| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JOptionPane.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/JMenuItem.AccessibleJMenuItem.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JOptionPane.AccessibleJOptionPane.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JOptionPane.html)    [**NO FRAMES**](http://docs.google.com/JOptionPane.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | [FIELD](#4d34og8) | [CONSTR](#26in1rg) | [METHOD](#lnxbz9) | DETAIL: [FIELD](#z337ya) | [CONSTR](#4k668n3) | [METHOD](#4bvk7pj) |

## **javax.swing**

Class JOptionPane

[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.JOptionPane**

**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)

public class **JOptionPane**extends [JComponent](http://docs.google.com/javax/swing/JComponent.html)implements [Accessible](http://docs.google.com/javax/accessibility/Accessible.html)

JOptionPane makes it easy to pop up a standard dialog box that prompts users for a value or informs them of something. For information about using JOptionPane, see [How to Make Dialogs](http://java.sun.com/docs/books/tutorial/uiswing/components/dialog.html), a section in *The Java Tutorial*.

While the JOptionPane class may appear complex because of the large number of methods, almost all uses of this class are one-line calls to one of the static showXxxDialog methods shown below:

| Method Name | Description |
| --- | --- |
| showConfirmDialog | Asks a confirming question, like yes/no/cancel. |
| showInputDialog | Prompt for some input. |
| showMessageDialog | Tell the user about something that has happened. |
| showOptionDialog | The Grand Unification of the above three. |

Each of these methods also comes in a showInternalXXX flavor, which uses an internal frame to hold the dialog box (see [JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html)). Multiple convenience methods have also been defined -- overloaded versions of the basic methods that use different parameter lists.

All dialogs are modal. Each showXxxDialog method blocks the caller until the user's interaction is complete.

| icon | message |
| --- | --- |
| input value |
| option buttons | |

The basic appearance of one of these dialog boxes is generally similar to the picture at the right, although the various look-and-feels are ultimately responsible for the final result. In particular, the look-and-feels will adjust the layout to accommodate the option pane's ComponentOrientation property.

**Parameters:**

The parameters to these methods follow consistent patterns:

parentComponent Defines the Component that is to be the parent of this dialog box. It is used in two ways: the Frame that contains it is used as the Frame parent for the dialog box, and its screen coordinates are used in the placement of the dialog box. In general, the dialog box is placed just below the component. This parameter may be null, in which case a default Frame is used as the parent, and the dialog will be centered on the screen (depending on the L&F). message A descriptive message to be placed in the dialog box. In the most common usage, message is just a String or String constant. However, the type of this parameter is actually Object. Its interpretation depends on its type: Object[]An array of objects is interpreted as a series of messages (one per object) arranged in a vertical stack. The interpretation is recursive -- each object in the array is interpreted according to its type. ComponentThe Component is displayed in the dialog. IconThe Icon is wrapped in a JLabel and displayed in the dialog. othersThe object is converted to a String by calling its toString method. The result is wrapped in a JLabel and displayed. messageTypeDefines the style of the message. The Look and Feel manager may lay out the dialog differently depending on this value, and will often provide a default icon. The possible values are:

* ERROR\_MESSAGE
* INFORMATION\_MESSAGE
* WARNING\_MESSAGE
* QUESTION\_MESSAGE
* PLAIN\_MESSAGE

optionTypeDefines the set of option buttons that appear at the bottom of the dialog box:

* DEFAULT\_OPTION
* YES\_NO\_OPTION
* YES\_NO\_CANCEL\_OPTION
* OK\_CANCEL\_OPTION

You aren't limited to this set of option buttons. You can provide any buttons you want using the options parameter. optionsA more detailed description of the set of option buttons that will appear at the bottom of the dialog box. The usual value for the options parameter is an array of Strings. But the parameter type is an array of Objects. A button is created for each object depending on its type: ComponentThe component is added to the button row directly. IconA JButton is created with this as its label. otherThe Object is converted to a string using its toString method and the result is used to label a JButton. iconA decorative icon to be placed in the dialog box. A default value for this is determined by the messageType parameter. titleThe title for the dialog box. initialValueThe default selection (input value).

When the selection is changed, setValue is invoked, which generates a PropertyChangeEvent.

If a JOptionPane has configured to all input setWantsInput the bound property JOptionPane.INPUT\_VALUE\_PROPERTY can also be listened to, to determine when the user has input or selected a value.

When one of the showXxxDialog methods returns an integer, the possible values are:

* YES\_OPTION
* NO\_OPTION
* CANCEL\_OPTION
* OK\_OPTION
* CLOSED\_OPTION

**Examples:** Show an error dialog that displays the message, 'alert': JOptionPane.showMessageDialog(null, "alert", "alert", JOptionPane.ERROR\_MESSAGE);

Show an internal information dialog with the message, 'information': JOptionPane.showInternalMessageDialog(frame, "information",

"information", JOptionPane.INFORMATION\_MESSAGE);

Show an information panel with the options yes/no and message 'choose one': JOptionPane.showConfirmDialog(null,

"choose one", "choose one", JOptionPane.YES\_NO\_OPTION);

Show an internal information dialog with the options yes/no/cancel and message 'please choose one' and title information: JOptionPane.showInternalConfirmDialog(frame,

"please choose one", "information",

JOptionPane.YES\_NO\_CANCEL\_OPTION, JOptionPane.INFORMATION\_MESSAGE);

Show a warning dialog with the options OK, CANCEL, title 'Warning', and message 'Click OK to continue': Object[] options = { "OK", "CANCEL" };

JOptionPane.showOptionDialog(null, "Click OK to continue", "Warning",

JOptionPane.DEFAULT\_OPTION, JOptionPane.WARNING\_MESSAGE,

null, options, options[0]);

Show a dialog asking the user to type in a String: String inputValue = JOptionPane.showInputDialog("Please input a value");

Show a dialog asking the user to select a String: Object[] possibleValues = { "First", "Second", "Third" };

Object selectedValue = JOptionPane.showInputDialog(null,

"Choose one", "Input",

JOptionPane.INFORMATION\_MESSAGE, null,

possibleValues, possibleValues[0]);

**Direct Use:**

To create and use an JOptionPane directly, the standard pattern is roughly as follows:

JOptionPane pane = new JOptionPane(*arguments*);  
 pane.set*.Xxxx(...); // Configure*  
 JDialog dialog = pane.createDialog(*parentComponent, title*);  
 dialog.show();  
 Object selectedValue = pane.getValue();  
 if(selectedValue == null)  
 return CLOSED\_OPTION;  
 *//If there is* ***not*** *an array of option buttons:*  
 if(options == null) {  
 if(selectedValue instanceof Integer)  
 return ((Integer)selectedValue).intValue();  
 return CLOSED\_OPTION;  
 }  
 *//If there is an array of option buttons:*  
 for(int counter = 0, maxCounter = options.length;  
 counter < maxCounter; counter++) {  
 if(options[counter].equals(selectedValue))  
 return counter;  
 }  
 return CLOSED\_OPTION;

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

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

| **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** | |
| --- | --- |
| static int | [**CANCEL\_OPTION**](http://docs.google.com/javax/swing/JOptionPane.html#CANCEL_OPTION)            Return value from class method if CANCEL is chosen. |
| static int | [**CLOSED\_OPTION**](http://docs.google.com/javax/swing/JOptionPane.html#CLOSED_OPTION)            Return value from class method if user closes window without selecting anything, more than likely this should be treated as either a CANCEL\_OPTION or NO\_OPTION. |
| static int | [**DEFAULT\_OPTION**](http://docs.google.com/javax/swing/JOptionPane.html#DEFAULT_OPTION)            Type meaning Look and Feel should not supply any options -- only use the options from the JOptionPane. |
| static int | [**ERROR\_MESSAGE**](http://docs.google.com/javax/swing/JOptionPane.html#ERROR_MESSAGE)            Used for error messages. |
| protected  [Icon](http://docs.google.com/javax/swing/Icon.html) | [**icon**](http://docs.google.com/javax/swing/JOptionPane.html#icon)            Icon used in pane. |
| static [String](http://docs.google.com/java/lang/String.html) | [**ICON\_PROPERTY**](http://docs.google.com/javax/swing/JOptionPane.html#ICON_PROPERTY)            Bound property name for icon. |
| static int | [**INFORMATION\_MESSAGE**](http://docs.google.com/javax/swing/JOptionPane.html#INFORMATION_MESSAGE)            Used for information messages. |
| static [String](http://docs.google.com/java/lang/String.html) | [**INITIAL\_SELECTION\_VALUE\_PROPERTY**](http://docs.google.com/javax/swing/JOptionPane.html#INITIAL_SELECTION_VALUE_PROPERTY)            Bound property name for initialSelectionValue. |
| static [String](http://docs.google.com/java/lang/String.html) | [**INITIAL\_VALUE\_PROPERTY**](http://docs.google.com/javax/swing/JOptionPane.html#INITIAL_VALUE_PROPERTY)            Bound property name for initialValue. |
| protected  [Object](http://docs.google.com/java/lang/Object.html) | [**initialSelectionValue**](http://docs.google.com/javax/swing/JOptionPane.html#initialSelectionValue)            Initial value to select in selectionValues. |
| protected  [Object](http://docs.google.com/java/lang/Object.html) | [**initialValue**](http://docs.google.com/javax/swing/JOptionPane.html#initialValue)            Value that should be initially selected in options. |
| static [String](http://docs.google.com/java/lang/String.html) | [**INPUT\_VALUE\_PROPERTY**](http://docs.google.com/javax/swing/JOptionPane.html#INPUT_VALUE_PROPERTY)            Bound property name for inputValue. |
| protected  [Object](http://docs.google.com/java/lang/Object.html) | [**inputValue**](http://docs.google.com/javax/swing/JOptionPane.html#inputValue)            Value the user has input. |
| protected  [Object](http://docs.google.com/java/lang/Object.html) | [**message**](http://docs.google.com/javax/swing/JOptionPane.html#message)            Message to display. |
| static [String](http://docs.google.com/java/lang/String.html) | [**MESSAGE\_PROPERTY**](http://docs.google.com/javax/swing/JOptionPane.html#MESSAGE_PROPERTY)            Bound property name for message. |
| static [String](http://docs.google.com/java/lang/String.html) | [**MESSAGE\_TYPE\_PROPERTY**](http://docs.google.com/javax/swing/JOptionPane.html#MESSAGE_TYPE_PROPERTY)            Bound property name for type. |
| protected  int | [**messageType**](http://docs.google.com/javax/swing/JOptionPane.html#messageType)            Message type. |
| static int | [**NO\_OPTION**](http://docs.google.com/javax/swing/JOptionPane.html#NO_OPTION)            Return value from class method if NO is chosen. |
| static int | [**OK\_CANCEL\_OPTION**](http://docs.google.com/javax/swing/JOptionPane.html#OK_CANCEL_OPTION)            Type used for showConfirmDialog. |
| static int | [**OK\_OPTION**](http://docs.google.com/javax/swing/JOptionPane.html#OK_OPTION)            Return value form class method if OK is chosen. |
| static [String](http://docs.google.com/java/lang/String.html) | [**OPTION\_TYPE\_PROPERTY**](http://docs.google.com/javax/swing/JOptionPane.html#OPTION_TYPE_PROPERTY)            Bound property name for optionType. |
| protected  [Object](http://docs.google.com/java/lang/Object.html)[] | [**options**](http://docs.google.com/javax/swing/JOptionPane.html#options)            Options to display to the user. |
| static [String](http://docs.google.com/java/lang/String.html) | [**OPTIONS\_PROPERTY**](http://docs.google.com/javax/swing/JOptionPane.html#OPTIONS_PROPERTY)            Bound property name for option. |
| protected  int | [**optionType**](http://docs.google.com/javax/swing/JOptionPane.html#optionType)            Option type, one of DEFAULT\_OPTION, YES\_NO\_OPTION, YES\_NO\_CANCEL\_OPTION or OK\_CANCEL\_OPTION. |
| static int | [**PLAIN\_MESSAGE**](http://docs.google.com/javax/swing/JOptionPane.html#PLAIN_MESSAGE)            No icon is used. |
| static int | [**QUESTION\_MESSAGE**](http://docs.google.com/javax/swing/JOptionPane.html#QUESTION_MESSAGE)            Used for questions. |
| static [String](http://docs.google.com/java/lang/String.html) | [**SELECTION\_VALUES\_PROPERTY**](http://docs.google.com/javax/swing/JOptionPane.html#SELECTION_VALUES_PROPERTY)            Bound property name for selectionValues. |
| protected  [Object](http://docs.google.com/java/lang/Object.html)[] | [**selectionValues**](http://docs.google.com/javax/swing/JOptionPane.html#selectionValues)            Array of values the user can choose from. |
| static [Object](http://docs.google.com/java/lang/Object.html) | [**UNINITIALIZED\_VALUE**](http://docs.google.com/javax/swing/JOptionPane.html#UNINITIALIZED_VALUE)            Indicates that the user has not yet selected a value. |
| protected  [Object](http://docs.google.com/java/lang/Object.html) | [**value**](http://docs.google.com/javax/swing/JOptionPane.html#value)            Currently selected value, will be a valid option, or UNINITIALIZED\_VALUE or null. |
| static [String](http://docs.google.com/java/lang/String.html) | [**VALUE\_PROPERTY**](http://docs.google.com/javax/swing/JOptionPane.html#VALUE_PROPERTY)            Bound property name for value. |
| static [String](http://docs.google.com/java/lang/String.html) | [**WANTS\_INPUT\_PROPERTY**](http://docs.google.com/javax/swing/JOptionPane.html#WANTS_INPUT_PROPERTY)            Bound property name for wantsInput. |
| protected  boolean | [**wantsInput**](http://docs.google.com/javax/swing/JOptionPane.html#wantsInput)            If true, a UI widget will be provided to the user to get input. |
| static int | [**WARNING\_MESSAGE**](http://docs.google.com/javax/swing/JOptionPane.html#WARNING_MESSAGE)            Used for warning messages. |
| static int | [**YES\_NO\_CANCEL\_OPTION**](http://docs.google.com/javax/swing/JOptionPane.html#YES_NO_CANCEL_OPTION)            Type used for showConfirmDialog. |
| static int | [**YES\_NO\_OPTION**](http://docs.google.com/javax/swing/JOptionPane.html#YES_NO_OPTION)            Type used for showConfirmDialog. |
| static int | [**YES\_OPTION**](http://docs.google.com/javax/swing/JOptionPane.html#YES_OPTION)            Return value from class method if YES is chosen. |

| **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 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** | |
| --- | --- |
| [**JOptionPane**](http://docs.google.com/javax/swing/JOptionPane.html#JOptionPane())()            Creates a JOptionPane with a test message. |
| [**JOptionPane**](http://docs.google.com/javax/swing/JOptionPane.html#JOptionPane(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) message)            Creates a instance of JOptionPane to display a message using the plain-message message type and the default options delivered by the UI. |
| [**JOptionPane**](http://docs.google.com/javax/swing/JOptionPane.html#JOptionPane(java.lang.Object,%20int))([Object](http://docs.google.com/java/lang/Object.html) message, int messageType)            Creates an instance of JOptionPane to display a message with the specified message type and the default options, |
| [**JOptionPane**](http://docs.google.com/javax/swing/JOptionPane.html#JOptionPane(java.lang.Object,%20int,%20int))([Object](http://docs.google.com/java/lang/Object.html) message, int messageType, int optionType)            Creates an instance of JOptionPane to display a message with the specified message type and options. |
| [**JOptionPane**](http://docs.google.com/javax/swing/JOptionPane.html#JOptionPane(java.lang.Object,%20int,%20int,%20javax.swing.Icon))([Object](http://docs.google.com/java/lang/Object.html) message, int messageType, int optionType, [Icon](http://docs.google.com/javax/swing/Icon.html) icon)            Creates an instance of JOptionPane to display a message with the specified message type, options, and icon. |
| [**JOptionPane**](http://docs.google.com/javax/swing/JOptionPane.html#JOptionPane(java.lang.Object,%20int,%20int,%20javax.swing.Icon,%20java.lang.Object%5B%5D))([Object](http://docs.google.com/java/lang/Object.html) message, int messageType, int optionType, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, [Object](http://docs.google.com/java/lang/Object.html)[] options)            Creates an instance of JOptionPane to display a message with the specified message type, icon, and options. |
| [**JOptionPane**](http://docs.google.com/javax/swing/JOptionPane.html#JOptionPane(java.lang.Object,%20int,%20int,%20javax.swing.Icon,%20java.lang.Object%5B%5D,%20java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) message, int messageType, int optionType, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, [Object](http://docs.google.com/java/lang/Object.html)[] options, [Object](http://docs.google.com/java/lang/Object.html) initialValue)            Creates an instance of JOptionPane to display a message with the specified message type, icon, and options, with the initially-selected option specified. |

| **Method Summary** | |
| --- | --- |
| [JDialog](http://docs.google.com/javax/swing/JDialog.html) | [**createDialog**](http://docs.google.com/javax/swing/JOptionPane.html#createDialog(java.awt.Component,%20java.lang.String))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [String](http://docs.google.com/java/lang/String.html) title)            Creates and returns a new JDialog wrapping this centered on the parentComponent in the parentComponent's frame. |
| [JDialog](http://docs.google.com/javax/swing/JDialog.html) | [**createDialog**](http://docs.google.com/javax/swing/JOptionPane.html#createDialog(java.lang.String))([String](http://docs.google.com/java/lang/String.html) title)            Creates and returns a new parentless JDialog with the specified title. |
| [JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) | [**createInternalFrame**](http://docs.google.com/javax/swing/JOptionPane.html#createInternalFrame(java.awt.Component,%20java.lang.String))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [String](http://docs.google.com/java/lang/String.html) title)            Creates and returns an instance of JInternalFrame. |
| [AccessibleContext](http://docs.google.com/javax/accessibility/AccessibleContext.html) | [**getAccessibleContext**](http://docs.google.com/javax/swing/JOptionPane.html#getAccessibleContext())()            Returns the AccessibleContext associated with this JOptionPane. |
| static [JDesktopPane](http://docs.google.com/javax/swing/JDesktopPane.html) | [**getDesktopPaneForComponent**](http://docs.google.com/javax/swing/JOptionPane.html#getDesktopPaneForComponent(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) parentComponent)            Returns the specified component's desktop pane. |
| static [Frame](http://docs.google.com/java/awt/Frame.html) | [**getFrameForComponent**](http://docs.google.com/javax/swing/JOptionPane.html#getFrameForComponent(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) parentComponent)            Returns the specified component's Frame. |
| [Icon](http://docs.google.com/javax/swing/Icon.html) | [**getIcon**](http://docs.google.com/javax/swing/JOptionPane.html#getIcon())()            Returns the icon this pane displays. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getInitialSelectionValue**](http://docs.google.com/javax/swing/JOptionPane.html#getInitialSelectionValue())()            Returns the input value that is displayed as initially selected to the user. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getInitialValue**](http://docs.google.com/javax/swing/JOptionPane.html#getInitialValue())()            Returns the initial value. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getInputValue**](http://docs.google.com/javax/swing/JOptionPane.html#getInputValue())()            Returns the value the user has input, if wantsInput is true. |
| int | [**getMaxCharactersPerLineCount**](http://docs.google.com/javax/swing/JOptionPane.html#getMaxCharactersPerLineCount())()            Returns the maximum number of characters to place on a line in a message. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getMessage**](http://docs.google.com/javax/swing/JOptionPane.html#getMessage())()            Returns the message-object this pane displays. |
| int | [**getMessageType**](http://docs.google.com/javax/swing/JOptionPane.html#getMessageType())()            Returns the message type. |
| [Object](http://docs.google.com/java/lang/Object.html)[] | [**getOptions**](http://docs.google.com/javax/swing/JOptionPane.html#getOptions())()            Returns the choices the user can make. |
| int | [**getOptionType**](http://docs.google.com/javax/swing/JOptionPane.html#getOptionType())()            Returns the type of options that are displayed. |
| static [Frame](http://docs.google.com/java/awt/Frame.html) | [**getRootFrame**](http://docs.google.com/javax/swing/JOptionPane.html#getRootFrame())()            Returns the Frame to use for the class methods in which a frame is not provided. |
| [Object](http://docs.google.com/java/lang/Object.html)[] | [**getSelectionValues**](http://docs.google.com/javax/swing/JOptionPane.html#getSelectionValues())()            Returns the input selection values. |
| [OptionPaneUI](http://docs.google.com/javax/swing/plaf/OptionPaneUI.html) | [**getUI**](http://docs.google.com/javax/swing/JOptionPane.html#getUI())()            Returns the UI object which implements the L&F for this component. |
| [String](http://docs.google.com/java/lang/String.html) | [**getUIClassID**](http://docs.google.com/javax/swing/JOptionPane.html#getUIClassID())()            Returns the name of the UI class that implements the L&F for this component. |
| [Object](http://docs.google.com/java/lang/Object.html) | [**getValue**](http://docs.google.com/javax/swing/JOptionPane.html#getValue())()            Returns the value the user has selected. |
| boolean | [**getWantsInput**](http://docs.google.com/javax/swing/JOptionPane.html#getWantsInput())()            Returns the value of the wantsInput property. |
| protected  [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/javax/swing/JOptionPane.html#paramString())()            Returns a string representation of this JOptionPane. |
| void | [**selectInitialValue**](http://docs.google.com/javax/swing/JOptionPane.html#selectInitialValue())()            Requests that the initial value be selected, which will set focus to the initial value. |
| void | [**setIcon**](http://docs.google.com/javax/swing/JOptionPane.html#setIcon(javax.swing.Icon))([Icon](http://docs.google.com/javax/swing/Icon.html) newIcon)            Sets the icon to display. |
| void | [**setInitialSelectionValue**](http://docs.google.com/javax/swing/JOptionPane.html#setInitialSelectionValue(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) newValue)            Sets the input value that is initially displayed as selected to the user. |
| void | [**setInitialValue**](http://docs.google.com/javax/swing/JOptionPane.html#setInitialValue(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) newInitialValue)            Sets the initial value that is to be enabled -- the Component that has the focus when the pane is initially displayed. |
| void | [**setInputValue**](http://docs.google.com/javax/swing/JOptionPane.html#setInputValue(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) newValue)            Sets the input value that was selected or input by the user. |
| void | [**setMessage**](http://docs.google.com/javax/swing/JOptionPane.html#setMessage(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) newMessage)            Sets the option pane's message-object. |
| void | [**setMessageType**](http://docs.google.com/javax/swing/JOptionPane.html#setMessageType(int))(int newType)            Sets the option pane's message type. |
| void | [**setOptions**](http://docs.google.com/javax/swing/JOptionPane.html#setOptions(java.lang.Object%5B%5D))([Object](http://docs.google.com/java/lang/Object.html)[] newOptions)            Sets the options this pane displays. |
| void | [**setOptionType**](http://docs.google.com/javax/swing/JOptionPane.html#setOptionType(int))(int newType)            Sets the options to display. |
| static void | [**setRootFrame**](http://docs.google.com/javax/swing/JOptionPane.html#setRootFrame(java.awt.Frame))([Frame](http://docs.google.com/java/awt/Frame.html) newRootFrame)            Sets the frame to use for class methods in which a frame is not provided. |
| void | [**setSelectionValues**](http://docs.google.com/javax/swing/JOptionPane.html#setSelectionValues(java.lang.Object%5B%5D))([Object](http://docs.google.com/java/lang/Object.html)[] newValues)            Sets the input selection values for a pane that provides the user with a list of items to choose from. |
| void | [**setUI**](http://docs.google.com/javax/swing/JOptionPane.html#setUI(javax.swing.plaf.OptionPaneUI))([OptionPaneUI](http://docs.google.com/javax/swing/plaf/OptionPaneUI.html) ui)            Sets the UI object which implements the L&F for this component. |
| void | [**setValue**](http://docs.google.com/javax/swing/JOptionPane.html#setValue(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) newValue)            Sets the value the user has chosen. |
| void | [**setWantsInput**](http://docs.google.com/javax/swing/JOptionPane.html#setWantsInput(boolean))(boolean newValue)            Sets the wantsInput property. |
| static int | [**showConfirmDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showConfirmDialog(java.awt.Component,%20java.lang.Object))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message)            Brings up a dialog with the options *Yes*, *No* and *Cancel*; with the title, **Select an Option**. |
| static int | [**showConfirmDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showConfirmDialog(java.awt.Component,%20java.lang.Object,%20java.lang.String,%20int))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message, [String](http://docs.google.com/java/lang/String.html) title, int optionType)            Brings up a dialog where the number of choices is determined by the optionType parameter. |
| static int | [**showConfirmDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showConfirmDialog(java.awt.Component,%20java.lang.Object,%20java.lang.String,%20int,%20int))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message, [String](http://docs.google.com/java/lang/String.html) title, int optionType, int messageType)            Brings up a dialog where the number of choices is determined by the optionType parameter, where the messageType parameter determines the icon to display. |
| static int | [**showConfirmDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showConfirmDialog(java.awt.Component,%20java.lang.Object,%20java.lang.String,%20int,%20int,%20javax.swing.Icon))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message, [String](http://docs.google.com/java/lang/String.html) title, int optionType, int messageType, [Icon](http://docs.google.com/javax/swing/Icon.html) icon)            Brings up a dialog with a specified icon, where the number of choices is determined by the optionType parameter. |
| static [String](http://docs.google.com/java/lang/String.html) | [**showInputDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showInputDialog(java.awt.Component,%20java.lang.Object))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message)            Shows a question-message dialog requesting input from the user parented to parentComponent. |
| static [String](http://docs.google.com/java/lang/String.html) | [**showInputDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showInputDialog(java.awt.Component,%20java.lang.Object,%20java.lang.Object))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message, [Object](http://docs.google.com/java/lang/Object.html) initialSelectionValue)            Shows a question-message dialog requesting input from the user and parented to parentComponent. |
| static [String](http://docs.google.com/java/lang/String.html) | [**showInputDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showInputDialog(java.awt.Component,%20java.lang.Object,%20java.lang.String,%20int))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message, [String](http://docs.google.com/java/lang/String.html) title, int messageType)            Shows a dialog requesting input from the user parented to parentComponent with the dialog having the title title and message type messageType. |
| static [Object](http://docs.google.com/java/lang/Object.html) | [**showInputDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showInputDialog(java.awt.Component,%20java.lang.Object,%20java.lang.String,%20int,%20javax.swing.Icon,%20java.lang.Object%5B%5D,%20java.lang.Object))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message, [String](http://docs.google.com/java/lang/String.html) title, int messageType, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, [Object](http://docs.google.com/java/lang/Object.html)[] selectionValues, [Object](http://docs.google.com/java/lang/Object.html) initialSelectionValue)            Prompts the user for input in a blocking dialog where the initial selection, possible selections, and all other options can be specified. |
| static [String](http://docs.google.com/java/lang/String.html) | [**showInputDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showInputDialog(java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) message)            Shows a question-message dialog requesting input from the user. |
| static [String](http://docs.google.com/java/lang/String.html) | [**showInputDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showInputDialog(java.lang.Object,%20java.lang.Object))([Object](http://docs.google.com/java/lang/Object.html) message, [Object](http://docs.google.com/java/lang/Object.html) initialSelectionValue)            Shows a question-message dialog requesting input from the user, with the input value initialized to initialSelectionValue. |
| static int | [**showInternalConfirmDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showInternalConfirmDialog(java.awt.Component,%20java.lang.Object))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message)            Brings up an internal dialog panel with the options *Yes*, *No* and *Cancel*; with the title, **Select an Option**. |
| static int | [**showInternalConfirmDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showInternalConfirmDialog(java.awt.Component,%20java.lang.Object,%20java.lang.String,%20int))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message, [String](http://docs.google.com/java/lang/String.html) title, int optionType)            Brings up a internal dialog panel where the number of choices is determined by the optionType parameter. |
| static int | [**showInternalConfirmDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showInternalConfirmDialog(java.awt.Component,%20java.lang.Object,%20java.lang.String,%20int,%20int))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message, [String](http://docs.google.com/java/lang/String.html) title, int optionType, int messageType)            Brings up an internal dialog panel where the number of choices is determined by the optionType parameter, where the messageType parameter determines the icon to display. |
| static int | [**showInternalConfirmDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showInternalConfirmDialog(java.awt.Component,%20java.lang.Object,%20java.lang.String,%20int,%20int,%20javax.swing.Icon))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message, [String](http://docs.google.com/java/lang/String.html) title, int optionType, int messageType, [Icon](http://docs.google.com/javax/swing/Icon.html) icon)            Brings up an internal dialog panel with a specified icon, where the number of choices is determined by the optionType parameter. |
| static [String](http://docs.google.com/java/lang/String.html) | [**showInternalInputDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showInternalInputDialog(java.awt.Component,%20java.lang.Object))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message)            Shows an internal question-message dialog requesting input from the user parented to parentComponent. |
| static [String](http://docs.google.com/java/lang/String.html) | [**showInternalInputDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showInternalInputDialog(java.awt.Component,%20java.lang.Object,%20java.lang.String,%20int))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message, [String](http://docs.google.com/java/lang/String.html) title, int messageType)            Shows an internal dialog requesting input from the user parented to parentComponent with the dialog having the title title and message type messageType. |
| static [Object](http://docs.google.com/java/lang/Object.html) | [**showInternalInputDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showInternalInputDialog(java.awt.Component,%20java.lang.Object,%20java.lang.String,%20int,%20javax.swing.Icon,%20java.lang.Object%5B%5D,%20java.lang.Object))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message, [String](http://docs.google.com/java/lang/String.html) title, int messageType, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, [Object](http://docs.google.com/java/lang/Object.html)[] selectionValues, [Object](http://docs.google.com/java/lang/Object.html) initialSelectionValue)            Prompts the user for input in a blocking internal dialog where the initial selection, possible selections, and all other options can be specified. |
| static void | [**showInternalMessageDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showInternalMessageDialog(java.awt.Component,%20java.lang.Object))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message)            Brings up an internal confirmation dialog panel. |
| static void | [**showInternalMessageDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showInternalMessageDialog(java.awt.Component,%20java.lang.Object,%20java.lang.String,%20int))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message, [String](http://docs.google.com/java/lang/String.html) title, int messageType)            Brings up an internal dialog panel that displays a message using a default icon determined by the messageType parameter. |
| static void | [**showInternalMessageDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showInternalMessageDialog(java.awt.Component,%20java.lang.Object,%20java.lang.String,%20int,%20javax.swing.Icon))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message, [String](http://docs.google.com/java/lang/String.html) title, int messageType, [Icon](http://docs.google.com/javax/swing/Icon.html) icon)            Brings up an internal dialog panel displaying a message, specifying all parameters. |
| static int | [**showInternalOptionDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showInternalOptionDialog(java.awt.Component,%20java.lang.Object,%20java.lang.String,%20int,%20int,%20javax.swing.Icon,%20java.lang.Object%5B%5D,%20java.lang.Object))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message, [String](http://docs.google.com/java/lang/String.html) title, int optionType, int messageType, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, [Object](http://docs.google.com/java/lang/Object.html)[] options, [Object](http://docs.google.com/java/lang/Object.html) initialValue)            Brings up an internal dialog panel with a specified icon, where the initial choice is determined by the initialValue parameter and the number of choices is determined by the optionType parameter. |
| static void | [**showMessageDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showMessageDialog(java.awt.Component,%20java.lang.Object))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message)            Brings up an information-message dialog titled "Message". |
| static void | [**showMessageDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showMessageDialog(java.awt.Component,%20java.lang.Object,%20java.lang.String,%20int))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message, [String](http://docs.google.com/java/lang/String.html) title, int messageType)            Brings up a dialog that displays a message using a default icon determined by the messageType parameter. |
| static void | [**showMessageDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showMessageDialog(java.awt.Component,%20java.lang.Object,%20java.lang.String,%20int,%20javax.swing.Icon))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message, [String](http://docs.google.com/java/lang/String.html) title, int messageType, [Icon](http://docs.google.com/javax/swing/Icon.html) icon)            Brings up a dialog displaying a message, specifying all parameters. |
| static int | [**showOptionDialog**](http://docs.google.com/javax/swing/JOptionPane.html#showOptionDialog(java.awt.Component,%20java.lang.Object,%20java.lang.String,%20int,%20int,%20javax.swing.Icon,%20java.lang.Object%5B%5D,%20java.lang.Object))([Component](http://docs.google.com/java/awt/Component.html) parentComponent, [Object](http://docs.google.com/java/lang/Object.html) message, [String](http://docs.google.com/java/lang/String.html) title, int optionType, int messageType, [Icon](http://docs.google.com/javax/swing/Icon.html) icon, [Object](http://docs.google.com/java/lang/Object.html)[] options, [Object](http://docs.google.com/java/lang/Object.html) initialValue)            Brings up a dialog with a specified icon, where the initial choice is determined by the initialValue parameter and the number of choices is determined by the optionType parameter. |
| void | [**updateUI**](http://docs.google.com/javax/swing/JOptionPane.html#updateUI())()            Notification from the UIManager that the L&F 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()), [getRootPane](http://docs.google.com/javax/swing/JComponent.html#getRootPane()), [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)), [paintComponent](http://docs.google.com/javax/swing/JComponent.html#paintComponent(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()), [reshape](http://docs.google.com/javax/swing/JComponent.html#reshape(int,%20int,%20int,%20int)), [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)), [addImpl](http://docs.google.com/java/awt/Container.html#addImpl(java.awt.Component,%20java.lang.Object,%20int)), [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()), [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(java.awt.Component)), [remove](http://docs.google.com/java/awt/Container.html#remove(int)), [removeAll](http://docs.google.com/java/awt/Container.html#removeAll()), [removeContainerListener](http://docs.google.com/java/awt/Container.html#removeContainerListener(java.awt.event.ContainerListener)), [setComponentZOrder](http://docs.google.com/java/awt/Container.html#setComponentZOrder(java.awt.Component,%20int)), [setFocusCycleRoot](http://docs.google.com/java/awt/Container.html#setFocusCycleRoot(boolean)), [setFocusTraversalPolicy](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicy(java.awt.FocusTraversalPolicy)), [setFocusTraversalPolicyProvider](http://docs.google.com/java/awt/Container.html#setFocusTraversalPolicyProvider(boolean)), [setLayout](http://docs.google.com/java/awt/Container.html#setLayout(java.awt.LayoutManager)), [transferFocusBackward](http://docs.google.com/java/awt/Container.html#transferFocusBackward()), [transferFocusDownCycle](http://docs.google.com/java/awt/Container.html#transferFocusDownCycle()), [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()), [getFocusCycleRootAncestor](http://docs.google.com/java/awt/Component.html#getFocusCycleRootAncestor()), [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()), [hide](http://docs.google.com/java/awt/Component.html#hide()), [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)), [setCursor](http://docs.google.com/java/awt/Component.html#setCursor(java.awt.Cursor)), [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()), [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** |
| --- |

### UNINITIALIZED\_VALUE

public static final [Object](http://docs.google.com/java/lang/Object.html) **UNINITIALIZED\_VALUE**

Indicates that the user has not yet selected a value.

### DEFAULT\_OPTION

public static final int **DEFAULT\_OPTION**

Type meaning Look and Feel should not supply any options -- only use the options from the JOptionPane.

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

### YES\_NO\_OPTION

public static final int **YES\_NO\_OPTION**

Type used for showConfirmDialog.

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

### YES\_NO\_CANCEL\_OPTION

public static final int **YES\_NO\_CANCEL\_OPTION**

Type used for showConfirmDialog.

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

### OK\_CANCEL\_OPTION

public static final int **OK\_CANCEL\_OPTION**

Type used for showConfirmDialog.

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

### YES\_OPTION

public static final int **YES\_OPTION**

Return value from class method if YES is chosen.

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

### NO\_OPTION

public static final int **NO\_OPTION**

Return value from class method if NO is chosen.

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

### CANCEL\_OPTION

public static final int **CANCEL\_OPTION**

Return value from class method if CANCEL is chosen.

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

### OK\_OPTION

public static final int **OK\_OPTION**

Return value form class method if OK is chosen.

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

### CLOSED\_OPTION

public static final int **CLOSED\_OPTION**

Return value from class method if user closes window without selecting anything, more than likely this should be treated as either a CANCEL\_OPTION or NO\_OPTION.

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

### ERROR\_MESSAGE

public static final int **ERROR\_MESSAGE**

Used for error messages.

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

### INFORMATION\_MESSAGE

public static final int **INFORMATION\_MESSAGE**

Used for information messages.

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

### WARNING\_MESSAGE

public static final int **WARNING\_MESSAGE**

Used for warning messages.

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

### QUESTION\_MESSAGE

public static final int **QUESTION\_MESSAGE**

Used for questions.

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

### PLAIN\_MESSAGE

public static final int **PLAIN\_MESSAGE**

No icon is used.

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

### ICON\_PROPERTY

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

Bound property name for icon.

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

### MESSAGE\_PROPERTY

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

Bound property name for message.

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

### VALUE\_PROPERTY

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

Bound property name for value.

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

### OPTIONS\_PROPERTY

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

Bound property name for option.

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

### INITIAL\_VALUE\_PROPERTY

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

Bound property name for initialValue.

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

### MESSAGE\_TYPE\_PROPERTY

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

Bound property name for type.

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

### OPTION\_TYPE\_PROPERTY

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

Bound property name for optionType.

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

### SELECTION\_VALUES\_PROPERTY

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

Bound property name for selectionValues.

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

### INITIAL\_SELECTION\_VALUE\_PROPERTY

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

Bound property name for initialSelectionValue.

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

### INPUT\_VALUE\_PROPERTY

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

Bound property name for inputValue.

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

### WANTS\_INPUT\_PROPERTY

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

Bound property name for wantsInput.

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

### icon

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

Icon used in pane.

### message

protected transient [Object](http://docs.google.com/java/lang/Object.html) **message**

Message to display.

### options

protected transient [Object](http://docs.google.com/java/lang/Object.html)[] **options**

Options to display to the user.

### initialValue

protected transient [Object](http://docs.google.com/java/lang/Object.html) **initialValue**

Value that should be initially selected in options.

### messageType

protected int **messageType**

Message type.

### optionType

protected int **optionType**

Option type, one of DEFAULT\_OPTION, YES\_NO\_OPTION, YES\_NO\_CANCEL\_OPTION or OK\_CANCEL\_OPTION.

### value

protected transient [Object](http://docs.google.com/java/lang/Object.html) **value**

Currently selected value, will be a valid option, or UNINITIALIZED\_VALUE or null.

### selectionValues

protected transient [Object](http://docs.google.com/java/lang/Object.html)[] **selectionValues**

Array of values the user can choose from. Look and feel will provide the UI component to choose this from.

### inputValue

protected transient [Object](http://docs.google.com/java/lang/Object.html) **inputValue**

Value the user has input.

### initialSelectionValue

protected transient [Object](http://docs.google.com/java/lang/Object.html) **initialSelectionValue**

Initial value to select in selectionValues.

### wantsInput

protected boolean **wantsInput**

If true, a UI widget will be provided to the user to get input.

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

### JOptionPane

public **JOptionPane**()

Creates a JOptionPane with a test message.

### JOptionPane

public **JOptionPane**([Object](http://docs.google.com/java/lang/Object.html) message)

Creates a instance of JOptionPane to display a message using the plain-message message type and the default options delivered by the UI.

**Parameters:**message - the Object to display

### JOptionPane

public **JOptionPane**([Object](http://docs.google.com/java/lang/Object.html) message,  
 int messageType)

Creates an instance of JOptionPane to display a message with the specified message type and the default options,

**Parameters:**message - the Object to displaymessageType - the type of message to be displayed: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGE

### JOptionPane

public **JOptionPane**([Object](http://docs.google.com/java/lang/Object.html) message,  
 int messageType,  
 int optionType)

Creates an instance of JOptionPane to display a message with the specified message type and options.

**Parameters:**message - the Object to displaymessageType - the type of message to be displayed: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGEoptionType - the options to display in the pane: DEFAULT\_OPTION, YES\_NO\_OPTION, YES\_NO\_CANCEL\_OPTION, OK\_CANCEL\_OPTION

### JOptionPane

public **JOptionPane**([Object](http://docs.google.com/java/lang/Object.html) message,  
 int messageType,  
 int optionType,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon)

Creates an instance of JOptionPane to display a message with the specified message type, options, and icon.

**Parameters:**message - the Object to displaymessageType - the type of message to be displayed: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGEoptionType - the options to display in the pane: DEFAULT\_OPTION, YES\_NO\_OPTION, YES\_NO\_CANCEL\_OPTION, OK\_CANCEL\_OPTIONicon - the Icon image to display

### JOptionPane

public **JOptionPane**([Object](http://docs.google.com/java/lang/Object.html) message,  
 int messageType,  
 int optionType,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 [Object](http://docs.google.com/java/lang/Object.html)[] options)

Creates an instance of JOptionPane to display a message with the specified message type, icon, and options. None of the options is initially selected.

The options objects should contain either instances of Components, (which are added directly) or Strings (which are wrapped in a JButton). If you provide Components, you must ensure that when the Component is clicked it messages setValue in the created JOptionPane.

**Parameters:**message - the Object to displaymessageType - the type of message to be displayed: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGEoptionType - the options to display in the pane: DEFAULT\_OPTION, YES\_NO\_OPTION, YES\_NO\_CANCEL\_OPTION, OK\_CANCEL\_OPTIONicon - the Icon image to displayoptions - the choices the user can select

### JOptionPane

public **JOptionPane**([Object](http://docs.google.com/java/lang/Object.html) message,  
 int messageType,  
 int optionType,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 [Object](http://docs.google.com/java/lang/Object.html)[] options,  
 [Object](http://docs.google.com/java/lang/Object.html) initialValue)

Creates an instance of JOptionPane to display a message with the specified message type, icon, and options, with the initially-selected option specified.

**Parameters:**message - the Object to displaymessageType - the type of message to be displayed: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGEoptionType - the options to display in the pane: DEFAULT\_OPTION, YES\_NO\_OPTION, YES\_NO\_CANCEL\_OPTION, OK\_CANCEL\_OPTIONicon - the Icon image to displayoptions - the choices the user can selectinitialValue - the choice that is initially selected; if null, then nothing will be initially selected; only meaningful if options is used

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

### showInputDialog

public static [String](http://docs.google.com/java/lang/String.html) **showInputDialog**([Object](http://docs.google.com/java/lang/Object.html) message)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Shows a question-message dialog requesting input from the user. The dialog uses the default frame, which usually means it is centered on the screen.

**Parameters:**message - the Object to display **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### showInputDialog

public static [String](http://docs.google.com/java/lang/String.html) **showInputDialog**([Object](http://docs.google.com/java/lang/Object.html) message,  
 [Object](http://docs.google.com/java/lang/Object.html) initialSelectionValue)

Shows a question-message dialog requesting input from the user, with the input value initialized to initialSelectionValue. The dialog uses the default frame, which usually means it is centered on the screen.

**Parameters:**message - the Object to displayinitialSelectionValue - the value used to initialize the input field**Since:** 1.4

### showInputDialog

public static [String](http://docs.google.com/java/lang/String.html) **showInputDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Shows a question-message dialog requesting input from the user parented to parentComponent. The dialog is displayed on top of the Component's frame, and is usually positioned below the Component.

**Parameters:**parentComponent - the parent Component for the dialogmessage - the Object to display **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### showInputDialog

public static [String](http://docs.google.com/java/lang/String.html) **showInputDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message,  
 [Object](http://docs.google.com/java/lang/Object.html) initialSelectionValue)

Shows a question-message dialog requesting input from the user and parented to parentComponent. The input value will be initialized to initialSelectionValue. The dialog is displayed on top of the Component's frame, and is usually positioned below the Component.

**Parameters:**parentComponent - the parent Component for the dialogmessage - the Object to displayinitialSelectionValue - the value used to initialize the input field**Since:** 1.4

### showInputDialog

public static [String](http://docs.google.com/java/lang/String.html) **showInputDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 int messageType)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Shows a dialog requesting input from the user parented to parentComponent with the dialog having the title title and message type messageType.

**Parameters:**parentComponent - the parent Component for the dialogmessage - the Object to displaytitle - the String to display in the dialog title barmessageType - the type of message that is to be displayed: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGE **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### showInputDialog

public static [Object](http://docs.google.com/java/lang/Object.html) **showInputDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 int messageType,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 [Object](http://docs.google.com/java/lang/Object.html)[] selectionValues,  
 [Object](http://docs.google.com/java/lang/Object.html) initialSelectionValue)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Prompts the user for input in a blocking dialog where the initial selection, possible selections, and all other options can be specified. The user will able to choose from selectionValues, where null implies the user can input whatever they wish, usually by means of a JTextField. initialSelectionValue is the initial value to prompt the user with. It is up to the UI to decide how best to represent the selectionValues, but usually a JComboBox, JList, or JTextField will be used.

**Parameters:**parentComponent - the parent Component for the dialogmessage - the Object to displaytitle - the String to display in the dialog title barmessageType - the type of message to be displayed: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGEicon - the Icon image to displayselectionValues - an array of Objects that gives the possible selectionsinitialSelectionValue - the value used to initialize the input field **Returns:**user's input, or null meaning the user canceled the input **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### showMessageDialog

public static void **showMessageDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Brings up an information-message dialog titled "Message".

**Parameters:**parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is usedmessage - the Object to display **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### showMessageDialog

public static void **showMessageDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 int messageType)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Brings up a dialog that displays a message using a default icon determined by the messageType parameter.

**Parameters:**parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is usedmessage - the Object to displaytitle - the title string for the dialogmessageType - the type of message to be displayed: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGE **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### showMessageDialog

public static void **showMessageDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 int messageType,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Brings up a dialog displaying a message, specifying all parameters.

**Parameters:**parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is usedmessage - the Object to displaytitle - the title string for the dialogmessageType - the type of message to be displayed: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGEicon - an icon to display in the dialog that helps the user identify the kind of message that is being displayed **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### showConfirmDialog

public static int **showConfirmDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Brings up a dialog with the options *Yes*, *No* and *Cancel*; with the title, **Select an Option**.

**Parameters:**parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is usedmessage - the Object to display **Returns:**an integer indicating the option selected by the user **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### showConfirmDialog

public static int **showConfirmDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 int optionType)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Brings up a dialog where the number of choices is determined by the optionType parameter.

**Parameters:**parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is usedmessage - the Object to displaytitle - the title string for the dialogoptionType - an int designating the options available on the dialog: YES\_NO\_OPTION, YES\_NO\_CANCEL\_OPTION, or OK\_CANCEL\_OPTION **Returns:**an int indicating the option selected by the user **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### showConfirmDialog

public static int **showConfirmDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 int optionType,  
 int messageType)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Brings up a dialog where the number of choices is determined by the optionType parameter, where the messageType parameter determines the icon to display. The messageType parameter is primarily used to supply a default icon from the Look and Feel.

**Parameters:**parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is used.message - the Object to displaytitle - the title string for the dialogoptionType - an integer designating the options available on the dialog: YES\_NO\_OPTION, YES\_NO\_CANCEL\_OPTION, or OK\_CANCEL\_OPTIONmessageType - an integer designating the kind of message this is; primarily used to determine the icon from the pluggable Look and Feel: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGE **Returns:**an integer indicating the option selected by the user **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### showConfirmDialog

public static int **showConfirmDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 int optionType,  
 int messageType,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Brings up a dialog with a specified icon, where the number of choices is determined by the optionType parameter. The messageType parameter is primarily used to supply a default icon from the look and feel.

**Parameters:**parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is usedmessage - the Object to displaytitle - the title string for the dialogoptionType - an int designating the options available on the dialog: YES\_NO\_OPTION, YES\_NO\_CANCEL\_OPTION, or OK\_CANCEL\_OPTIONmessageType - an int designating the kind of message this is, primarily used to determine the icon from the pluggable Look and Feel: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGEicon - the icon to display in the dialog **Returns:**an int indicating the option selected by the user **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### showOptionDialog

public static int **showOptionDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 int optionType,  
 int messageType,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 [Object](http://docs.google.com/java/lang/Object.html)[] options,  
 [Object](http://docs.google.com/java/lang/Object.html) initialValue)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Brings up a dialog with a specified icon, where the initial choice is determined by the initialValue parameter and the number of choices is determined by the optionType parameter.

If optionType is YES\_NO\_OPTION, or YES\_NO\_CANCEL\_OPTION and the options parameter is null, then the options are supplied by the look and feel.

The messageType parameter is primarily used to supply a default icon from the look and feel.

**Parameters:**parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is usedmessage - the Object to displaytitle - the title string for the dialogoptionType - an integer designating the options available on the dialog: DEFAULT\_OPTION, YES\_NO\_OPTION, YES\_NO\_CANCEL\_OPTION, or OK\_CANCEL\_OPTIONmessageType - an integer designating the kind of message this is, primarily used to determine the icon from the pluggable Look and Feel: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGEicon - the icon to display in the dialogoptions - an array of objects indicating the possible choices the user can make; if the objects are components, they are rendered properly; non-String objects are rendered using their toString methods; if this parameter is null, the options are determined by the Look and FeelinitialValue - the object that represents the default selection for the dialog; only meaningful if options is used; can be null **Returns:**an integer indicating the option chosen by the user, or CLOSED\_OPTION if the user closed the dialog **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### createDialog

public [JDialog](http://docs.google.com/javax/swing/JDialog.html) **createDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [String](http://docs.google.com/java/lang/String.html) title)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Creates and returns a new JDialog wrapping this centered on the parentComponent in the parentComponent's frame. title is the title of the returned dialog. The returned JDialog will not be resizable by the user, however programs can invoke setResizable on the JDialog instance to change this property. The returned JDialog will be set up such that once it is closed, or the user clicks on one of the buttons, the optionpane's value property will be set accordingly and the dialog will be closed. Each time the dialog is made visible, it will reset the option pane's value property to JOptionPane.UNINITIALIZED\_VALUE to ensure the user's subsequent action closes the dialog properly.

**Parameters:**parentComponent - determines the frame in which the dialog is displayed; if the parentComponent has no Frame, a default Frame is usedtitle - the title string for the dialog **Returns:**a new JDialog containing this instance **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### createDialog

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

Creates and returns a new parentless JDialog with the specified title. The returned JDialog will not be resizable by the user, however programs can invoke setResizable on the JDialog instance to change this property. The returned JDialog will be set up such that once it is closed, or the user clicks on one of the buttons, the optionpane's value property will be set accordingly and the dialog will be closed. Each time the dialog is made visible, it will reset the option pane's value property to JOptionPane.UNINITIALIZED\_VALUE to ensure the user's subsequent action closes the dialog properly.

**Parameters:**title - the title string for the dialog **Returns:**a new JDialog containing this instance **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**Since:** 1.6 **See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### showInternalMessageDialog

public static void **showInternalMessageDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message)

Brings up an internal confirmation dialog panel. The dialog is a information-message dialog titled "Message".

**Parameters:**parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is usedmessage - the object to display

### showInternalMessageDialog

public static void **showInternalMessageDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 int messageType)

Brings up an internal dialog panel that displays a message using a default icon determined by the messageType parameter.

**Parameters:**parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is usedmessage - the Object to displaytitle - the title string for the dialogmessageType - the type of message to be displayed: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGE

### showInternalMessageDialog

public static void **showInternalMessageDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 int messageType,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon)

Brings up an internal dialog panel displaying a message, specifying all parameters.

**Parameters:**parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is usedmessage - the Object to displaytitle - the title string for the dialogmessageType - the type of message to be displayed: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGEicon - an icon to display in the dialog that helps the user identify the kind of message that is being displayed

### showInternalConfirmDialog

public static int **showInternalConfirmDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message)

Brings up an internal dialog panel with the options *Yes*, *No* and *Cancel*; with the title, **Select an Option**.

**Parameters:**parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is usedmessage - the Object to display **Returns:**an integer indicating the option selected by the user

### showInternalConfirmDialog

public static int **showInternalConfirmDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 int optionType)

Brings up a internal dialog panel where the number of choices is determined by the optionType parameter.

**Parameters:**parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is usedmessage - the object to display in the dialog; a Component object is rendered as a Component; a String object is rendered as a string; other objects are converted to a String using the toString methodtitle - the title string for the dialogoptionType - an integer designating the options available on the dialog: YES\_NO\_OPTION, or YES\_NO\_CANCEL\_OPTION **Returns:**an integer indicating the option selected by the user

### showInternalConfirmDialog

public static int **showInternalConfirmDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 int optionType,  
 int messageType)

Brings up an internal dialog panel where the number of choices is determined by the optionType parameter, where the messageType parameter determines the icon to display. The messageType parameter is primarily used to supply a default icon from the Look and Feel.

**Parameters:**parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is usedmessage - the object to display in the dialog; a Component object is rendered as a Component; a String object is rendered as a string; other objects are converted to a String using the toString methodtitle - the title string for the dialogoptionType - an integer designating the options available on the dialog: YES\_NO\_OPTION, or YES\_NO\_CANCEL\_OPTIONmessageType - an integer designating the kind of message this is, primarily used to determine the icon from the pluggable Look and Feel: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGE **Returns:**an integer indicating the option selected by the user

### showInternalConfirmDialog

public static int **showInternalConfirmDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 int optionType,  
 int messageType,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon)

Brings up an internal dialog panel with a specified icon, where the number of choices is determined by the optionType parameter. The messageType parameter is primarily used to supply a default icon from the look and feel.

**Parameters:**parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is usedmessage - the object to display in the dialog; a Component object is rendered as a Component; a String object is rendered as a string; other objects are converted to a String using the toString methodtitle - the title string for the dialogoptionType - an integer designating the options available on the dialog: YES\_NO\_OPTION, or YES\_NO\_CANCEL\_OPTION.messageType - an integer designating the kind of message this is, primarily used to determine the icon from the pluggable Look and Feel: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGEicon - the icon to display in the dialog **Returns:**an integer indicating the option selected by the user

### showInternalOptionDialog

public static int **showInternalOptionDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 int optionType,  
 int messageType,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 [Object](http://docs.google.com/java/lang/Object.html)[] options,  
 [Object](http://docs.google.com/java/lang/Object.html) initialValue)

Brings up an internal dialog panel with a specified icon, where the initial choice is determined by the initialValue parameter and the number of choices is determined by the optionType parameter.

If optionType is YES\_NO\_OPTION, or YES\_NO\_CANCEL\_OPTION and the options parameter is null, then the options are supplied by the Look and Feel.

The messageType parameter is primarily used to supply a default icon from the look and feel.

**Parameters:**parentComponent - determines the Frame in which the dialog is displayed; if null, or if the parentComponent has no Frame, a default Frame is usedmessage - the object to display in the dialog; a Component object is rendered as a Component; a String object is rendered as a string. Other objects are converted to a String using the toString methodtitle - the title string for the dialogoptionType - an integer designating the options available on the dialog: YES\_NO\_OPTION, or YES\_NO\_CANCEL\_OPTIONmessageType - an integer designating the kind of message this is; primarily used to determine the icon from the pluggable Look and Feel: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGEicon - the icon to display in the dialogoptions - an array of objects indicating the possible choices the user can make; if the objects are components, they are rendered properly; non-String objects are rendered using their toString methods; if this parameter is null, the options are determined by the Look and FeelinitialValue - the object that represents the default selection for the dialog; only meaningful if options is used; can be null **Returns:**an integer indicating the option chosen by the user, or CLOSED\_OPTION if the user closed the Dialog

### showInternalInputDialog

public static [String](http://docs.google.com/java/lang/String.html) **showInternalInputDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message)

Shows an internal question-message dialog requesting input from the user parented to parentComponent. The dialog is displayed in the Component's frame, and is usually positioned below the Component.

**Parameters:**parentComponent - the parent Component for the dialogmessage - the Object to display

### showInternalInputDialog

public static [String](http://docs.google.com/java/lang/String.html) **showInternalInputDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 int messageType)

Shows an internal dialog requesting input from the user parented to parentComponent with the dialog having the title title and message type messageType.

**Parameters:**parentComponent - the parent Component for the dialogmessage - the Object to displaytitle - the String to display in the dialog title barmessageType - the type of message that is to be displayed: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGE

### showInternalInputDialog

public static [Object](http://docs.google.com/java/lang/Object.html) **showInternalInputDialog**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [Object](http://docs.google.com/java/lang/Object.html) message,  
 [String](http://docs.google.com/java/lang/String.html) title,  
 int messageType,  
 [Icon](http://docs.google.com/javax/swing/Icon.html) icon,  
 [Object](http://docs.google.com/java/lang/Object.html)[] selectionValues,  
 [Object](http://docs.google.com/java/lang/Object.html) initialSelectionValue)

Prompts the user for input in a blocking internal dialog where the initial selection, possible selections, and all other options can be specified. The user will able to choose from selectionValues, where null implies the user can input whatever they wish, usually by means of a JTextField. initialSelectionValue is the initial value to prompt the user with. It is up to the UI to decide how best to represent the selectionValues, but usually a JComboBox, JList, or JTextField will be used.

**Parameters:**parentComponent - the parent Component for the dialogmessage - the Object to displaytitle - the String to display in the dialog title barmessageType - the type of message to be displayed: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGEicon - the Icon image to displayselectionValues - an array of Objects that gives the possible selectionsinitialSelectionValue - the value used to initialize the input field **Returns:**user's input, or null meaning the user canceled the input

### createInternalFrame

public [JInternalFrame](http://docs.google.com/javax/swing/JInternalFrame.html) **createInternalFrame**([Component](http://docs.google.com/java/awt/Component.html) parentComponent,  
 [String](http://docs.google.com/java/lang/String.html) title)

Creates and returns an instance of JInternalFrame. The internal frame is created with the specified title, and wrapping the JOptionPane. The returned JInternalFrame is added to the JDesktopPane ancestor of parentComponent, or components parent if one its ancestors isn't a JDesktopPane, or if parentComponent doesn't have a parent then a RuntimeException is thrown.

**Parameters:**parentComponent - the parent Component for the internal frametitle - the String to display in the frame's title bar **Returns:**a JInternalFrame containing a JOptionPane **Throws:** [RuntimeException](http://docs.google.com/java/lang/RuntimeException.html) - if parentComponent does not have a valid parent

### getFrameForComponent

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

Returns the specified component's Frame.

**Parameters:**parentComponent - the Component to check for a Frame **Returns:**the Frame that contains the component, or getRootFrame if the component is null, or does not have a valid Frame parent **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[getRootFrame()](http://docs.google.com/javax/swing/JOptionPane.html#getRootFrame()), [GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### getDesktopPaneForComponent

public static [JDesktopPane](http://docs.google.com/javax/swing/JDesktopPane.html) **getDesktopPaneForComponent**([Component](http://docs.google.com/java/awt/Component.html) parentComponent)

Returns the specified component's desktop pane.

**Parameters:**parentComponent - the Component to check for a desktop **Returns:**the JDesktopPane that contains the component, or null if the component is null or does not have an ancestor that is a JInternalFrame

### setRootFrame

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

Sets the frame to use for class methods in which a frame is not provided.

**Note:** It is recommended that rather than using this method you supply a valid parent.

**Parameters:**newRootFrame - the default Frame to use

### getRootFrame

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

Returns the Frame to use for the class methods in which a frame is not provided.

**Returns:**the default Frame to use **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless returns true**See Also:**[setRootFrame(java.awt.Frame)](http://docs.google.com/javax/swing/JOptionPane.html#setRootFrame(java.awt.Frame)), [GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### setUI

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

Sets the UI object which implements the L&F for this component.

**Parameters:**ui - the OptionPaneUI L&F object**See Also:**[UIDefaults.getUI(javax.swing.JComponent)](http://docs.google.com/javax/swing/UIDefaults.html#getUI(javax.swing.JComponent))

### getUI

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

Returns the UI object which implements the L&F for this component.

**Returns:**the OptionPaneUI object

### updateUI

public void **updateUI**()

Notification from the UIManager that the L&F 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 UI class that implements the L&F for 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 "OptionPaneUI"**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))

### setMessage

public void **setMessage**([Object](http://docs.google.com/java/lang/Object.html) newMessage)

Sets the option pane's message-object.

**Parameters:**newMessage - the Object to display**See Also:**[getMessage()](http://docs.google.com/javax/swing/JOptionPane.html#getMessage())

### getMessage

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

Returns the message-object this pane displays.

**Returns:**the Object that is displayed**See Also:**[setMessage(java.lang.Object)](http://docs.google.com/javax/swing/JOptionPane.html#setMessage(java.lang.Object))

### setIcon

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

Sets the icon to display. If non-null, the look and feel does not provide an icon.

**Parameters:**newIcon - the Icon to display**See Also:**[getIcon()](http://docs.google.com/javax/swing/JOptionPane.html#getIcon())

### getIcon

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

Returns the icon this pane displays.

**Returns:**the Icon that is displayed**See Also:**[setIcon(javax.swing.Icon)](http://docs.google.com/javax/swing/JOptionPane.html#setIcon(javax.swing.Icon))

### setValue

public void **setValue**([Object](http://docs.google.com/java/lang/Object.html) newValue)

Sets the value the user has chosen.

**Parameters:**newValue - the chosen value**See Also:**[getValue()](http://docs.google.com/javax/swing/JOptionPane.html#getValue())

### getValue

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

Returns the value the user has selected. UNINITIALIZED\_VALUE implies the user has not yet made a choice, null means the user closed the window with out choosing anything. Otherwise the returned value will be one of the options defined in this object.

**Returns:**the Object chosen by the user, UNINITIALIZED\_VALUE if the user has not yet made a choice, or null if the user closed the window without making a choice**See Also:**[setValue(java.lang.Object)](http://docs.google.com/javax/swing/JOptionPane.html#setValue(java.lang.Object))

### setOptions

public void **setOptions**([Object](http://docs.google.com/java/lang/Object.html)[] newOptions)

Sets the options this pane displays. If an element in newOptions is a Component it is added directly to the pane, otherwise a button is created for the element.

**Parameters:**newOptions - an array of Objects that create the buttons the user can click on, or arbitrary Components to add to the pane**See Also:**[getOptions()](http://docs.google.com/javax/swing/JOptionPane.html#getOptions())

### getOptions

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

Returns the choices the user can make.

**Returns:**the array of Objects that give the user's choices**See Also:**[setOptions(java.lang.Object[])](http://docs.google.com/javax/swing/JOptionPane.html#setOptions(java.lang.Object%5B%5D))

### setInitialValue

public void **setInitialValue**([Object](http://docs.google.com/java/lang/Object.html) newInitialValue)

Sets the initial value that is to be enabled -- the Component that has the focus when the pane is initially displayed.

**Parameters:**newInitialValue - the Object that gets the initial keyboard focus**See Also:**[getInitialValue()](http://docs.google.com/javax/swing/JOptionPane.html#getInitialValue())

### getInitialValue

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

Returns the initial value.

**Returns:**the Object that gets the initial keyboard focus**See Also:**[setInitialValue(java.lang.Object)](http://docs.google.com/javax/swing/JOptionPane.html#setInitialValue(java.lang.Object))

### setMessageType

public void **setMessageType**(int newType)

Sets the option pane's message type. The message type is used by the Look and Feel to determine the icon to display (if not supplied) as well as potentially how to lay out the parentComponent.

**Parameters:**newType - an integer specifying the kind of message to display: ERROR\_MESSAGE, INFORMATION\_MESSAGE, WARNING\_MESSAGE, QUESTION\_MESSAGE, or PLAIN\_MESSAGE **Throws:** [RuntimeException](http://docs.google.com/java/lang/RuntimeException.html) - if newType is not one of the legal values listed above**See Also:**[getMessageType()](http://docs.google.com/javax/swing/JOptionPane.html#getMessageType())

### getMessageType

public int **getMessageType**()

Returns the message type.

**Returns:**an integer specifying the message type**See Also:**[setMessageType(int)](http://docs.google.com/javax/swing/JOptionPane.html#setMessageType(int))

### setOptionType

public void **setOptionType**(int newType)

Sets the options to display. The option type is used by the Look and Feel to determine what buttons to show (unless options are supplied).

**Parameters:**newType - an integer specifying the options the L&F is to display: DEFAULT\_OPTION, YES\_NO\_OPTION, YES\_NO\_CANCEL\_OPTION, or OK\_CANCEL\_OPTION **Throws:** [RuntimeException](http://docs.google.com/java/lang/RuntimeException.html) - if newType is not one of the legal values listed above**See Also:**[getOptionType()](http://docs.google.com/javax/swing/JOptionPane.html#getOptionType()), [setOptions(java.lang.Object[])](http://docs.google.com/javax/swing/JOptionPane.html#setOptions(java.lang.Object%5B%5D))

### getOptionType

public int **getOptionType**()

Returns the type of options that are displayed.

**Returns:**an integer specifying the user-selectable options**See Also:**[setOptionType(int)](http://docs.google.com/javax/swing/JOptionPane.html#setOptionType(int))

### setSelectionValues

public void **setSelectionValues**([Object](http://docs.google.com/java/lang/Object.html)[] newValues)

Sets the input selection values for a pane that provides the user with a list of items to choose from. (The UI provides a widget for choosing one of the values.) A null value implies the user can input whatever they wish, usually by means of a JTextField.

Sets wantsInput to true. Use setInitialSelectionValue to specify the initially-chosen value. After the pane as been enabled, inputValue is set to the value the user has selected.

**Parameters:**newValues - an array of Objects the user to be displayed (usually in a list or combo-box) from which the user can make a selection**See Also:**[setWantsInput(boolean)](http://docs.google.com/javax/swing/JOptionPane.html#setWantsInput(boolean)), [setInitialSelectionValue(java.lang.Object)](http://docs.google.com/javax/swing/JOptionPane.html#setInitialSelectionValue(java.lang.Object)), [getSelectionValues()](http://docs.google.com/javax/swing/JOptionPane.html#getSelectionValues())

### getSelectionValues

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

Returns the input selection values.

**Returns:**the array of Objects the user can select**See Also:**[setSelectionValues(java.lang.Object[])](http://docs.google.com/javax/swing/JOptionPane.html#setSelectionValues(java.lang.Object%5B%5D))

### setInitialSelectionValue

public void **setInitialSelectionValue**([Object](http://docs.google.com/java/lang/Object.html) newValue)

Sets the input value that is initially displayed as selected to the user. Only used if wantsInput is true.

**Parameters:**newValue - the initially selected value**See Also:**[setSelectionValues(java.lang.Object[])](http://docs.google.com/javax/swing/JOptionPane.html#setSelectionValues(java.lang.Object%5B%5D)), [getInitialSelectionValue()](http://docs.google.com/javax/swing/JOptionPane.html#getInitialSelectionValue())

### getInitialSelectionValue

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

Returns the input value that is displayed as initially selected to the user.

**Returns:**the initially selected value**See Also:**[setInitialSelectionValue(java.lang.Object)](http://docs.google.com/javax/swing/JOptionPane.html#setInitialSelectionValue(java.lang.Object)), [setSelectionValues(java.lang.Object[])](http://docs.google.com/javax/swing/JOptionPane.html#setSelectionValues(java.lang.Object%5B%5D))

### setInputValue

public void **setInputValue**([Object](http://docs.google.com/java/lang/Object.html) newValue)

Sets the input value that was selected or input by the user. Only used if wantsInput is true. Note that this method is invoked internally by the option pane (in response to user action) and should generally not be called by client programs. To set the input value initially displayed as selected to the user, use setInitialSelectionValue.

**Parameters:**newValue - the Object used to set the value that the user specified (usually in a text field)**See Also:**[setSelectionValues(java.lang.Object[])](http://docs.google.com/javax/swing/JOptionPane.html#setSelectionValues(java.lang.Object%5B%5D)), [setInitialSelectionValue(java.lang.Object)](http://docs.google.com/javax/swing/JOptionPane.html#setInitialSelectionValue(java.lang.Object)), [setWantsInput(boolean)](http://docs.google.com/javax/swing/JOptionPane.html#setWantsInput(boolean)), [getInputValue()](http://docs.google.com/javax/swing/JOptionPane.html#getInputValue())

### getInputValue

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

Returns the value the user has input, if wantsInput is true.

**Returns:**the Object the user specified, if it was one of the objects, or a String if it was a value typed into a field**See Also:**[setSelectionValues(java.lang.Object[])](http://docs.google.com/javax/swing/JOptionPane.html#setSelectionValues(java.lang.Object%5B%5D)), [setWantsInput(boolean)](http://docs.google.com/javax/swing/JOptionPane.html#setWantsInput(boolean)), [setInputValue(java.lang.Object)](http://docs.google.com/javax/swing/JOptionPane.html#setInputValue(java.lang.Object))

### getMaxCharactersPerLineCount

public int **getMaxCharactersPerLineCount**()

Returns the maximum number of characters to place on a line in a message. Default is to return Integer.MAX\_VALUE. The value can be changed by overriding this method in a subclass.

**Returns:**an integer giving the maximum number of characters on a line

### setWantsInput

public void **setWantsInput**(boolean newValue)

Sets the wantsInput property. If newValue is true, an input component (such as a text field or combo box) whose parent is parentComponent is provided to allow the user to input a value. If getSelectionValues returns a non-null array, the input value is one of the objects in that array. Otherwise the input value is whatever the user inputs.

This is a bound property.

**See Also:**[setSelectionValues(java.lang.Object[])](http://docs.google.com/javax/swing/JOptionPane.html#setSelectionValues(java.lang.Object%5B%5D)), [setInputValue(java.lang.Object)](http://docs.google.com/javax/swing/JOptionPane.html#setInputValue(java.lang.Object))

### getWantsInput

public boolean **getWantsInput**()

Returns the value of the wantsInput property.

**Returns:**true if an input component will be provided**See Also:**[setWantsInput(boolean)](http://docs.google.com/javax/swing/JOptionPane.html#setWantsInput(boolean))

### selectInitialValue

public void **selectInitialValue**()

Requests that the initial value be selected, which will set focus to the initial value. This method should be invoked after the window containing the option pane is made visible.

### paramString

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

Returns a string representation of this JOptionPane. 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 JOptionPane

### getAccessibleContext

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

Returns the AccessibleContext associated with this JOptionPane. For option panes, the AccessibleContext takes the form of an AccessibleJOptionPane. A new AccessibleJOptionPane 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 AccessibleJOptionPane that serves as the AccessibleContext of this AccessibleJOptionPane

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/JOptionPane.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/JMenuItem.AccessibleJMenuItem.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/JOptionPane.AccessibleJOptionPane.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/JOptionPane.html)    [**NO FRAMES**](http://docs.google.com/JOptionPane.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#2et92p0) | [FIELD](#4d34og8) | [CONSTR](#26in1rg) | [METHOD](#lnxbz9) | DETAIL: [FIELD](#z337ya) | [CONSTR](#4k668n3) | [METHOD](#4bvk7pj) |

[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).