| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/MouseWheelEvent.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/MouseMotionListener.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/event/MouseWheelListener.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/event/MouseWheelEvent.html)    [**NO FRAMES**](http://docs.google.com/MouseWheelEvent.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2s8eyo1) | [METHOD](#17dp8vu) | DETAIL: [FIELD](#2jxsxqh) | [CONSTR](#1y810tw) | [METHOD](#1ci93xb) |

## **java.awt.event**

Class MouseWheelEvent

[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](http://docs.google.com/java/awt/event/MouseEvent.html)  
 **java.awt.event.MouseWheelEvent**

**All Implemented Interfaces:** [Serializable](http://docs.google.com/java/io/Serializable.html)

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

An event which indicates that the mouse wheel was rotated in a component.

A wheel mouse is a mouse which has a wheel in place of the middle button. This wheel can be rotated towards or away from the user. Mouse wheels are most often used for scrolling, though other uses are possible.

A MouseWheelEvent object is passed to every MouseWheelListener object which registered to receive the "interesting" mouse events using the component's addMouseWheelListener method. Each such listener object gets a MouseEvent containing the mouse event.

Due to the mouse wheel's special relationship to scrolling Components, MouseWheelEvents are delivered somewhat differently than other MouseEvents. This is because while other MouseEvents usually affect a change on the Component directly under the mouse cursor (for instance, when clicking a button), MouseWheelEvents often have an effect away from the mouse cursor (moving the wheel while over a Component inside a ScrollPane should scroll one of the Scrollbars on the ScrollPane).

MouseWheelEvents start delivery from the Component underneath the mouse cursor. If MouseWheelEvents are not enabled on the Component, the event is delivered to the first ancestor Container with MouseWheelEvents enabled. This will usually be a ScrollPane with wheel scrolling enabled. The source Component and x,y coordinates will be relative to the event's final destination (the ScrollPane). This allows a complex GUI to be installed without modification into a ScrollPane, and for all MouseWheelEvents to be delivered to the ScrollPane for scrolling.

Some AWT Components are implemented using native widgets which display their own scrollbars and handle their own scrolling. The particular Components for which this is true will vary from platform to platform. When the mouse wheel is moved over one of these Components, the event is delivered straight to the native widget, and not propagated to ancestors.

Platforms offer customization of the amount of scrolling that should take place when the mouse wheel is moved. The two most common settings are to scroll a certain number of "units" (commonly lines of text in a text-based component) or an entire "block" (similar to page-up/page-down). The MouseWheelEvent offers methods for conforming to the underlying platform settings. These platform settings can be changed at any time by the user. MouseWheelEvents reflect the most recent settings.

**Since:** 1.4 **See Also:**[MouseWheelListener](http://docs.google.com/java/awt/event/MouseWheelListener.html), [ScrollPane](http://docs.google.com/java/awt/ScrollPane.html), [ScrollPane.setWheelScrollingEnabled(boolean)](http://docs.google.com/java/awt/ScrollPane.html#setWheelScrollingEnabled(boolean)), [JScrollPane](http://docs.google.com/javax/swing/JScrollPane.html), [JScrollPane.setWheelScrollingEnabled(boolean)](http://docs.google.com/javax/swing/JScrollPane.html#setWheelScrollingEnabled(boolean)), [Serialized Form](http://docs.google.com/serialized-form.html#java.awt.event.MouseWheelEvent)

| **Field Summary** | |
| --- | --- |
| static int | [**WHEEL\_BLOCK\_SCROLL**](http://docs.google.com/java/awt/event/MouseWheelEvent.html#WHEEL_BLOCK_SCROLL)            Constant representing scrolling by a "block" (like scrolling with page-up, page-down keys) |
| static int | [**WHEEL\_UNIT\_SCROLL**](http://docs.google.com/java/awt/event/MouseWheelEvent.html#WHEEL_UNIT_SCROLL)            Constant representing scrolling by "units" (like scrolling with the arrow keys) |

| **Fields inherited from class java.awt.event.**[**MouseEvent**](http://docs.google.com/java/awt/event/MouseEvent.html) |
| --- |
| [BUTTON1](http://docs.google.com/java/awt/event/MouseEvent.html#BUTTON1), [BUTTON2](http://docs.google.com/java/awt/event/MouseEvent.html#BUTTON2), [BUTTON3](http://docs.google.com/java/awt/event/MouseEvent.html#BUTTON3), [MOUSE\_CLICKED](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_CLICKED), [MOUSE\_DRAGGED](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_DRAGGED), [MOUSE\_ENTERED](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_ENTERED), [MOUSE\_EXITED](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_EXITED), [MOUSE\_FIRST](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_FIRST), [MOUSE\_LAST](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_LAST), [MOUSE\_MOVED](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_MOVED), [MOUSE\_PRESSED](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_PRESSED), [MOUSE\_RELEASED](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_RELEASED), [MOUSE\_WHEEL](http://docs.google.com/java/awt/event/MouseEvent.html#MOUSE_WHEEL), [NOBUTTON](http://docs.google.com/java/awt/event/MouseEvent.html#NOBUTTON) |

| **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** | |
| --- | --- |
| [**MouseWheelEvent**](http://docs.google.com/java/awt/event/MouseWheelEvent.html#MouseWheelEvent(java.awt.Component,%20int,%20long,%20int,%20int,%20int,%20int,%20boolean,%20int,%20int,%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 scrollType, int scrollAmount, int wheelRotation)            Constructs a MouseWheelEvent object with the specified source component, type, modifiers, coordinates, scroll type, scroll amount, and wheel rotation. |
| [**MouseWheelEvent**](http://docs.google.com/java/awt/event/MouseWheelEvent.html#MouseWheelEvent(java.awt.Component,%20int,%20long,%20int,%20int,%20int,%20int,%20int,%20int,%20boolean,%20int,%20int,%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 scrollType, int scrollAmount, int wheelRotation)            Constructs a MouseWheelEvent object with the specified source component, type, modifiers, coordinates, absolute coordinates, scroll type, scroll amount, and wheel rotation. |

| **Method Summary** | |
| --- | --- |
| int | [**getScrollAmount**](http://docs.google.com/java/awt/event/MouseWheelEvent.html#getScrollAmount())()            Returns the number of units that should be scrolled per click of mouse wheel rotation. |
| int | [**getScrollType**](http://docs.google.com/java/awt/event/MouseWheelEvent.html#getScrollType())()            Returns the type of scrolling that should take place in response to this event. |
| int | [**getUnitsToScroll**](http://docs.google.com/java/awt/event/MouseWheelEvent.html#getUnitsToScroll())()            This is a convenience method to aid in the implementation of the common-case MouseWheelListener - to scroll a ScrollPane or JScrollPane by an amount which conforms to the platform settings. |
| int | [**getWheelRotation**](http://docs.google.com/java/awt/event/MouseWheelEvent.html#getWheelRotation())()            Returns the number of "clicks" the mouse wheel was rotated. |
| [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/java/awt/event/MouseWheelEvent.html#paramString())()            Returns a parameter string identifying this event. |

| **Methods inherited from class java.awt.event.**[**MouseEvent**](http://docs.google.com/java/awt/event/MouseEvent.html) |
| --- |
| [getButton](http://docs.google.com/java/awt/event/MouseEvent.html#getButton()), [getClickCount](http://docs.google.com/java/awt/event/MouseEvent.html#getClickCount()), [getLocationOnScreen](http://docs.google.com/java/awt/event/MouseEvent.html#getLocationOnScreen()), [getMouseModifiersText](http://docs.google.com/java/awt/event/MouseEvent.html#getMouseModifiersText(int)), [getPoint](http://docs.google.com/java/awt/event/MouseEvent.html#getPoint()), [getX](http://docs.google.com/java/awt/event/MouseEvent.html#getX()), [getXOnScreen](http://docs.google.com/java/awt/event/MouseEvent.html#getXOnScreen()), [getY](http://docs.google.com/java/awt/event/MouseEvent.html#getY()), [getYOnScreen](http://docs.google.com/java/awt/event/MouseEvent.html#getYOnScreen()), [isPopupTrigger](http://docs.google.com/java/awt/event/MouseEvent.html#isPopupTrigger()), [translatePoint](http://docs.google.com/java/awt/event/MouseEvent.html#translatePoint(int,%20int)) |

| **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** |
| --- |

### WHEEL\_UNIT\_SCROLL

public static final int **WHEEL\_UNIT\_SCROLL**

Constant representing scrolling by "units" (like scrolling with the arrow keys)

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

### WHEEL\_BLOCK\_SCROLL

public static final int **WHEEL\_BLOCK\_SCROLL**

Constant representing scrolling by a "block" (like scrolling with page-up, page-down keys)

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

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

### MouseWheelEvent

public **MouseWheelEvent**([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 scrollType,  
 int scrollAmount,  
 int wheelRotation)

Constructs a MouseWheelEvent object with the specified source component, type, modifiers, coordinates, scroll type, scroll amount, and wheel rotation.

Absolute coordinates xAbs and yAbs are set to source's location on screen plus relative coordinates x and y. xAbs and yAbs are set to zero if the source is not showing.

Note that passing in an invalid id results in unspecified behavior. 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 that gives the time the event occurredmodifiers - the modifier keys down during event (shift, ctrl, alt, meta)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-menuscrollType - the type of scrolling which should take place in response to this event; valid values are WHEEL\_UNIT\_SCROLL and WHEEL\_BLOCK\_SCROLLscrollAmount - for scrollType WHEEL\_UNIT\_SCROLL, the number of units to be scrolledwheelRotation - the amount that the mouse wheel was rotated (the number of "clicks") **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if source is null**See Also:**[MouseEvent.MouseEvent(java.awt.Component, int, long, int, int, int, int, boolean)](http://docs.google.com/java/awt/event/MouseEvent.html#MouseEvent(java.awt.Component,%20int,%20long,%20int,%20int,%20int,%20int,%20boolean)), [MouseEvent.MouseEvent(java.awt.Component, int, long, int, int, int, int, int, int, boolean, int)](http://docs.google.com/java/awt/event/MouseEvent.html#MouseEvent(java.awt.Component,%20int,%20long,%20int,%20int,%20int,%20int,%20int,%20int,%20boolean,%20int))

### MouseWheelEvent

public **MouseWheelEvent**([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 scrollType,  
 int scrollAmount,  
 int wheelRotation)

Constructs a MouseWheelEvent object with the specified source component, type, modifiers, coordinates, absolute coordinates, scroll type, scroll amount, and wheel rotation.

Note that passing in an invalid id results in unspecified behavior. This method throws an IllegalArgumentException if source is null.

Even if inconsistent values for relative and absolute coordinates are passed to the constructor, the MouseWheelEvent instance is still created and no exception is thrown.

**Parameters:**source - the Component that originated the eventid - the integer that identifies the eventwhen - a long that gives the time the event occurredmodifiers - the modifier keys down during event (shift, ctrl, alt, meta)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-menuscrollType - the type of scrolling which should take place in response to this event; valid values are WHEEL\_UNIT\_SCROLL and WHEEL\_BLOCK\_SCROLLscrollAmount - for scrollType WHEEL\_UNIT\_SCROLL, the number of units to be scrolledwheelRotation - the amount that the mouse wheel was rotated (the number of "clicks") **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if source is null**Since:** 1.6 **See Also:**[MouseEvent.MouseEvent(java.awt.Component, int, long, int, int, int, int, boolean)](http://docs.google.com/java/awt/event/MouseEvent.html#MouseEvent(java.awt.Component,%20int,%20long,%20int,%20int,%20int,%20int,%20boolean)), [MouseEvent.MouseEvent(java.awt.Component, int, long, int, int, int, int, int, int, boolean, int)](http://docs.google.com/java/awt/event/MouseEvent.html#MouseEvent(java.awt.Component,%20int,%20long,%20int,%20int,%20int,%20int,%20int,%20int,%20boolean,%20int))

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

### getScrollType

public int **getScrollType**()

Returns the type of scrolling that should take place in response to this event. This is determined by the native platform. Legal values are:

* MouseWheelEvent.WHEEL\_UNIT\_SCROLL
* MouseWheelEvent.WHEEL\_BLOCK\_SCROLL

**Returns:**either MouseWheelEvent.WHEEL\_UNIT\_SCROLL or MouseWheelEvent.WHEEL\_BLOCK\_SCROLL, depending on the configuration of the native platform.**See Also:**[Adjustable.getUnitIncrement()](http://docs.google.com/java/awt/Adjustable.html#getUnitIncrement()), [Adjustable.getBlockIncrement()](http://docs.google.com/java/awt/Adjustable.html#getBlockIncrement()), [Scrollable.getScrollableUnitIncrement(java.awt.Rectangle, int, int)](http://docs.google.com/javax/swing/Scrollable.html#getScrollableUnitIncrement(java.awt.Rectangle,%20int,%20int)), [Scrollable.getScrollableBlockIncrement(java.awt.Rectangle, int, int)](http://docs.google.com/javax/swing/Scrollable.html#getScrollableBlockIncrement(java.awt.Rectangle,%20int,%20int))

### getScrollAmount

public int **getScrollAmount**()

Returns the number of units that should be scrolled per click of mouse wheel rotation. Only valid if getScrollType returns MouseWheelEvent.WHEEL\_UNIT\_SCROLL

**Returns:**number of units to scroll, or an undefined value if getScrollType returns MouseWheelEvent.WHEEL\_BLOCK\_SCROLL**See Also:**[getScrollType()](http://docs.google.com/java/awt/event/MouseWheelEvent.html#getScrollType())

### getWheelRotation

public int **getWheelRotation**()

Returns the number of "clicks" the mouse wheel was rotated.

**Returns:**negative values if the mouse wheel was rotated up/away from the user, and positive values if the mouse wheel was rotated down/ towards the user

### getUnitsToScroll

public int **getUnitsToScroll**()

This is a convenience method to aid in the implementation of the common-case MouseWheelListener - to scroll a ScrollPane or JScrollPane by an amount which conforms to the platform settings. (Note, however, that ScrollPane and JScrollPane already have this functionality built in.)

This method returns the number of units to scroll when scroll type is MouseWheelEvent.WHEEL\_UNIT\_SCROLL, and should only be called if getScrollType returns MouseWheelEvent.WHEEL\_UNIT\_SCROLL.

Direction of scroll, amount of wheel movement, and platform settings for wheel scrolling are all accounted for. This method does not and cannot take into account value of the Adjustable/Scrollable unit increment, as this will vary among scrolling components.

A simplified example of how this method might be used in a listener:

mouseWheelMoved(MouseWheelEvent event) {  
 ScrollPane sp = getScrollPaneFromSomewhere();   
 Adjustable adj = sp.getVAdjustable()  
 if (MouseWheelEvent.getScrollType() == WHEEL\_UNIT\_SCROLL) {  
 int totalScrollAmount =  
 event.getUnitsToScroll() \*  
 adj.getUnitIncrement();  
 adj.setValue(adj.getValue() + totalScrollAmount);  
 }  
 }

**Returns:**the number of units to scroll based on the direction and amount of mouse wheel rotation, and on the wheel scrolling settings of the native platform**See Also:**[getScrollType()](http://docs.google.com/java/awt/event/MouseWheelEvent.html#getScrollType()), [getScrollAmount()](http://docs.google.com/java/awt/event/MouseWheelEvent.html#getScrollAmount()), [MouseWheelListener](http://docs.google.com/java/awt/event/MouseWheelListener.html), [Adjustable](http://docs.google.com/java/awt/Adjustable.html), [Adjustable.getUnitIncrement()](http://docs.google.com/java/awt/Adjustable.html#getUnitIncrement()), [Scrollable](http://docs.google.com/javax/swing/Scrollable.html), [Scrollable.getScrollableUnitIncrement(java.awt.Rectangle, int, int)](http://docs.google.com/javax/swing/Scrollable.html#getScrollableUnitIncrement(java.awt.Rectangle,%20int,%20int)), [ScrollPane](http://docs.google.com/java/awt/ScrollPane.html), [ScrollPane.setWheelScrollingEnabled(boolean)](http://docs.google.com/java/awt/ScrollPane.html#setWheelScrollingEnabled(boolean)), [JScrollPane](http://docs.google.com/javax/swing/JScrollPane.html), [JScrollPane.setWheelScrollingEnabled(boolean)](http://docs.google.com/javax/swing/JScrollPane.html#setWheelScrollingEnabled(boolean))

### 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/MouseEvent.html#paramString()) in class [MouseEvent](http://docs.google.com/java/awt/event/MouseEvent.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/MouseWheelEvent.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/MouseMotionListener.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/event/MouseWheelListener.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/event/MouseWheelEvent.html)    [**NO FRAMES**](http://docs.google.com/MouseWheelEvent.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2s8eyo1) | [METHOD](#17dp8vu) | DETAIL: [FIELD](#2jxsxqh) | [CONSTR](#1y810tw) | [METHOD](#1ci93xb) |

[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).