getClientProperty(java.lang.Object)), [getComponentGraphics](http://docs.google.com/javax/swing/JComponent.html#getComponentGraphics(java.awt.Graphics)), [getComponentPopupMenu](http://docs.google.com/javax/swing/JComponent.html#getComponentPopupMenu()), [getConditionForKeyStroke](http://docs.google.com/javax/swing/JComponent.html#getConditionForKeyStroke(javax.swing.KeyStroke)), [getDebugGraphicsOptions](http://docs.google.com/javax/swing/JComponent.html#getDebugGraphicsOptions()), [getDefaultLocale](http://docs.google.com/javax/swing/JComponent.html#getDefaultLocale()), [getFontMetrics](http://docs.google.com/javax/swing/JComponent.html#getFontMetrics(java.awt.Font)), [getGraphics](http://docs.google.com/javax/swing/JComponent.html#getGraphics()), [getHeight](http://docs.google.com/javax/swing/JComponent.html#getHeight()), [getInheritsPopupMenu](http://docs.google.com/javax/swing/JComponent.html#getInheritsPopupMenu()), [getInputMap](http://docs.google.com/javax/swing/JComponent.html#getInputMap()), [getInputMap](http://docs.google.com/javax/swing/JComponent.html#getInputMap(int)), [getInputVerifier](http://docs.google.com/javax/swing/JComponent.html#getInputVerifier()), [getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets()), [getInsets](http://docs.google.com/javax/swing/JComponent.html#getInsets(java.awt.Insets)), [getListeners](http://docs.google.com/javax/swing/JComponent.html#getListeners(java.lang.Class)), [getLocation](http://docs.google.com/javax/swing/JComponent.html#getLocation(java.awt.Point)), [getMaximumSize](http://docs.google.com/javax/swing/JComponent.html#getMaximumSize()), [getMinimumSize](http://docs.google.com/javax/swing/JComponent.html#getMinimumSize()), [getNextFocusableComponent](http://docs.google.com/javax/swing/JComponent.html#getNextFocusableComponent()), [getPopupLocation](http://docs.google.com/javax/swing/JComponent.html#getPopupLocation(java.awt.event.MouseEvent)), [getPreferredSize](http://docs.google.com/javax/swing/JComponent.html#getPreferredSize()), [getRegisteredKeyStrokes](http://docs.google.com/javax/swing/JComponent.html#getRegisteredKeyStrokes()), [getSize](http://docs.google.com/javax/swing/JComponent.html#getSize(java.awt.Dimension)), [getToolTipLocation](http://docs.google.com/javax/swing/JComponent.html#getToolTipLocation(java.awt.event.MouseEvent)), [getToolTipText](http://docs.google.com/javax/swing/JComponent.html#getToolTipText()), [getToolTipText](http://docs.google.com/javax/swing/JComponent.html#getToolTipText(java.awt.event.MouseEvent)), [getTopLevelAncestor](http://docs.google.com/javax/swing/JComponent.html#getTopLevelAncestor()), [getTransferHandler](http://docs.google.com/javax/swing/JComponent.html#getTransferHandler()), [getVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#getVerifyInputWhenFocusTarget()), [getVetoableChangeListeners](http://docs.google.com/javax/swing/JComponent.html#getVetoableChangeListeners()), [getVisibleRect](http://docs.google.com/javax/swing/JComponent.html#getVisibleRect()), [getWidth](http://docs.google.com/javax/swing/JComponent.html#getWidth()), [getX](http://docs.google.com/javax/swing/JComponent.html#getX()), [getY](http://docs.google.com/javax/swing/JComponent.html#getY()), [grabFocus](http://docs.google.com/javax/swing/JComponent.html#grabFocus()), [isDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#isDoubleBuffered()), [isLightweightComponent](http://docs.google.com/javax/swing/JComponent.html#isLightweightComponent(java.awt.Component)), [isManagingFocus](http://docs.google.com/javax/swing/JComponent.html#isManagingFocus()), [isOpaque](http://docs.google.com/javax/swing/JComponent.html#isOpaque()), [isOptimizedDrawingEnabled](http://docs.google.com/javax/swing/JComponent.html#isOptimizedDrawingEnabled()), [isPaintingForPrint](http://docs.google.com/javax/swing/JComponent.html#isPaintingForPrint()), [isPaintingTile](http://docs.google.com/javax/swing/JComponent.html#isPaintingTile()), [isRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#isRequestFocusEnabled()), [isValidateRoot](http://docs.google.com/javax/swing/JComponent.html#isValidateRoot()), [paint](http://docs.google.com/javax/swing/JComponent.html#paint(java.awt.Graphics)), [paintBorder](http://docs.google.com/javax/swing/JComponent.html#paintBorder(java.awt.Graphics)), [paintChildren](http://docs.google.com/javax/swing/JComponent.html#paintChildren(java.awt.Graphics)), [paintImmediately](http://docs.google.com/javax/swing/JComponent.html#paintImmediately(int,%20int,%20int,%20int)), [paintImmediately](http://docs.google.com/javax/swing/JComponent.html#paintImmediately(java.awt.Rectangle)), [print](http://docs.google.com/javax/swing/JComponent.html#print(java.awt.Graphics)), [printAll](http://docs.google.com/javax/swing/JComponent.html#printAll(java.awt.Graphics)), [printBorder](http://docs.google.com/javax/swing/JComponent.html#printBorder(java.awt.Graphics)), [printChildren](http://docs.google.com/javax/swing/JComponent.html#printChildren(java.awt.Graphics)), [printComponent](http://docs.google.com/javax/swing/JComponent.html#printComponent(java.awt.Graphics)), [processComponentKeyEvent](http://docs.google.com/javax/swing/JComponent.html#processComponentKeyEvent(java.awt.event.KeyEvent)), [processKeyBinding](http://docs.google.com/javax/swing/JComponent.html#processKeyBinding(javax.swing.KeyStroke,%20java.awt.event.KeyEvent,%20int,%20boolean)), [processKeyEvent](http://docs.google.com/javax/swing/JComponent.html#processKeyEvent(java.awt.event.KeyEvent)), [processMouseEvent](http://docs.google.com/javax/swing/JComponent.html#processMouseEvent(java.awt.event.MouseEvent)), [processMouseMotionEvent](http://docs.google.com/javax/swing/JComponent.html#processMouseMotionEvent(java.awt.event.MouseEvent)), [putClientProperty](http://docs.google.com/javax/swing/JComponent.html#putClientProperty(java.lang.Object,%20java.lang.Object)), [registerKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#registerKeyboardAction(java.awt.event.ActionListener,%20javax.swing.KeyStroke,%20int)), [registerKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#registerKeyboardAction(java.awt.event.ActionListener,%20java.lang.String,%20javax.swing.KeyStroke,%20int)), [removeAncestorListener](http://docs.google.com/javax/swing/JComponent.html#removeAncestorListener(javax.swing.event.AncestorListener)), [removeNotify](http://docs.google.com/javax/swing/JComponent.html#removeNotify()), [removeVetoableChangeListener](http://docs.google.com/javax/swing/JComponent.html#removeVetoableChangeListener(java.beans.VetoableChangeListener)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(long,%20int,%20int,%20int,%20int)), [repaint](http://docs.google.com/javax/swing/JComponent.html#repaint(java.awt.Rectangle)), [requestDefaultFocus](http://docs.google.com/javax/swing/JComponent.html#requestDefaultFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus()), [requestFocus](http://docs.google.com/javax/swing/JComponent.html#requestFocus(boolean)), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow()), [requestFocusInWindow](http://docs.google.com/javax/swing/JComponent.html#requestFocusInWindow(boolean)), [resetKeyboardActions](http://docs.google.com/javax/swing/JComponent.html#resetKeyboardActions()), [revalidate](http://docs.google.com/javax/swing/JComponent.html#revalidate()), [scrollRectToVisible](http://docs.google.com/javax/swing/JComponent.html#scrollRectToVisible(java.awt.Rectangle)), [setActionMap](http://docs.google.com/javax/swing/JComponent.html#setActionMap(javax.swing.ActionMap)), [setAlignmentX](http://docs.google.com/javax/swing/JComponent.html#setAlignmentX(float)), [setAlignmentY](http://docs.google.com/javax/swing/JComponent.html#setAlignmentY(float)), [setAutoscrolls](http://docs.google.com/javax/swing/JComponent.html#setAutoscrolls(boolean)), [setBackground](http://docs.google.com/javax/swing/JComponent.html#setBackground(java.awt.Color)), [setBorder](http://docs.google.com/javax/swing/JComponent.html#setBorder(javax.swing.border.Border)), [setComponentPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setComponentPopupMenu(javax.swing.JPopupMenu)), [setDebugGraphicsOptions](http://docs.google.com/javax/swing/JComponent.html#setDebugGraphicsOptions(int)), [setDefaultLocale](http://docs.google.com/javax/swing/JComponent.html#setDefaultLocale(java.util.Locale)), [setDoubleBuffered](http://docs.google.com/javax/swing/JComponent.html#setDoubleBuffered(boolean)), [setEnabled](http://docs.google.com/javax/swing/JComponent.html#setEnabled(boolean)), [setFocusTraversalKeys](http://docs.google.com/javax/swing/JComponent.html#setFocusTraversalKeys(int,%20java.util.Set)), [setFont](http://docs.google.com/javax/swing/JComponent.html#setFont(java.awt.Font)), [setForeground](http://docs.google.com/javax/swing/JComponent.html#setForeground(java.awt.Color)), [setInheritsPopupMenu](http://docs.google.com/javax/swing/JComponent.html#setInheritsPopupMenu(boolean)), [setInputMap](http://docs.google.com/javax/swing/JComponent.html#setInputMap(int,%20javax.swing.InputMap)), [setInputVerifier](http://docs.google.com/javax/swing/JComponent.html#setInputVerifier(javax.swing.InputVerifier)), [setMaximumSize](http://docs.google.com/javax/swing/JComponent.html#setMaximumSize(java.awt.Dimension)), [setMinimumSize](http://docs.google.com/javax/swing/JComponent.html#setMinimumSize(java.awt.Dimension)), [setNextFocusableComponent](http://docs.google.com/javax/swing/JComponent.html#setNextFocusableComponent(java.awt.Component)), [setOpaque](http://docs.google.com/javax/swing/JComponent.html#setOpaque(boolean)), [setPreferredSize](http://docs.google.com/javax/swing/JComponent.html#setPreferredSize(java.awt.Dimension)), [setRequestFocusEnabled](http://docs.google.com/javax/swing/JComponent.html#setRequestFocusEnabled(boolean)), [setToolTipText](http://docs.google.com/javax/swing/JComponent.html#setToolTipText(java.lang.String)), [setTransferHandler](http://docs.google.com/javax/swing/JComponent.html#setTransferHandler(javax.swing.TransferHandler)), [setUI](http://docs.google.com/javax/swing/JComponent.html#setUI(javax.swing.plaf.ComponentUI)), [setVerifyInputWhenFocusTarget](http://docs.google.com/javax/swing/JComponent.html#setVerifyInputWhenFocusTarget(boolean)), [setVisible](http://docs.google.com/javax/swing/JComponent.html#setVisible(boolean)), [unregisterKeyboardAction](http://docs.google.com/javax/swing/JComponent.html#unregisterKeyboardAction(javax.swing.KeyStroke)), [update](http://docs.google.com/javax/swing/JComponent.html#update(java.awt.Graphics)) |

| **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)), [addPropertyChangeListener](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.beans.PropertyChangeListener)), [addPropertyChangeListener](http://docs.google.com/java/awt/Container.html#addPropertyChangeListener(java.lang.String,%20java.beans.PropertyChangeListener)), [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)), [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()), [getFocusTraversalKeys](http://docs.google.com/java/awt/Container.html#getFocusTraversalKeys(int)), [getFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#getFocusTraversalPolicy()), [getLayout](http://docs.google.com/java/awt/Container.html#getLayout()), [getMousePosition](http://docs.google.com/java/awt/Container.html#getMousePosition(boolean)), [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()), [paintComponents](http://docs.google.com/java/awt/Container.html#paintComponents(java.awt.Graphics)), [preferredSize](http://docs.google.com/java/awt/Container.html#preferredSize()), [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)), [processEvent](http://docs.google.com/java/awt/Container.html#processEvent(java.awt.AWTEvent)), [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)), [setFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [setFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicyProvider(boolean)), [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(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)), [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(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,%20byte,%20byte)), [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,%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()), [getBounds](http://docs.google.com/java/awt/Component.html#getBounds()), [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()), [getForeground](http://docs.google.com/java/awt/Component.html#getForeground()), [getGraphicsConfiguration](http://docs.google.com/java/awt/Component.html#getGraphicsConfiguration()), [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()), [getInputContext](http://docs.google.com/java/awt/Component.html#getInputContext()), [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()), [getLocale](http://docs.google.com/java/awt/Component.html#getLocale()), [getLocation](http://docs.google.com/java/awt/Component.html#getLocation()), [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()), [getToolkit](http://docs.google.com/java/awt/Component.html#getToolkit()), [getTreeLock](http://docs.google.com/java/awt/Component.html#getTreeLock()), [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()), [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()), [isPreferredSizeSet](http://docs.google.com/java/awt/Component.html#isPreferredSizeSet()), [isShowing](http://docs.google.com/java/awt/Component.html#isShowing()), [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)), [postEvent](http://docs.google.com/java/awt/Component.html#postEvent(java.awt.Event)), [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)), [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)), [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)), [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)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%20int)), [setBounds](http://docs.google.com/java/awt/Component.html#setBounds(java.awt.Rectangle)), [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)), [setFocusable](http://docs.google.com/java/awt/Component.html#setFocusable(boolean)), [setFocusTraversalKeysEnabled](http://docs.google.com/java/awt/Component.html#setFocusTraversalKeysEnabled(boolean)), [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)), [setName](http://docs.google.com/java/awt/Component.html#setName(java.lang.String)), [setSize](http://docs.google.com/java/awt/Component.html#setSize(java.awt.Dimension)), [setSize](http://docs.google.com/java/awt/Component.html#setSize(int,%20int)), [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**

The JRootPane instance that manages the content pane and optional menu bar for this internal frame, as well as the glass pane.

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

### rootPaneCheckingEnabled

protected boolean **rootPaneCheckingEnabled**

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

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

### closable

protected boolean **closable**

The frame can be closed.

### isClosed

protected boolean **isClosed**

The frame has been closed.

### maximizable

protected boolean **maximizable**

The frame can be expanded to the size of the desktop pane.

### isMaximum

protected boolean **isMaximum**

The frame has been expanded to its maximum size.

**See Also:**[maximizable](http://docs.google.com/javax/swing/JInternalFrame.html#maximizable)

### iconable

protected boolean **iconable**

The frame can "iconified" (shrunk down and displayed as an icon-image).

**See Also:**[JInternalFrame.JDesktopIcon](http://docs.google.com/javax/swing/JInternalFrame.JDesktopIcon.html), [setIconifiable(boolean)](http://docs.google.com/javax/swing/JInternalFrame.html#setIconifiable(boolean))

### isIcon

protected boolean **isIcon**

The frame has been iconified.

**See Also:**[isIcon()](http://docs.google.com/javax/swing/JInternalFrame.html#isIcon())

### resizable

protected boolean **resizable**

The frame's size can be changed.

### isSelected

protected boolean **isSelected**

The frame is currently selected.

### frameIcon

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

The icon shown in the top-left corner of this internal frame.

### title

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

The title displayed in this internal frame's title bar.

### desktopIcon

protected [JInternalFrame.JDesktopIcon](http://docs.google.com/javax/swing/JInternalFrame.JDesktopIcon.html) **desktopIcon**

The icon that is displayed when this internal frame is iconified.

**See Also:**[iconable](http://docs.google.com/javax/swing/JInternalFrame.html#iconable)

### CONTENT\_PANE\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **CONTENT\_PANE\_PROPERTY**

Bound property name.

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

### MENU\_BAR\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **MENU\_BAR\_PROPERTY**

Bound property name.

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

### TITLE\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **TITLE\_PROPERTY**

Bound property name.

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

### LAYERED\_PANE\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **LAYERED\_PANE\_PROPERTY**

Bound property name.

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

### ROOT\_PANE\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **ROOT\_PANE\_PROPERTY**

Bound property name.

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

### GLASS\_PANE\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **GLASS\_PANE\_PROPERTY**

Bound property name.

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

### FRAME\_ICON\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **FRAME\_ICON\_PROPERTY**

Bound property name.

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

### IS\_SELECTED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **IS\_SELECTED\_PROPERTY**

Constrained property name indicated that this frame has selected status.

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

### IS\_CLOSED\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **IS\_CLOSED\_PROPERTY**

Constrained property name indicating that the internal frame is closed.

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

### IS\_MAXIMUM\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **IS\_MAXIMUM\_PROPERTY**

Constrained property name indicating that the internal frame is maximized.

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

### IS\_ICON\_PROPERTY

public static final [String](http://docs.google.com/java/lang/String.html) **IS\_ICON\_PROPERTY**

Constrained property name indicating that the internal frame is iconified.

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

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

### JInternalFrame

public **JInternalFrame**()

Creates a non-resizable, non-closable, non-maximizable, non-iconifiable JInternalFrame with no title.

### JInternalFrame

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

Creates a non-resizable, non-closable, non-maximizable, non-iconifiable JInternalFrame with the specified title. Note that passing in a null title results in unspecified behavior and possibly an exception.

**Parameters:**title - the non-null String to display in the title bar

### JInternalFrame

public **JInternalFrame**([String](http://docs.google.com/java/lang/String.html) title,  
 boolean resizable)

Creates a non-closable, non-maximizable, non-iconifiable JInternalFrame with the specified title and resizability.

**Parameters:**title - the String to display in the title barresizable - if true, the internal frame can be resized

### JInternalFrame

public **JInternalFrame**([String](http://docs.google.com/java/lang/String.html) title,  
 boolean resizable,  
 boolean closable)

Creates a non-maximizable, non-iconifiable JInternalFrame with the specified title, resizability, and closability.

**Parameters:**title - the String to display in the title barresizable - if true, the internal frame can be resizedclosable - if true, the internal frame can be closed

### JInternalFrame

public **JInternalFrame**([String](http://docs.google.com/java/lang/String.html) title,  
 boolean resizable,  
 boolean closable,  
 boolean maximizable)

Creates a non-iconifiable JInternalFrame with the specified title, resizability, closability, and maximizability.

**Parameters:**title - the String to display in the title barresizable - if true, the internal frame can be resizedclosable - if true, the internal frame can be closedmaximizable - if true, the internal frame can be maximized

### JInternalFrame

public **JInternalFrame**([String](http://docs.google.com/java/lang/String.html) title,  
 boolean resizable,  
 boolean closable,  
 boolean maximizable,  
 boolean iconifiable)

Creates a JInternalFrame with the specified title, resizability, closability, maximizability, and iconifiability. All JInternalFrame constructors use this one.

**Parameters:**title - the String to display in the title barresizable - if true, the internal frame can be resizedclosable - if true, the internal frame can be closedmaximizable - if true, the internal frame can be maximizediconifiable - if true, the internal frame can be iconified

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

### createRootPane

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

Called by the constructor to set up the JRootPane.

**Returns:**a new JRootPane**See Also:**[JRootPane](http://docs.google.com/javax/swing/JRootPane.html)

### getUI

public [InternalFrameUI](http://docs.google.com/javax/swing/plaf/InternalFrameUI.html) **getUI**()

Returns the look-and-feel object that renders this component.

**Returns:**the InternalFrameUI object that renders this component

### setUI

public void **setUI**([InternalFrameUI](http://docs.google.com/javax/swing/plaf/InternalFrameUI.html) ui)

Sets the UI delegate for this JInternalFrame.

**Parameters:**ui - the UI delegate

### updateUI

public void **updateUI**()

Notification from the UIManager that the look and feel has changed. Replaces the current UI object with the latest version from the UIManager.

**Overrides:**[updateUI](http://docs.google.com/javax/swing/JComponent.html#updateUI()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **See Also:**[JComponent.updateUI()](http://docs.google.com/javax/swing/JComponent.html#updateUI())

### getUIClassID

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

Returns the name of the look-and-feel class that renders this component.

**Overrides:**[getUIClassID](http://docs.google.com/javax/swing/JComponent.html#getUIClassID()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**the string "InternalFrameUI"**See Also:**[JComponent.getUIClassID()](http://docs.google.com/javax/swing/JComponent.html#getUIClassID()), [UIDefaults.getUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/UIDefaults.html#getUI(javax.swing.JComponent))

### 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/JInternalFrame.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [setLayout(java.awt.LayoutManager)](http://docs.google.com/javax/swing/JInternalFrame.html#setLayout(java.awt.LayoutManager)), [setRootPaneCheckingEnabled(boolean)](http://docs.google.com/javax/swing/JInternalFrame.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 JInternalFrame.**See Also:**[addImpl(java.awt.Component, java.lang.Object, int)](http://docs.google.com/javax/swing/JInternalFrame.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [setLayout(java.awt.LayoutManager)](http://docs.google.com/javax/swing/JInternalFrame.html#setLayout(java.awt.LayoutManager)), [isRootPaneCheckingEnabled()](http://docs.google.com/javax/swing/JInternalFrame.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/JInternalFrame.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 a child of the JInternalFrame this will forward the call to the 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)

Ensures that, by default, the layout of this component cannot be set. 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/JInternalFrame.html#setRootPaneCheckingEnabled(boolean))

### getMenuBar

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public [JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) **getMenuBar**()

**Deprecated.** *As of Swing version 1.0.3, replaced by getJMenuBar().*

Returns the current JMenuBar for this JInternalFrame, or null if no menu bar has been set.

**Returns:**the current menu bar, or null if none has been set

### getJMenuBar

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

Returns the current JMenuBar for this JInternalFrame, or null if no menu bar has been set.

**Returns:**the JMenuBar used by this internal frame**See Also:**[setJMenuBar(javax.swing.JMenuBar)](http://docs.google.com/javax/swing/JInternalFrame.html#setJMenuBar(javax.swing.JMenuBar))

### setMenuBar

[@Deprecated](http://docs.google.com/java/lang/Deprecated.html)  
public void **setMenuBar**([JMenuBar](http://docs.google.com/javax/swing/JMenuBar.html) m)

**Deprecated.** *As of Swing version 1.0.3 replaced by setJMenuBar(JMenuBar m).*

Sets the menuBar property for this JInternalFrame.

**Parameters:**m - the JMenuBar to use in this internal frame**See Also:**[getJMenuBar()](http://docs.google.com/javax/swing/JInternalFrame.html#getJMenuBar())

### setJMenuBar

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

Sets the menuBar property for this JInternalFrame.

**Parameters:**m - the JMenuBar to use in this internal frame**See Also:**[getJMenuBar()](http://docs.google.com/javax/swing/JInternalFrame.html#getJMenuBar())

### getContentPane

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

Returns the content pane for this internal frame.

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

### setContentPane

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

Sets this JInternalFrame's contentPane property.

**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:**c - the content pane for this internal frame **Throws:** [IllegalComponentStateException](http://docs.google.com/java/awt/IllegalComponentStateException.html) - (a runtime exception) if the content pane parameter is null**See Also:**[RootPaneContainer.getContentPane()](http://docs.google.com/javax/swing/RootPaneContainer.html#getContentPane())

### getLayeredPane

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

Returns the layered pane for this internal frame.

**Specified by:**[getLayeredPane](http://docs.google.com/javax/swing/RootPaneContainer.html#getLayeredPane()) in interface [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html) **Returns:**a JLayeredPane object**See Also:**[RootPaneContainer.setLayeredPane(javax.swing.JLayeredPane)](http://docs.google.com/javax/swing/RootPaneContainer.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) layered)

Sets this JInternalFrame's layeredPane property.

**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:**layered - the JLayeredPane for this internal frame **Throws:** [IllegalComponentStateException](http://docs.google.com/java/awt/IllegalComponentStateException.html) - (a runtime exception) if the layered pane parameter is null**See Also:**[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 glass pane for this internal frame.

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

### setGlassPane

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

Sets this JInternalFrame's glassPane property.

**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:**glass - the glass pane for this internal frame**See Also:**[RootPaneContainer.getGlassPane()](http://docs.google.com/javax/swing/RootPaneContainer.html#getGlassPane())

### getRootPane

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

Returns the rootPane object for this internal frame.

**Specified by:**[getRootPane](http://docs.google.com/javax/swing/RootPaneContainer.html#getRootPane()) in interface [RootPaneContainer](http://docs.google.com/javax/swing/RootPaneContainer.html)**Overrides:**[getRootPane](http://docs.google.com/javax/swing/JComponent.html#getRootPane()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**the rootPane property**See Also:**[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 for this JInternalFrame. This method is called by the constructor.

**Parameters:**root - the new JRootPane object

### setClosable

public void **setClosable**(boolean b)

Sets whether this JInternalFrame can be closed by some user action.

**Parameters:**b - a boolean value, where true means this internal frame can be closed

### isClosable

public boolean **isClosable**()

Returns whether this JInternalFrame can be closed by some user action.

**Returns:**true if this internal frame can be closed

### isClosed

public boolean **isClosed**()

Returns whether this JInternalFrame is currently closed.

**Returns:**true if this internal frame is closed, false otherwise

### setClosed

public void **setClosed**(boolean b)  
 throws [PropertyVetoException](http://docs.google.com/java/beans/PropertyVetoException.html)

Closes this internal frame if the argument is true. Do not invoke this method with a false argument; the result of invoking setClosed(false) is unspecified.

If the internal frame is already closed, this method does nothing and returns immediately. Otherwise, this method begins by firing an INTERNAL\_FRAME\_CLOSING event. Then this method sets the closed property to true unless a listener vetoes the property change. This method finishes by making the internal frame invisible and unselected, and then firing an INTERNAL\_FRAME\_CLOSED event.

**Note:** To reuse an internal frame that has been closed, you must add it to a container (even if you never removed it from its previous container). Typically, this container will be the JDesktopPane that previously contained the internal frame.

**Parameters:**b - must be true **Throws:** [PropertyVetoException](http://docs.google.com/java/beans/PropertyVetoException.html) - when the attempt to set the property is vetoed by the JInternalFrame**See Also:**[isClosed()](http://docs.google.com/javax/swing/JInternalFrame.html#isClosed()), [setDefaultCloseOperation(int)](http://docs.google.com/javax/swing/JInternalFrame.html#setDefaultCloseOperation(int)), [dispose()](http://docs.google.com/javax/swing/JInternalFrame.html#dispose()), [InternalFrameEvent.INTERNAL\_FRAME\_CLOSING](http://docs.google.com/javax/swing/event/InternalFrameEvent.html#INTERNAL_FRAME_CLOSING)

### setResizable

public void **setResizable**(boolean b)

Sets whether the JInternalFrame can be resized by some user action.

**Parameters:**b - a boolean, where true means this internal frame can be resized

### isResizable

public boolean **isResizable**()

Returns whether the JInternalFrame can be resized by some user action.

**Returns:**true if this internal frame can be resized, false otherwise

### setIconifiable

public void **setIconifiable**(boolean b)

Sets the iconable property, which must be true for the user to be able to make the JInternalFrame an icon. Some look and feels might not implement iconification; they will ignore this property.

**Parameters:**b - a boolean, where true means this internal frame can be iconified

### isIconifiable

public boolean **isIconifiable**()

Gets the iconable property, which by default is false.

**Returns:**the value of the iconable property.**See Also:**[setIconifiable(boolean)](http://docs.google.com/javax/swing/JInternalFrame.html#setIconifiable(boolean))

### isIcon

public boolean **isIcon**()

Returns whether the JInternalFrame is currently iconified.

**Returns:**true if this internal frame is iconified

### setIcon

public void **setIcon**(boolean b)  
 throws [PropertyVetoException](http://docs.google.com/java/beans/PropertyVetoException.html)

Iconifies or de-iconifies this internal frame, if the look and feel supports iconification. If the internal frame's state changes to iconified, this method fires an INTERNAL\_FRAME\_ICONIFIED event. If the state changes to de-iconified, an INTERNAL\_FRAME\_DEICONIFIED event is fired.

**Parameters:**b - a boolean, where true means to iconify this internal frame and false means to de-iconify it **Throws:** [PropertyVetoException](http://docs.google.com/java/beans/PropertyVetoException.html) - when the attempt to set the property is vetoed by the JInternalFrame**See Also:**[InternalFrameEvent.INTERNAL\_FRAME\_ICONIFIED](http://docs.google.com/javax/swing/event/InternalFrameEvent.html#INTERNAL_FRAME_ICONIFIED), [InternalFrameEvent.INTERNAL\_FRAME\_DEICONIFIED](http://docs.google.com/javax/swing/event/InternalFrameEvent.html#INTERNAL_FRAME_DEICONIFIED)

### setMaximizable

public void **setMaximizable**(boolean b)

Sets the maximizable property, which determines whether the JInternalFrame can be maximized by some user action. Some look and feels might not support maximizing internal frames; they will ignore this property.

**Parameters:**b - true to specify that this internal frame should be maximizable; false to specify that it should not be

### isMaximizable

public boolean **isMaximizable**()

Gets the value of the maximizable property.

**Returns:**the value of the maximizable property**See Also:**[setMaximizable(boolean)](http://docs.google.com/javax/swing/JInternalFrame.html#setMaximizable(boolean))

### isMaximum

public boolean **isMaximum**()

Returns whether the JInternalFrame is currently maximized.

**Returns:**true if this internal frame is maximized, false otherwise

### setMaximum

public void **setMaximum**(boolean b)  
 throws [PropertyVetoException](http://docs.google.com/java/beans/PropertyVetoException.html)

Maximizes and restores this internal frame. A maximized frame is resized to fully fit the JDesktopPane area associated with the JInternalFrame. A restored frame's size is set to the JInternalFrame's actual size.

**Parameters:**b - a boolean, where true maximizes this internal frame and false restores it **Throws:** [PropertyVetoException](http://docs.google.com/java/beans/PropertyVetoException.html) - when the attempt to set the property is vetoed by the JInternalFrame

### getTitle

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

Returns the title of the JInternalFrame.

**Returns:**a String containing this internal frame's title**See Also:**[setTitle(java.lang.String)](http://docs.google.com/javax/swing/JInternalFrame.html#setTitle(java.lang.String))

### setTitle

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

Sets the JInternalFrame title. title may have a null value.

**Parameters:**title - the String to display in the title bar**See Also:**[getTitle()](http://docs.google.com/javax/swing/JInternalFrame.html#getTitle())

### setSelected

public void **setSelected**(boolean selected)  
 throws [PropertyVetoException](http://docs.google.com/java/beans/PropertyVetoException.html)

Selects or deselects the internal frame if it's showing. A JInternalFrame normally draws its title bar differently if it is the selected frame, which indicates to the user that this internal frame has the focus. When this method changes the state of the internal frame from deselected to selected, it fires an InternalFrameEvent.INTERNAL\_FRAME\_ACTIVATED event. If the change is from selected to deselected, an InternalFrameEvent.INTERNAL\_FRAME\_DEACTIVATED event is fired.

**Parameters:**selected - a boolean, where true means this internal frame should become selected (currently active) and false means it should become deselected **Throws:** [PropertyVetoException](http://docs.google.com/java/beans/PropertyVetoException.html) - when the attempt to set the property is vetoed by the JInternalFrame**See Also:**[Component.isShowing()](http://docs.google.com/java/awt/Component.html#isShowing()), [InternalFrameEvent.INTERNAL\_FRAME\_ACTIVATED](http://docs.google.com/javax/swing/event/InternalFrameEvent.html#INTERNAL_FRAME_ACTIVATED), [InternalFrameEvent.INTERNAL\_FRAME\_DEACTIVATED](http://docs.google.com/javax/swing/event/InternalFrameEvent.html#INTERNAL_FRAME_DEACTIVATED)

### isSelected

public boolean **isSelected**()

Returns whether the JInternalFrame is the currently "selected" or active frame.

**Returns:**true if this internal frame is currently selected (active)**See Also:**[setSelected(boolean)](http://docs.google.com/javax/swing/JInternalFrame.html#setSelected(boolean))

### setFrameIcon

public void **setFrameIcon**([Icon](http://docs.google.com/javax/swing/Icon.html) icon)

Sets an image to be displayed in the titlebar of this internal frame (usually in the top-left corner). This image is not the desktopIcon object, which is the image displayed in the JDesktop when this internal frame is iconified. Passing null to this function is valid, but the look and feel can choose the appropriate behavior for that situation, such as displaying no icon or a default icon for the look and feel.

**Parameters:**icon - the Icon to display in the title bar**See Also:**[getFrameIcon()](http://docs.google.com/javax/swing/JInternalFrame.html#getFrameIcon())

### getFrameIcon

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

Returns the image displayed in the title bar of this internal frame (usually in the top-left corner).

**Returns:**the Icon displayed in the title bar**See Also:**[setFrameIcon(javax.swing.Icon)](http://docs.google.com/javax/swing/JInternalFrame.html#setFrameIcon(javax.swing.Icon))

### moveToFront

public void **moveToFront**()

Convenience method that moves this component to position 0 if its parent is a JLayeredPane.

### moveToBack

public void **moveToBack**()

Convenience method that moves this component to position -1 if its parent is a JLayeredPane.

### getLastCursor

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

Returns the last Cursor that was set by the setCursor method that is not a resizable Cursor.

**Returns:**the last non-resizable Cursor**Since:** 1.6

### setCursor

public void **setCursor**([Cursor](http://docs.google.com/java/awt/Cursor.html) cursor)

Sets the cursor image to the specified cursor. This cursor image is displayed when the contains method for this component returns true for the current cursor location, and this Component is visible, displayable, and enabled. Setting the cursor of a Container causes that cursor to be displayed within all of the container's subcomponents, except for those that have a non-null cursor.

The method may have no visual effect if the Java platform implementation and/or the native system do not support changing the mouse cursor shape.

**Overrides:**[setCursor](http://docs.google.com/java/awt/Component.html#setCursor(java.awt.Cursor)) in class [Component](http://docs.google.com/java/awt/Component.html) **Parameters:**cursor - One of the constants defined by the Cursor class; if this parameter is null then this component will inherit the cursor of its parent**Since:** 1.6 **See Also:**[Component.isEnabled()](http://docs.google.com/java/awt/Component.html#isEnabled()), [Component.isShowing()](http://docs.google.com/java/awt/Component.html#isShowing()), [Component.getCursor()](http://docs.google.com/java/awt/Component.html#getCursor()), [Component.contains(int, int)](http://docs.google.com/java/awt/Component.html#contains(int,%20int)), [Toolkit.createCustomCursor(java.awt.Image, java.awt.Point, java.lang.String)](http://docs.google.com/java/awt/Toolkit.html#createCustomCursor(java.awt.Image,%20java.awt.Point,%20java.lang.String)), [Cursor](http://docs.google.com/java/awt/Cursor.html)

### setLayer

public void **setLayer**([Integer](http://docs.google.com/java/lang/Integer.html) layer)

Convenience method for setting the layer attribute of this component.

**Parameters:**layer - an Integer object specifying this frame's desktop layer**See Also:**[JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html)

### setLayer

public void **setLayer**(int layer)

Convenience method for setting the layer attribute of this component. The method setLayer(Integer) should be used for layer values predefined in JLayeredPane. When using setLayer(int), care must be taken not to accidentally clash with those values.

**Parameters:**layer - an integer specifying this internal frame's desktop layer**Since:** 1.3 **See Also:**[setLayer(Integer)](http://docs.google.com/javax/swing/JInternalFrame.html#setLayer(java.lang.Integer)), [JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html)

### getLayer

public int **getLayer**()

Convenience method for getting the layer attribute of this component.

**Returns:**an Integer object specifying this frame's desktop layer**See Also:**[JLayeredPane](http://docs.google.com/javax/swing/JLayeredPane.html)

### getDesktopPane

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

Convenience method that searches the ancestor hierarchy for a JDesktop instance. If JInternalFrame finds none, the desktopIcon tree is searched.

**Returns:**the JDesktopPane this internal frame belongs to, or null if none is found

### setDesktopIcon

public void **setDesktopIcon**([JInternalFrame.JDesktopIcon](http://docs.google.com/javax/swing/JInternalFrame.JDesktopIcon.html) d)

Sets the JDesktopIcon associated with this JInternalFrame.

**Parameters:**d - the JDesktopIcon to display on the desktop**See Also:**[getDesktopIcon()](http://docs.google.com/javax/swing/JInternalFrame.html#getDesktopIcon())

### getDesktopIcon

public [JInternalFrame.JDesktopIcon](http://docs.google.com/javax/swing/JInternalFrame.JDesktopIcon.html) **getDesktopIcon**()

Returns the JDesktopIcon used when this JInternalFrame is iconified.

**Returns:**the JDesktopIcon displayed on the desktop**See Also:**[setDesktopIcon(javax.swing.JInternalFrame.JDesktopIcon)](http://docs.google.com/javax/swing/JInternalFrame.html#setDesktopIcon(javax.swing.JInternalFrame.JDesktopIcon))

### getNormalBounds

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

If the JInternalFrame is not in maximized state, returns getBounds(); otherwise, returns the bounds that the JInternalFrame would be restored to.

**Returns:**a Rectangle containing the bounds of this frame when in the normal state**Since:** 1.3

### setNormalBounds

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

Sets the normal bounds for this internal frame, the bounds that this internal frame would be restored to from its maximized state. This method is intended for use only by desktop managers.

**Parameters:**r - the bounds that this internal frame should be restored to**Since:** 1.3

### getFocusOwner

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

If this JInternalFrame is active, returns the child that has focus. Otherwise, returns null.

**Returns:**the component with focus, or null if no children have focus**Since:** 1.3

### getMostRecentFocusOwner

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

Returns the child component of this JInternalFrame that will receive the focus when this JInternalFrame is selected. If this JInternalFrame is currently selected, this method returns the same component as the getFocusOwner method. If this JInternalFrame is not selected, then the child component that most recently requested focus will be returned. If no child component has ever requested focus, then this JInternalFrame's initial focusable component is returned. If no such child exists, then this JInternalFrame's default component to focus is returned.

**Returns:**the child component that will receive focus when this JInternalFrame is selected**Since:** 1.4 **See Also:**[getFocusOwner()](http://docs.google.com/javax/swing/JInternalFrame.html#getFocusOwner()), [isSelected](http://docs.google.com/javax/swing/JInternalFrame.html#isSelected)

### restoreSubcomponentFocus

public void **restoreSubcomponentFocus**()

Requests the internal frame to restore focus to the last subcomponent that had focus. This is used by the UI when the user selected this internal frame -- for example, by clicking on the title bar.

**Since:** 1.3

### reshape

public void **reshape**(int x,  
 int y,  
 int width,  
 int height)

Moves and resizes this component. Unlike other components, this implementation also forces re-layout, so that frame decorations such as the title bar are always redisplayed.

**Overrides:**[reshape](http://docs.google.com/javax/swing/JComponent.html#reshape(int,%20int,%20int,%20int)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**x - an integer giving the component's new horizontal position measured in pixels from the left of its containery - an integer giving the component's new vertical position, measured in pixels from the bottom of its containerwidth - an integer giving the component's new width in pixelsheight - an integer giving the component's new height in pixels**See Also:**[Component.setBounds(int, int, int, int)](http://docs.google.com/java/awt/Component.html#setBounds(int,%20int,%20int,%20int))

### addInternalFrameListener

public void **addInternalFrameListener**([InternalFrameListener](http://docs.google.com/javax/swing/event/InternalFrameListener.html) l)

Adds the specified listener to receive internal frame events from this internal frame.

**Parameters:**l - the internal frame listener

### removeInternalFrameListener

public void **removeInternalFrameListener**([InternalFrameListener](http://docs.google.com/javax/swing/event/InternalFrameListener.html) l)

Removes the specified internal frame listener so that it no longer receives internal frame events from this internal frame.

**Parameters:**l - the internal frame listener

### getInternalFrameListeners

public [InternalFrameListener](http://docs.google.com/javax/swing/event/InternalFrameListener.html)[] **getInternalFrameListeners**()

Returns an array of all the InternalFrameListeners added to this JInternalFrame with addInternalFrameListener.

**Returns:**all of the InternalFrameListeners added or an empty array if no listeners have been added**Since:** 1.4 **See Also:**[addInternalFrameListener(javax.swing.event.InternalFrameListener)](http://docs.google.com/javax/swing/JInternalFrame.html#addInternalFrameListener(javax.swing.event.InternalFrameListener))

### fireInternalFrameEvent

protected void **fireInternalFrameEvent**(int id)

Fires an internal frame event.

**Parameters:**id - the type of the event being fired; one of the following:

* InternalFrameEvent.INTERNAL\_FRAME\_OPENED
* InternalFrameEvent.INTERNAL\_FRAME\_CLOSING
* InternalFrameEvent.INTERNAL\_FRAME\_CLOSED
* InternalFrameEvent.INTERNAL\_FRAME\_ICONIFIED
* InternalFrameEvent.INTERNAL\_FRAME\_DEICONIFIED
* InternalFrameEvent.INTERNAL\_FRAME\_ACTIVATED
* InternalFrameEvent.INTERNAL\_FRAME\_DEACTIVATED

If the event type is not one of the above, nothing happens.

### doDefaultCloseAction

public void **doDefaultCloseAction**()

Fires an INTERNAL\_FRAME\_CLOSING event and then performs the action specified by the internal frame's default close operation. This method is typically invoked by the look-and-feel-implemented action handler for the internal frame's close button.

**Since:** 1.3 **See Also:**[setDefaultCloseOperation(int)](http://docs.google.com/javax/swing/JInternalFrame.html#setDefaultCloseOperation(int)), [InternalFrameEvent.INTERNAL\_FRAME\_CLOSING](http://docs.google.com/javax/swing/event/InternalFrameEvent.html#INTERNAL_FRAME_CLOSING)

### setDefaultCloseOperation

public void **setDefaultCloseOperation**(int operation)

Sets the operation that will happen by default when the user initiates a "close" on this internal frame. The possible choices are:

DO\_NOTHING\_ON\_CLOSE Do nothing. This requires the program to handle the operation in the windowClosing method of a registered InternalFrameListener object. HIDE\_ON\_CLOSE Automatically make the internal frame invisible. DISPOSE\_ON\_CLOSE Automatically dispose of the internal frame.

The default value is DISPOSE\_ON\_CLOSE. Before performing the specified close operation, the internal frame fires an INTERNAL\_FRAME\_CLOSING event.

**Parameters:**operation - one of the following constants defined in javax.swing.WindowConstants (an interface implemented by JInternalFrame): DO\_NOTHING\_ON\_CLOSE, HIDE\_ON\_CLOSE, or DISPOSE\_ON\_CLOSE**See Also:**[addInternalFrameListener(javax.swing.event.InternalFrameListener)](http://docs.google.com/javax/swing/JInternalFrame.html#addInternalFrameListener(javax.swing.event.InternalFrameListener)), [getDefaultCloseOperation()](http://docs.google.com/javax/swing/JInternalFrame.html#getDefaultCloseOperation()), [JComponent.setVisible(boolean)](http://docs.google.com/javax/swing/JComponent.html#setVisible(boolean)), [dispose()](http://docs.google.com/javax/swing/JInternalFrame.html#dispose()), [InternalFrameEvent.INTERNAL\_FRAME\_CLOSING](http://docs.google.com/javax/swing/event/InternalFrameEvent.html#INTERNAL_FRAME_CLOSING)

### getDefaultCloseOperation

public int **getDefaultCloseOperation**()

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

**Returns:**the operation that will occur when the user closes the internal frame**See Also:**[setDefaultCloseOperation(int)](http://docs.google.com/javax/swing/JInternalFrame.html#setDefaultCloseOperation(int))

### pack

public void **pack**()

Causes subcomponents of this JInternalFrame to be laid out at their preferred size. Internal frames that are iconized or maximized are first restored and then packed. If the internal frame is unable to be restored its state is not changed and will not be packed.

**See Also:**[Window.pack()](http://docs.google.com/java/awt/Window.html#pack())

### show

public void **show**()

If the internal frame is not visible, brings the internal frame to the front, makes it visible, and attempts to select it. The first time the internal frame is made visible, this method also fires an INTERNAL\_FRAME\_OPENED event. This method does nothing if the internal frame is already visible. Invoking this method has the same result as invoking setVisible(true).

**Overrides:**[show](http://docs.google.com/java/awt/Component.html#show()) in class [Component](http://docs.google.com/java/awt/Component.html) **See Also:**[moveToFront()](http://docs.google.com/javax/swing/JInternalFrame.html#moveToFront()), [setSelected(boolean)](http://docs.google.com/javax/swing/JInternalFrame.html#setSelected(boolean)), [InternalFrameEvent.INTERNAL\_FRAME\_OPENED](http://docs.google.com/javax/swing/event/InternalFrameEvent.html#INTERNAL_FRAME_OPENED), [JComponent.setVisible(boolean)](http://docs.google.com/javax/swing/JComponent.html#setVisible(boolean))

### hide

public void **hide**()

**Overrides:**[hide](http://docs.google.com/java/awt/Component.html#hide()) in class [Component](http://docs.google.com/java/awt/Component.html)

### dispose

public void **dispose**()

Makes this internal frame invisible, unselected, and closed. If the frame is not already closed, this method fires an INTERNAL\_FRAME\_CLOSED event. The results of invoking this method are similar to setClosed(true), but dispose always succeeds in closing the internal frame and does not fire an INTERNAL\_FRAME\_CLOSING event.

**See Also:**[InternalFrameEvent.INTERNAL\_FRAME\_CLOSED](http://docs.google.com/javax/swing/event/InternalFrameEvent.html#INTERNAL_FRAME_CLOSED), [JComponent.setVisible(boolean)](http://docs.google.com/javax/swing/JComponent.html#setVisible(boolean)), [setSelected(boolean)](http://docs.google.com/javax/swing/JInternalFrame.html#setSelected(boolean)), [setClosed(boolean)](http://docs.google.com/javax/swing/JInternalFrame.html#setClosed(boolean))

### toFront

public void **toFront**()

Brings this internal frame to the front. Places this internal frame at the top of the stacking order and makes the corresponding adjustment to other visible internal frames.

**See Also:**[Window.toFront()](http://docs.google.com/java/awt/Window.html#toFront()), [moveToFront()](http://docs.google.com/javax/swing/JInternalFrame.html#moveToFront())

### toBack

public void **toBack**()

Sends this internal frame to the back. Places this internal frame at the bottom of the stacking order and makes the corresponding adjustment to other visible internal frames.

**See Also:**[Window.toBack()](http://docs.google.com/java/awt/Window.html#toBack()), [moveToBack()](http://docs.google.com/javax/swing/JInternalFrame.html#moveToBack())

### setFocusCycleRoot

public final void **setFocusCycleRoot**(boolean focusCycleRoot)

Does nothing because JInternalFrames must always be roots of a focus traversal cycle.

**Overrides:**[setFocusCycleRoot](http://docs.google.com/java/awt/Container.html#setFocusCycleRoot(boolean)) in class [Container](http://docs.google.com/java/awt/Container.html) **Parameters:**focusCycleRoot - this value is ignored**Since:** 1.4 **See Also:**[isFocusCycleRoot()](http://docs.google.com/javax/swing/JInternalFrame.html#isFocusCycleRoot()), [Container.setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [Container.getFocusTraversalPolicy()](http://docs.google.com/java/awt/Container.html#getFocusTraversalPolicy())

### isFocusCycleRoot

public final boolean **isFocusCycleRoot**()

Always returns true because all JInternalFrames must be roots of a focus traversal cycle.

**Overrides:**[isFocusCycleRoot](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot()) in class [Container](http://docs.google.com/java/awt/Container.html) **Returns:**true**Since:** 1.4 **See Also:**[setFocusCycleRoot(boolean)](http://docs.google.com/javax/swing/JInternalFrame.html#setFocusCycleRoot(boolean)), [Container.setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [Container.getFocusTraversalPolicy()](http://docs.google.com/java/awt/Container.html#getFocusTraversalPolicy())

### getFocusCycleRootAncestor

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

Always returns null because JInternalFrames must always be roots of a focus traversal cycle.

**Overrides:**[getFocusCycleRootAncestor](http://docs.google.com/java/awt/Component.html#getFocusCycleRootAncestor()) in class [Component](http://docs.google.com/java/awt/Component.html) **Returns:**null**Since:** 1.4 **See Also:**[Container.isFocusCycleRoot()](http://docs.google.com/java/awt/Container.html#isFocusCycleRoot())

### getWarningString

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

Gets the warning string that is displayed with this internal frame. Since an internal frame is always secure (since it's fully contained within a window that might need a warning string) this method always returns null.

**Returns:**null**See Also:**[Window.getWarningString()](http://docs.google.com/java/awt/Window.html#getWarningString())

### paramString

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

Returns a string representation of this JInternalFrame. 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/javax/swing/JComponent.html#paramString()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**a string representation of this JInternalFrame

### paintComponent

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

Overridden to allow optimized painting when the internal frame is being dragged.

**Overrides:**[paintComponent](http://docs.google.com/javax/swing/JComponent.html#paintComponent(java.awt.Graphics)) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Parameters:**g - the Graphics object to protect**See Also:**[JComponent.paint(java.awt.Graphics)](http://docs.google.com/javax/swing/JComponent.html#paint(java.awt.Graphics)), [ComponentUI](http://docs.google.com/javax/swing/plaf/ComponentUI.html)

### getAccessibleContext

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

Gets the AccessibleContext associated with this JInternalFrame. For internal frames, the AccessibleContext takes the form of an AccessibleJInternalFrame object. A new AccessibleJInternalFrame 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/javax/swing/JComponent.html#getAccessibleContext()) in class [JComponent](http://docs.google.com/javax/swing/JComponent.html) **Returns:**an AccessibleJInternalFrame that serves as the AccessibleContext of this JInternalFrame**See Also:**[JInternalFrame.AccessibleJInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.AccessibleJInternalFrame.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JInternalFrame.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/JFrame.AccessibleJFrame.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JInternalFrame.AccessibleJInternalFrame.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JInternalFrame.html)    [**NO FRAMES**](http://docs.google.com/JInternalFrame.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#lnxbz9) | DETAIL: [FIELD](#z337ya) | [CONSTR](#2u6wntf) | [METHOD](#46r0co2) |

[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).