| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MouseEvent.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/event/MouseAdapter.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/event/MouseListener.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/event/MouseEvent.html)    [**NO FRAMES**](http://docs.google.com/MouseEvent.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#4d34og8) | [METHOD](#2s8eyo1) | DETAIL: [FIELD](#1ksv4uv) | [CONSTR](#2p2csry) | [METHOD](#ihv636) |

## **java.awt.event**

Class MouseEvent

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.util.EventObject](http://docs.google.com/java/util/EventObject.html)  
 [java.awt.AWTEvent](http://docs.google.com/java/awt/AWTEvent.html)  
 [java.awt.event.ComponentEvent](http://docs.google.com/java/awt/event/ComponentEvent.html)  
 [java.awt.event.InputEvent](http://docs.google.com/java/awt/event/InputEvent.html)  
 **java.awt.event.MouseEvent**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html) **Direct Known Subclasses:** [MenuDragMouseEvent](http://docs.google.com/javax/swing/event/MenuDragMouseEvent.html), [MouseWheelEvent](http://docs.google.com/java/awt/event/MouseWheelEvent.html)

public class **MouseEvent**extends [InputEvent](http://docs.google.com/java/awt/event/InputEvent.html)

An event which indicates that a mouse action occurred in a component. A mouse action is considered to occur in a particular component if and only if the mouse cursor is over the unobscured part of the component's bounds when the action happens. For lightweight components, such as Swing's components, mouse events are only dispatched to the component if the mouse event type has been enabled on the component. A mouse event type is enabled by adding the appropriate mouse-based EventListener to the component ([MouseListener](http://docs.google.com/java/awt/event/MouseListener.html) or [MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html)), or by invoking [Component.enableEvents(long)](http://docs.google.com/java/awt/Component.html#enableEvents(long)) with the appropriate mask parameter (AWTEvent.MOUSE\_EVENT\_MASK or AWTEvent.MOUSE\_MOTION\_EVENT\_MASK). If the mouse event type has not been enabled on the component, the corresponding mouse events are dispatched to the first ancestor that has enabled the mouse event type.

For example, if a MouseListener has been added to a component, or enableEvents(AWTEvent.MOUSE\_EVENT\_MASK) has been invoked, then all the events defined by MouseListener are dispatched to the component. On the other hand, if a MouseMotionListener has not been added and enableEvents has not been invoked with AWTEvent.MOUSE\_MOTION\_EVENT\_MASK, then mouse motion events are not dispatched to the component. Instead the mouse motion events are dispatched to the first ancestors that has enabled mouse motion events.

This low-level event is generated by a component object for:

* Mouse Events
  + a mouse button is pressed
  + a mouse button is released
  + a mouse button is clicked (pressed and released)
  + the mouse cursor enters the unobscured part of component's geometry
  + the mouse cursor exits the unobscured part of component's geometry
* Mouse Motion Events
  + the mouse is moved
  + the mouse is dragged

A MouseEvent object is passed to every MouseListener or MouseAdapter object which is registered to receive the "interesting" mouse events using the component's addMouseListener method. (MouseAdapter objects implement the MouseListener interface.) Each such listener object gets a MouseEvent containing the mouse event.

A MouseEvent object is also passed to every MouseMotionListener or MouseMotionAdapter object which is registered to receive mouse motion events using the component's addMouseMotionListener method. (MouseMotionAdapter objects implement the MouseMotionListener interface.) Each such listener object gets a MouseEvent containing the mouse motion event.

When a mouse button is clicked, events are generated and sent to the registered MouseListeners. The state of modal keys can be retrieved using [InputEvent.getModifiers()](http://docs.google.com/java/awt/event/InputEvent.html#getModifiers()) and [InputEvent.getModifiersEx()](http://docs.google.com/java/awt/event/InputEvent.html#getModifiersEx()). The button mask returned by [InputEvent.getModifiers()](http://docs.google.com/java/awt/event/InputEvent.html#getModifiers()) reflects only the button that changed state, not the current state of all buttons. (Note: Due to overlap in the values of ALT\_MASK/BUTTON2\_MASK and META\_MASK/BUTTON3\_MASK, this is not always true for mouse events involving modifier keys). To get the state of all buttons and modifier keys, use [InputEvent.getModifiersEx()](http://docs.google.com/java/awt/event/InputEvent.html#getModifiersEx()). The button which has changed state is returned by [getButton()](http://docs.google.com/java/awt/event/MouseEvent.html#getButton())

For example, if the first mouse button is pressed, events are sent in the following order:

**id**  **modifiers**  **button**    
 MOUSE\_PRESSED: BUTTON1\_MASK BUTTON1  
 MOUSE\_RELEASED: BUTTON1\_MASK BUTTON1  
 MOUSE\_CLICKED: BUTTON1\_MASK BUTTON1

When multiple mouse buttons are pressed, each press, release, and click results in a separate event.

For example, if the user presses **button 1** followed by **button 2**, and then releases them in the same order, the following sequence of events is generated:

**id**  **modifiers**  **button**    
 MOUSE\_PRESSED: BUTTON1\_MASK BUTTON1  
 MOUSE\_PRESSED: BUTTON2\_MASK BUTTON2  
 MOUSE\_RELEASED: BUTTON1\_MASK BUTTON1  
 MOUSE\_CLICKED: BUTTON1\_MASK BUTTON1  
 MOUSE\_RELEASED: BUTTON2\_MASK BUTTON2  
 MOUSE\_CLICKED: BUTTON2\_MASK BUTTON2

If **button 2** is released first, the MOUSE\_RELEASED/MOUSE\_CLICKED pair for BUTTON2\_MASK arrives first, followed by the pair for BUTTON1\_MASK.

MOUSE\_DRAGGED events are delivered to the Component in which the mouse button was pressed until the mouse button is released (regardless of whether the mouse position is within the bounds of the Component). Due to platform-dependent Drag&Drop implementations, MOUSE\_DRAGGED events may not be delivered during a native Drag&Drop operation. In a multi-screen environment mouse drag events are delivered to the Component even if the mouse position is outside the bounds of the GraphicsConfiguration associated with that Component. However, the reported position for mouse drag events in this case may differ from the actual mouse position:

* In a multi-screen environment without a virtual device:  
  The reported coordinates for mouse drag events are clipped to fit within the bounds of the GraphicsConfiguration associated with the Component.
* In a multi-screen environment with a virtual device:  
  The reported coordinates for mouse drag events are clipped to fit within the bounds of the virtual device associated with the Component.

**Since:** 1.1 **See Also:**[MouseAdapter](http://docs.google.com/java/awt/event/MouseAdapter.html), [MouseListener](http://docs.google.com/java/awt/event/MouseListener.html), [MouseMotionAdapter](http://docs.google.com/java/awt/event/MouseMotionAdapter.html), [MouseMotionListener](http://docs.google.com/java/awt/event/MouseMotionListener.html), [MouseWheelListener](http://docs.google.com/java/awt/event/MouseWheelListener.html), [Tutorial: Writing a Mouse Listener](http://java.sun.com/docs/books/tutorial/post1.0/ui/mouselistener.html), [Tutorial: Writing a Mouse Motion Listener](http://java.sun.com/docs/books/tutorial/post1.0/ui/mousemotionlistener.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.awt.event.MouseEvent)

| **Field Summary** | |
| --- | --- |
| static int | [**BUTTON1**](http://docs.google.com/java/awt/event/MouseEvent.html#BUTTON1)            Indicates mouse button #1; used by [getButton()](http://docs.google.com/java/awt/event/MouseEvent.html#getButton()). |
| static int | [**BUTTON2**](http://docs.google.com/java/awt/event/MouseEvent.html#BUTTON2)            Indicates mouse button #2; used by [getButton()](http://docs.google.com/java/awt/event/MouseEvent.html#getButton()). |
| static int | [**BUTTON3**](http://docs.google.com/java/awt/event/MouseEvent.html#BUTTON3)            Indicates mouse button #3; used by [getButton()](http://docs.google.com/java/awt/event/MouseEvent.html#getButton()). |
| static int | [**MOUSE\_CLICKED**](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_CLICKED)            The "mouse clicked" event. |
| static int | [**MOUSE\_DRAGGED**](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_DRAGGED)            The "mouse dragged" event. |
| static int | [**MOUSE\_ENTERED**](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_ENTERED)            The "mouse entered" event. |
| static int | [**MOUSE\_EXITED**](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_EXITED)            The "mouse exited" event. |
| static int | [**MOUSE\_FIRST**](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_FIRST)            The first number in the range of ids used for mouse events. |
| static int | [**MOUSE\_LAST**](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_LAST)            The last number in the range of ids used for mouse events. |
| static int | [**MOUSE\_MOVED**](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_MOVED)            The "mouse moved" event. |
| static int | [**MOUSE\_PRESSED**](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_PRESSED)            The "mouse pressed" event. |
| static int | [**MOUSE\_RELEASED**](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_RELEASED)            The "mouse released" event. |
| static int | [**MOUSE\_WHEEL**](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_WHEEL)            The "mouse wheel" event. |
| static int | [**NOBUTTON**](http://docs.google.com/java/awt/event/MouseEvent.html#NOBUTTON)            Indicates no mouse buttons; used by [getButton()](http://docs.google.com/java/awt/event/MouseEvent.html#getButton()). |

| **Fields inherited from class java.awt.event.**[**InputEvent**](http://docs.google.com/java/awt/event/InputEvent.html) |
| --- |
| [ALT\_DOWN\_MASK](http://docs.google.com/java/awt/event/InputEvent.html#ALT_DOWN_MASK), [ALT\_GRAPH\_DOWN\_MASK](http://docs.google.com/java/awt/event/InputEvent.html#ALT_GRAPH_DOWN_MASK), [ALT\_GRAPH\_MASK](http://docs.google.com/java/awt/event/InputEvent.html#ALT_GRAPH_MASK), [ALT\_MASK](http://docs.google.com/java/awt/event/InputEvent.html#ALT_MASK), [BUTTON1\_DOWN\_MASK](http://docs.google.com/java/awt/event/InputEvent.html#BUTTON1_DOWN_MASK), [BUTTON1\_MASK](http://docs.google.com/java/awt/event/InputEvent.html#BUTTON1_MASK), [BUTTON2\_DOWN\_MASK](http://docs.google.com/java/awt/event/InputEvent.html#BUTTON2_DOWN_MASK), [BUTTON2\_MASK](http://docs.google.com/java/awt/event/InputEvent.html#BUTTON2_MASK), [BUTTON3\_DOWN\_MASK](http://docs.google.com/java/awt/event/InputEvent.html#BUTTON3_DOWN_MASK), [BUTTON3\_MASK](http://docs.google.com/java/awt/event/InputEvent.html#BUTTON3_MASK), [CTRL\_DOWN\_MASK](http://docs.google.com/java/awt/event/InputEvent.html#CTRL_DOWN_MASK), [CTRL\_MASK](http://docs.google.com/java/awt/event/InputEvent.html#CTRL_MASK), [META\_DOWN\_MASK](http://docs.google.com/java/awt/event/InputEvent.html#META_DOWN_MASK), [META\_MASK](http://docs.google.com/java/awt/event/InputEvent.html#META_MASK), [SHIFT\_DOWN\_MASK](http://docs.google.com/java/awt/event/InputEvent.html#SHIFT_DOWN_MASK), [SHIFT\_MASK](http://docs.google.com/java/awt/event/InputEvent.html#SHIFT_MASK) |

| **Fields inherited from class java.awt.event.**[**ComponentEvent**](http://docs.google.com/java/awt/event/ComponentEvent.html) |
| --- |
| [COMPONENT\_FIRST](http://docs.google.com/java/awt/event/ComponentEvent.html#COMPONENT_FIRST), [COMPONENT\_HIDDEN](http://docs.google.com/java/awt/event/ComponentEvent.html#COMPONENT_HIDDEN), [COMPONENT\_LAST](http://docs.google.com/java/awt/event/ComponentEvent.html#COMPONENT_LAST), [COMPONENT\_MOVED](http://docs.google.com/java/awt/event/ComponentEvent.html#COMPONENT_MOVED), [COMPONENT\_RESIZED](http://docs.google.com/java/awt/event/ComponentEvent.html#COMPONENT_RESIZED), [COMPONENT\_SHOWN](http://docs.google.com/java/awt/event/ComponentEvent.html#COMPONENT_SHOWN) |

| **Fields inherited from class java.awt.**[**AWTEvent**](http://docs.google.com/java/awt/AWTEvent.html) |
| --- |
| [ACTION\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#ACTION_EVENT_MASK), [ADJUSTMENT\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#ADJUSTMENT_EVENT_MASK), [COMPONENT\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#COMPONENT_EVENT_MASK), [consumed](http://docs.google.com/java/awt/AWTEvent.html#consumed), [CONTAINER\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#CONTAINER_EVENT_MASK), [FOCUS\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#FOCUS_EVENT_MASK), [HIERARCHY\_BOUNDS\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#HIERARCHY_BOUNDS_EVENT_MASK), [HIERARCHY\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#HIERARCHY_EVENT_MASK), [id](http://docs.google.com/java/awt/AWTEvent.html#id), [INPUT\_METHOD\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#INPUT_METHOD_EVENT_MASK), [INVOCATION\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#INVOCATION_EVENT_MASK), [ITEM\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#ITEM_EVENT_MASK), [KEY\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#KEY_EVENT_MASK), [MOUSE\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#MOUSE_EVENT_MASK), [MOUSE\_MOTION\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#MOUSE_MOTION_EVENT_MASK), [MOUSE\_WHEEL\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#MOUSE_WHEEL_EVENT_MASK), [PAINT\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#PAINT_EVENT_MASK), [RESERVED\_ID\_MAX](http://docs.google.com/java/awt/AWTEvent.html#RESERVED_ID_MAX), [TEXT\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#TEXT_EVENT_MASK), [WINDOW\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#WINDOW_EVENT_MASK), [WINDOW\_FOCUS\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#WINDOW_FOCUS_EVENT_MASK), [WINDOW\_STATE\_EVENT\_MASK](http://docs.google.com/java/awt/AWTEvent.html#WINDOW_STATE_EVENT_MASK) |

| **Fields inherited from class java.util.**[**EventObject**](http://docs.google.com/java/util/EventObject.html) |
| --- |
| [source](http://docs.google.com/java/util/EventObject.html#source) |

| **Constructor Summary** | |
| --- | --- |
| [**MouseEvent**](http://docs.google.com/java/awt/event/MouseEvent.html#MouseEvent(java.awt.Component,%20int,%20long,%20int,%20int,%20int,%20int,%20boolean))([Component](http://docs.google.com/java/awt/Component.html) source, int id, long when, int modifiers, int x, int y, int clickCount, boolean popupTrigger)            Constructs a MouseEvent object with the specified source component, type, modifiers, coordinates, and click count. |
| [**MouseEvent**](http://docs.google.com/java/awt/event/MouseEvent.html#MouseEvent(java.awt.Component,%20int,%20long,%20int,%20int,%20int,%20int,%20boolean,%20int))([Component](http://docs.google.com/java/awt/Component.html) source, int id, long when, int modifiers, int x, int y, int clickCount, boolean popupTrigger, int button)            Constructs a MouseEvent object with the specified source component, type, modifiers, coordinates, and click count. |
| [**MouseEvent**](http://docs.google.com/java/awt/event/MouseEvent.html#MouseEvent(java.awt.Component,%20int,%20long,%20int,%20int,%20int,%20int,%20int,%20int,%20boolean,%20int))([Component](http://docs.google.com/java/awt/Component.html) source, int id, long when, int modifiers, int x, int y, int xAbs, int yAbs, int clickCount, boolean popupTrigger, int button)            Constructs a MouseEvent object with the specified source component, type, modifiers, coordinates, absolute coordinates, and click count. |

| **Method Summary** | |
| --- | --- |
| int | [**getButton**](http://docs.google.com/java/awt/event/MouseEvent.html#getButton())()            Returns which, if any, of the mouse buttons has changed state. |
| int | [**getClickCount**](http://docs.google.com/java/awt/event/MouseEvent.html#getClickCount())()            Returns the number of mouse clicks associated with this event. |
| [Point](http://docs.google.com/java/awt/Point.html) | [**getLocationOnScreen**](http://docs.google.com/java/awt/event/MouseEvent.html#getLocationOnScreen())()            Returns the absolute x, y position of the event. |
| static [String](http://docs.google.com/java/lang/String.html) | [**getMouseModifiersText**](http://docs.google.com/java/awt/event/MouseEvent.html#getMouseModifiersText(int))(int modifiers)            Returns a String describing the modifier keys and mouse buttons that were down during the event, such as "Shift", or "Ctrl+Shift". |
| [Point](http://docs.google.com/java/awt/Point.html) | [**getPoint**](http://docs.google.com/java/awt/event/MouseEvent.html#getPoint())()            Returns the x,y position of the event relative to the source component. |
| int | [**getX**](http://docs.google.com/java/awt/event/MouseEvent.html#getX())()            Returns the horizontal x position of the event relative to the source component. |
| int | [**getXOnScreen**](http://docs.google.com/java/awt/event/MouseEvent.html#getXOnScreen())()            Returns the absolute horizontal x position of the event. |
| int | [**getY**](http://docs.google.com/java/awt/event/MouseEvent.html#getY())()            Returns the vertical y position of the event relative to the source component. |
| int | [**getYOnScreen**](http://docs.google.com/java/awt/event/MouseEvent.html#getYOnScreen())()            Returns the absolute vertical y position of the event. |
| boolean | [**isPopupTrigger**](http://docs.google.com/java/awt/event/MouseEvent.html#isPopupTrigger())()            Returns whether or not this mouse event is the popup menu trigger event for the platform. |
| [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/java/awt/event/MouseEvent.html#paramString())()            Returns a parameter string identifying this event. |
| void | [**translatePoint**](http://docs.google.com/java/awt/event/MouseEvent.html#translatePoint(int,%20int))(int x, int y)            Translates the event's coordinates to a new position by adding specified x (horizontal) and y (vertical) offsets. |

| **Methods inherited from class java.awt.event.**[**InputEvent**](http://docs.google.com/java/awt/event/InputEvent.html) |
| --- |
| [consume](http://docs.google.com/java/awt/event/InputEvent.html#consume()), [getModifiers](http://docs.google.com/java/awt/event/InputEvent.html#getModifiers()), [getModifiersEx](http://docs.google.com/java/awt/event/InputEvent.html#getModifiersEx()), [getModifiersExText](http://docs.google.com/java/awt/event/InputEvent.html#getModifiersExText(int)), [getWhen](http://docs.google.com/java/awt/event/InputEvent.html#getWhen()), [isAltDown](http://docs.google.com/java/awt/event/InputEvent.html#isAltDown()), [isAltGraphDown](http://docs.google.com/java/awt/event/InputEvent.html#isAltGraphDown()), [isConsumed](http://docs.google.com/java/awt/event/InputEvent.html#isConsumed()), [isControlDown](http://docs.google.com/java/awt/event/InputEvent.html#isControlDown()), [isMetaDown](http://docs.google.com/java/awt/event/InputEvent.html#isMetaDown()), [isShiftDown](http://docs.google.com/java/awt/event/InputEvent.html#isShiftDown()) |

| **Methods inherited from class java.awt.event.**[**ComponentEvent**](http://docs.google.com/java/awt/event/ComponentEvent.html) |
| --- |
| [getComponent](http://docs.google.com/java/awt/event/ComponentEvent.html#getComponent()) |

| **Methods inherited from class java.awt.**[**AWTEvent**](http://docs.google.com/java/awt/AWTEvent.html) |
| --- |
| [getID](http://docs.google.com/java/awt/AWTEvent.html#getID()), [setSource](http://docs.google.com/java/awt/AWTEvent.html#setSource(java.lang.Object)), [toString](http://docs.google.com/java/awt/AWTEvent.html#toString()) |

| **Methods inherited from class java.util.**[**EventObject**](http://docs.google.com/java/util/EventObject.html) |
| --- |
| [getSource](http://docs.google.com/java/util/EventObject.html#getSource()) |

| **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** |
| --- |

### MOUSE\_FIRST

public static final int **MOUSE\_FIRST**

The first number in the range of ids used for mouse events.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.event.MouseEvent.MOUSE_FIRST)

### MOUSE\_LAST

public static final int **MOUSE\_LAST**

The last number in the range of ids used for mouse events.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.event.MouseEvent.MOUSE_LAST)

### MOUSE\_CLICKED

public static final int **MOUSE\_CLICKED**

The "mouse clicked" event. This MouseEvent occurs when a mouse button is pressed and released.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.event.MouseEvent.MOUSE_CLICKED)

### MOUSE\_PRESSED

public static final int **MOUSE\_PRESSED**

The "mouse pressed" event. This MouseEvent occurs when a mouse button is pushed down.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.event.MouseEvent.MOUSE_PRESSED)

### MOUSE\_RELEASED

public static final int **MOUSE\_RELEASED**

The "mouse released" event. This MouseEvent occurs when a mouse button is let up.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.event.MouseEvent.MOUSE_RELEASED)

### MOUSE\_MOVED

public static final int **MOUSE\_MOVED**

The "mouse moved" event. This MouseEvent occurs when the mouse position changes.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.event.MouseEvent.MOUSE_MOVED)

### MOUSE\_ENTERED

public static final int **MOUSE\_ENTERED**

The "mouse entered" event. This MouseEvent occurs when the mouse cursor enters the unobscured part of component's geometry.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.event.MouseEvent.MOUSE_ENTERED)

### MOUSE\_EXITED

public static final int **MOUSE\_EXITED**

The "mouse exited" event. This MouseEvent occurs when the mouse cursor exits the unobscured part of component's geometry.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.event.MouseEvent.MOUSE_EXITED)

### MOUSE\_DRAGGED

public static final int **MOUSE\_DRAGGED**

The "mouse dragged" event. This MouseEvent occurs when the mouse position changes while a mouse button is pressed.

**See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.event.MouseEvent.MOUSE_DRAGGED)

### MOUSE\_WHEEL

public static final int **MOUSE\_WHEEL**

The "mouse wheel" event. This is the only MouseWheelEvent. It occurs when a mouse equipped with a wheel has its wheel rotated.

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.event.MouseEvent.MOUSE_WHEEL)

### NOBUTTON

public static final int **NOBUTTON**

Indicates no mouse buttons; used by [getButton()](http://docs.google.com/java/awt/event/MouseEvent.html#getButton()).

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.event.MouseEvent.NOBUTTON)

### BUTTON1

public static final int **BUTTON1**

Indicates mouse button #1; used by [getButton()](http://docs.google.com/java/awt/event/MouseEvent.html#getButton()).

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.event.MouseEvent.BUTTON1)

### BUTTON2

public static final int **BUTTON2**

Indicates mouse button #2; used by [getButton()](http://docs.google.com/java/awt/event/MouseEvent.html#getButton()).

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.event.MouseEvent.BUTTON2)

### BUTTON3

public static final int **BUTTON3**

Indicates mouse button #3; used by [getButton()](http://docs.google.com/java/awt/event/MouseEvent.html#getButton()).

**Since:** 1.4 **See Also:**[Constant Field Values](http://docs.google.com/constant-values.html#java.awt.event.MouseEvent.BUTTON3)

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

### MouseEvent

public **MouseEvent**([Component](http://docs.google.com/java/awt/Component.html) source,  
 int id,  
 long when,  
 int modifiers,  
 int x,  
 int y,  
 int clickCount,  
 boolean popupTrigger,  
 int button)

Constructs a MouseEvent object with the specified source component, type, modifiers, coordinates, and click count.

Note that passing in an invalid id results in unspecified behavior. Creating an invalid event (such as by using more than one of the old \_MASKs, or modifier/button values which don't match) results in unspecified behavior. An invocation of the form MouseEvent(source, id, when, modifiers, x, y, clickCount, popupTrigger, button) behaves in exactly the same way as the invocation [MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html#MouseEvent(java.awt.Component,%20int,%20long,%20int,%20int,%20int,%20int,%20int,%20int,%20boolean,%20int))(source, id, when, modifiers, x, y, xAbs, yAbs, clickCount, popupTrigger, button) where xAbs and yAbs defines as source's location on screen plus relative coordinates x and y. xAbs and yAbs are set to zero if the source is not showing. This method throws an IllegalArgumentException if source is null.

**Parameters:**source - the Component that originated the eventid - the integer that identifies the eventwhen - a long int that gives the time the event occurredmodifiers - the modifier keys down during event (e.g. shift, ctrl, alt, meta) Either extended \_DOWN\_MASK or old \_MASK modifiers should be used, but both models should not be mixed in one event. Use of the extended modifiers is preferred.x - the horizontal x coordinate for the mouse locationy - the vertical y coordinate for the mouse locationclickCount - the number of mouse clicks associated with eventpopupTrigger - a boolean, true if this event is a trigger for a popup menubutton - which of the mouse buttons has changed state. NOBUTTON, BUTTON1, BUTTON2 or BUTTON3. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if an invalid button value is passed in [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if source is null**Since:** 1.4

### MouseEvent

public **MouseEvent**([Component](http://docs.google.com/java/awt/Component.html) source,  
 int id,  
 long when,  
 int modifiers,  
 int x,  
 int y,  
 int clickCount,  
 boolean popupTrigger)

Constructs a MouseEvent object with the specified source component, type, modifiers, coordinates, and click count.

Note that passing in an invalid id results in unspecified behavior. An invocation of the form MouseEvent(source, id, when, modifiers, x, y, clickCount, popupTrigger) behaves in exactly the same way as the invocation [MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.html#MouseEvent(java.awt.Component,%20int,%20long,%20int,%20int,%20int,%20int,%20int,%20int,%20boolean,%20int))(source, id, when, modifiers, x, y, xAbs, yAbs, clickCount, popupTrigger, MouseEvent.NOBUTTON) where xAbs and yAbs defines as source's location on screen plus relative coordinates x and y. xAbs and yAbs are set to zero if the source is not showing. This method throws an IllegalArgumentException if source is null.

**Parameters:**source - the Component that originated the eventid - the integer that identifies the eventwhen - a long int that gives the time the event occurredmodifiers - the modifier keys down during event (e.g. shift, ctrl, alt, meta) Either extended \_DOWN\_MASK or old \_MASK modifiers should be used, but both models should not be mixed in one event. Use of the extended modifiers is preferred.x - the horizontal x coordinate for the mouse locationy - the vertical y coordinate for the mouse locationclickCount - the number of mouse clicks associated with eventpopupTrigger - a boolean, true if this event is a trigger for a popup menu **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if source is null

### MouseEvent

public **MouseEvent**([Component](http://docs.google.com/java/awt/Component.html) source,  
 int id,  
 long when,  
 int modifiers,  
 int x,  
 int y,  
 int xAbs,  
 int yAbs,  
 int clickCount,  
 boolean popupTrigger,  
 int button)

Constructs a MouseEvent object with the specified source component, type, modifiers, coordinates, absolute coordinates, and click count.

Note that passing in an invalid id results in unspecified behavior. Creating an invalid event (such as by using more than one of the old \_MASKs, or modifier/button values which don't match) results in unspecified behavior. Even if inconsistent values for relative and absolute coordinates are passed to the constructor, the mouse event instance is still created and no exception is thrown. This method throws an IllegalArgumentException if source is null.

**Parameters:**source - the Component that originated the eventid - the integer that identifies the eventwhen - a long int that gives the time the event occurredmodifiers - the modifier keys down during event (e.g. shift, ctrl, alt, meta) Either extended \_DOWN\_MASK or old \_MASK modifiers should be used, but both models should not be mixed in one event. Use of the extended modifiers is preferred.x - the horizontal x coordinate for the mouse locationy - the vertical y coordinate for the mouse locationxAbs - the absolute horizontal x coordinate for the mouse locationyAbs - the absolute vertical y coordinate for the mouse locationclickCount - the number of mouse clicks associated with eventpopupTrigger - a boolean, true if this event is a trigger for a popup menubutton - which of the mouse buttons has changed state. NOBUTTON, BUTTON1, BUTTON2 or BUTTON3. **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if an invalid button value is passed in [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if source is null**Since:** 1.6

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

### getLocationOnScreen

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

Returns the absolute x, y position of the event. In a virtual device multi-screen environment in which the desktop area could span multiple physical screen devices, these coordinates are relative to the virtual coordinate system. Otherwise, these coordinates are relative to the coordinate system associated with the Component's GraphicsConfiguration.

**Returns:**a Point object containing the absolute x and y coordinates.**Since:** 1.6 **See Also:**[GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html)

### getXOnScreen

public int **getXOnScreen**()

Returns the absolute horizontal x position of the event. In a virtual device multi-screen environment in which the desktop area could span multiple physical screen devices, this coordinate is relative to the virtual coordinate system. Otherwise, this coordinate is relative to the coordinate system associated with the Component's GraphicsConfiguration.

**Returns:**x an integer indicating absolute horizontal position.**Since:** 1.6 **See Also:**[GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html)

### getYOnScreen

public int **getYOnScreen**()

Returns the absolute vertical y position of the event. In a virtual device multi-screen environment in which the desktop area could span multiple physical screen devices, this coordinate is relative to the virtual coordinate system. Otherwise, this coordinate is relative to the coordinate system associated with the Component's GraphicsConfiguration.

**Returns:**y an integer indicating absolute vertical position.**Since:** 1.6 **See Also:**[GraphicsConfiguration](http://docs.google.com/java/awt/GraphicsConfiguration.html)

### getX

public int **getX**()

Returns the horizontal x position of the event relative to the source component.

**Returns:**x an integer indicating horizontal position relative to the component

### getY

public int **getY**()

Returns the vertical y position of the event relative to the source component.

**Returns:**y an integer indicating vertical position relative to the component

### getPoint

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

Returns the x,y position of the event relative to the source component.

**Returns:**a Point object containing the x and y coordinates relative to the source component

### translatePoint

public void **translatePoint**(int x,  
 int y)

Translates the event's coordinates to a new position by adding specified x (horizontal) and y (vertical) offsets.

**Parameters:**x - the horizontal x value to add to the current x coordinate positiony - the vertical y value to add to the current y coordinate position

### getClickCount

public int **getClickCount**()

Returns the number of mouse clicks associated with this event.

**Returns:**integer value for the number of clicks

### getButton

public int **getButton**()

Returns which, if any, of the mouse buttons has changed state.

**Returns:**one of the following constants: NOBUTTON, BUTTON1, BUTTON2 or BUTTON3.**Since:** 1.4

### isPopupTrigger

public boolean **isPopupTrigger**()

Returns whether or not this mouse event is the popup menu trigger event for the platform.

**Note**: Popup menus are triggered differently on different systems. Therefore, isPopupTrigger should be checked in both mousePressed and mouseReleased for proper cross-platform functionality.

**Returns:**boolean, true if this event is the popup menu trigger for this platform

### getMouseModifiersText

public static [String](http://docs.google.com/java/lang/String.html) **getMouseModifiersText**(int modifiers)

Returns a String describing the modifier keys and mouse buttons that were down during the event, such as "Shift", or "Ctrl+Shift". These strings can be localized by changing the awt.properties file.

Note that InputEvent.ALT\_MASK and InputEvent.BUTTON2\_MASK have the same value, so the string "Alt" is returned for both modifiers. Likewise, InputEvent.META\_MASK and InputEvent.BUTTON3\_MASK have the same value, so the string "Meta" is returned for both modifiers.

**Parameters:**modifiers - a modifier mask describing the modifier keys and mouse buttons that were down during the event **Returns:**string a text description of the combination of modifier keys and mouse buttons that were down during the event**Since:** 1.4 **See Also:**[InputEvent.getModifiersExText(int)](http://docs.google.com/java/awt/event/InputEvent.html#getModifiersExText(int))

### paramString

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

Returns a parameter string identifying this event. This method is useful for event-logging and for debugging.

**Overrides:**[paramString](http://docs.google.com/java/awt/event/ComponentEvent.html#paramString()) in class [ComponentEvent](http://docs.google.com/java/awt/event/ComponentEvent.html) **Returns:**a string identifying the event and its attributes

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MouseEvent.html) | [**Tree**](http://docs.google.com/package-tree.html) | [**Deprecated**](http://docs.google.com/deprecated-list.html) | [**Index**](http://docs.google.com/index-files/index-1.html) | [**Help**](http://docs.google.com/help-doc.html) | | --- | --- | --- | --- | --- | --- | --- | --- | | | ***Java™ Platform***  ***Standard Ed. 6*** |
| --- | --- | --- | --- | --- | --- | --- | --- | --- | --- | --- |
| [**PREV CLASS**](http://docs.google.com/java/awt/event/MouseAdapter.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/event/MouseListener.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/event/MouseEvent.html)    [**NO FRAMES**](http://docs.google.com/MouseEvent.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#4d34og8) | [METHOD](#2s8eyo1) | DETAIL: [FIELD](#1ksv4uv) | [CONSTR](#2p2csry) | [METHOD](#ihv636) |

[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).