

## Interface ActionListener

The listener interface for receiving action events. The object created with that class is registered with a component, using the component's **addActionListener** method.

Modifier and Type	Method and Description
void	<u><a>actionPerformed(ActionEvent e)</a></u> Invoked when an action occurs.

## Interface ComponentListener

The listener interface for receiving component events. The listener object created from that class is registered with a component using the component's **addComponentListener** method.

Modifier and Type	Method and Description
void	<u><a>componentHidden(ComponentEvent e)</a></u> Invoked when the component has been made invisible.
void	<u><a>componentMoved(ComponentEvent e)</a></u> Invoked when the component's position changes.
void	<u><a>componentResized(ComponentEvent e)</a></u> Invoked when the component's size changes.
void	<u><a>componentShown(ComponentEvent e)</a></u> Invoked when the component has been made visible.

## Interface ContainerListener

The listener interface for receiving container events. The listener object created from that class is registered with a component using the component's **addContainerListener** method.

Modifier and Type	Method and Description
void	<u><a>componentAdded(ContainerEvent e)</a></u> Invoked when a component has been added to the container.
void	<u><a>componentRemoved(ContainerEvent e)</a></u> Invoked when a component has been removed from the container.

## Interface FocusListener

The listener interface for receiving keyboard focus events on a component. The listener object created from that class is registered with a component using the component's **addFocusListener** method.

Modifier and Type	Method and Description
void	<u><a>focusGained(FocusEvent e)</a></u> Invoked when a component gains the keyboard focus.
void	<u><a>focusLost(FocusEvent e)</a></u> Invoked when a component loses the keyboard focus.

## Interface ItemListener

The listener interface for receiving item events. The object created with that class is registered with a component using the component's **addItemListener** method.

Modifier and Type	Method and Description
void	<u><a>itemStateChanged(ItemEvent e)</a></u> Invoked when an item has been selected or deselected by the user.

## Interface KeyListener

The listener interface for receiving keyboard events (keystrokes). The listener object created from that class is registered with a component using the component's **addKeyListener** method.

Modifier and Type	Method and Description
void	<u><a>keyPressed(KeyEvent e)</a></u> Invoked when a key has been pressed.
void	<u><a>keyReleased(KeyEvent e)</a></u> Invoked when a key has been released.
void	<u><a>keyTyped(KeyEvent e)</a></u> Invoked when a key has been typed.

## Interface MouseListener

The listener interface for receiving mouse events (press, release, click, enter, and exit) on a component. The listener object created from that class is registered with a component using the component's **addMouseListener** method.

Modifier and Type	Method and Description
void	<u><a>mouseClicked(MouseEvent e)</a></u> Invoked when the mouse button has been clicked (pressed and released) on a component.
void	<u><a>mouseEntered(MouseEvent e)</a></u> Invoked when the mouse enters a component.
void	<u><a>mouseExited(MouseEvent e)</a></u> Invoked when the mouse exits a component.

<b>void</b>	<b><u>mousePressed(MouseEvent e)</u></b> Invoked when a mouse button has been pressed on a component.
<b>void</b>	<b><u>mouseReleased(MouseEvent e)</u></b> Invoked when a mouse button has been released on a component.

## Interface MouseMotionListener

The listener interface for receiving mouse motion events on a component. The listener object created from that class is registered with a component using the component's **addMouseMotionListener** method.

Modifier and Type	Method and Description
<b>void</b>	<b><u>mouseDragged(MouseEvent e)</u></b> Invoked when a mouse button is pressed on a component and dragged.
<b>void</b>	<b><u>mouseMoved(MouseEvent e)</u></b> Invoked when the mouse cursor has been moved onto a component but no buttons have been pushed.

## Interface MouseWheelListener

The listener interface for receiving mouse wheel events on a component. The listener object created from that class is registered with a component using the component's **addMouseWheelListener** method.

Modifier and Type	Method and Description
<b>void</b>	<b><u>mouseWheelMoved(MouseWheelEvent e)</u></b> Invoked when the mouse wheel is rotated.

## Interface TextListener

The listener interface for receiving text events. The object created with that class is registered with a component using the component's **addTextListener** method.

Modifier and Type	Method and Description
<b>void</b>	<b><u>textValueChanged(TextEvent e)</u></b> Invoked when the value of the text has changed.

## Interface WindowFocusListener

The listener interface for receiving WindowEvents. The listener object created from that class is registered with a Window using the Window's **addWindowFocusListener** method.

Modifier and Type	Method and Description
<b>void</b>	<b><u>windowGainedFocus(WindowEvent e)</u></b> Invoked when the Window is set to be the focused Window, which means that the Window, or one of its subcomponents, will receive keyboard events.
<b>void</b>	<b><u>windowLostFocus(WindowEvent e)</u></b> Invoked when the Window is no longer the focused Window, which means that keyboard events will no longer be delivered to the Window or any of its subcomponents.

## Interface WindowListener

The listener interface for receiving window events. The listener object created from that class is registered with a Window using the window's **addWindowListener** method.

Modifier and Type	Method and Description
<b>void</b>	<b><u>windowActivated(WindowEvent e)</u></b> Invoked when the Window is set to be the active Window.
<b>void</b>	<b><u>windowClosed(WindowEvent e)</u></b> Invoked when a window has been closed as the result of calling dispose on the window.
<b>void</b>	<b><u>windowClosing(WindowEvent e)</u></b> Invoked when the user attempts to close the window from the window's system menu.
<b>void</b>	<b><u>windowDeactivated(WindowEvent e)</u></b> Invoked when a Window is no longer the active Window.
<b>void</b>	<b><u>windowDeiconified(WindowEvent e)</u></b> Invoked when a window is changed from a minimized to a normal state.
<b>void</b>	<b><u>windowIconified(WindowEvent e)</u></b> Invoked when a window is changed from a normal to a minimized state.
<b>void</b>	<b><u>windowOpened(WindowEvent e)</u></b> Invoked the first time a window is made visible.

## Interface WindowStateListener

The listener interface for receiving window state events. The listener object created from that class is registered with a window using the Window's **addWindowStateListener** method.

Modifier and Type	Method and Description
<b>void</b>	<b><u>windowStateChanged(WindowEvent e)</u></b> Invoked when window state is changed.