

# Class `java.awt.Event`

```
java.lang.Object
|
+----java.awt.Event
```

---

public class **Event**  
extends [Object](#)

Event is a platform-independent class that encapsulates events from the local Graphical User Interface(GUI) platform.

**Version:**  
1.27 10/02/95  
**Author:**  
Sami Shaio

---

## Variable Index

- o **ACTION\_EVENT**  
An action event.
- o **ALT\_MASK**  
The alt modifier constant.
- o **CTRL\_MASK**  
The control modifier constant.
- o **DOWN**  
The down arrow key.
- o **END**  
The end key.
- o **ESC**  
The ESC key
- o **F1**  
The F1 function key
- o **F10**  
The F10 function key
- o **F11**  
The F11 function key
- o **F12**

- o The F12 function key
- o **F2** The F2 function key
- o **F3** The F3 function key
- o **F4** The F4 function key
- o **F5** The F5 function key
- o **F6** The F6 function key
- o **F7** The F7 function key
- o **F8** The F8 function key
- o **F9** The F9 function key
- o **GOT\_FOCUS** A component gained the focus.
- o **HOME** The home key.
- o **KEY\_ACTION** The key action keyboard event.
- o **KEY\_ACTION\_RELEASE** The key action keyboard event.
- o **KEY\_PRESS** The key press keyboard event.
- o **KEY\_RELEASE** The key release keyboard event.
- o **LEFT** The left arrow key.
- o **LIST\_DESELECT**
- o **LIST\_SELECT**
- o **LOAD\_FILE** A file loading event.
- o **LOST\_FOCUS** A component lost the focus.
- o **META\_MASK** The meta modifier constant.
- o **MOUSE\_DOWN** The mouse down event.
- o **MOUSE\_DRAG** The mouse drag event.
- o **MOUSE\_ENTER** The mouse enter event.
- o **MOUSE\_EXIT** The mouse exit event.
- o **MOUSE\_MOVE** The mouse move event.

- o **MOUSE\_UP**  
The mouse up event.
- o **PGDN**  
The page down key.
- o **PGUP**  
The page up key.
- o **RIGHT**  
The right arrow key.
- o **SAVE\_FILE**  
A file saving event.
- o **SCROLL\_ABSOLUTE**  
The absolute scroll event.
- o **SCROLL\_LINE\_DOWN**  
The line down scroll event.
- o **SCROLL\_LINE\_UP**  
The line up scroll event.
- o **SCROLL\_PAGE\_DOWN**  
The page down scroll event.
- o **SCROLL\_PAGE\_UP**  
The page up scroll event.
- o **SHIFT\_MASK**  
The shift modifier constant.
- o **UP**  
The up arrow key.
- o **WINDOW\_DEICONIFY**  
The de-iconify window event.
- o **WINDOW\_DESTROY**  
The destroy window event.
- o **WINDOW\_EXPOSE**  
The expose window event.
- o **WINDOW\_ICONIFY**  
The iconify window event.
- o **WINDOW\_MOVED**  
The move window event.
- o **arg**  
An arbitrary argument.
- o **evt**  
The next event.
- o **id**  
The type of this event.
- o **key**  
The key that was pressed in a keyboard event.
- o **modifiers**  
The state of the modifier keys.
- o **target**  
The target component.
- o **when**  
The time stamp.
- o **x**

The x coordinate of the event.

o **y**

The y coordinate of the event.

## Constructor Index

o **Event**(Object, long, int, int, int, int, int, Object)

Constructs an event with the specified target component, time stamp, event type, x and y coordinates, keyboard key, state of the modifier keys and argument.

o **Event**(Object, long, int, int, int, int, int)

Constructs an event with the specified target component, time stamp, event type, x and y coordinates, keyboard key, state of the modifier keys and an argument set to null.

o **Event**(Object, int, Object)

Constructs an event with the specified target component, event type, and argument.

## Method Index

o **controlDown**()

Checks if the control key is down.

o **metaDown**()

Checks if the meta key is down.

o **paramString**()

Returns the parameter String of this Event.

o **shiftDown**()

Checks if the shift key is down.

o **toString**()

Returns the String representation of this Event's values.

o **translate**(int, int)

Translates an event relative to the given component.

## Variables

o **SHIFT\_MASK**

```
public final static int SHIFT_MASK
```

The shift modifier constant.

o **CTRL\_MASK**

```
public final static int CTRL_MASK
```

The control modifier constant.

o **META\_MASK**

```
public final static int META_MASK
```

The meta modifier constant.

#### o **ALT\_MASK**

```
public final static int ALT_MASK
```

The alt modifier constant.

#### o **HOME**

```
public final static int HOME
```

The home key.

#### o **END**

```
public final static int END
```

The end key.

#### o **PGUP**

```
public final static int PGUP
```

The page up key.

#### o **PGDN**

```
public final static int PGDN
```

The page down key.

#### o **UP**

```
public final static int UP
```

The up arrow key.

#### o **DOWN**

```
public final static int DOWN
```

The down arrow key.

#### o **LEFT**

```
public final static int LEFT
```

The left arrow key.

o **RIGHT**

```
public final static int RIGHT
```

The right arrow key.

o **F1**

```
public final static int F1
```

The F1 function key

o **F2**

```
public final static int F2
```

The F2 function key

o **F3**

```
public final static int F3
```

The F3 function key

o **F4**

```
public final static int F4
```

The F4 function key

o **F5**

```
public final static int F5
```

The F5 function key

o **F6**

```
public final static int F6
```

The F6 function key

o **F7**

```
public final static int F7
```

The F7 function key

o **F8**

```
public final static int F8
```

The F8 function key

o **F9**

```
public final static int F9
```

The F9 function key

o **F10**

```
public final static int F10
```

The F10 function key

o **F11**

```
public final static int F11
```

The F11 function key

o **F12**

```
public final static int F12
```

The F12 function key

o **ESC**

```
public final static int ESC
```

The ESC key

o **WINDOW\_DESTROY**

```
public final static int WINDOW_DESTROY
```

The destroy window event.

o **WINDOW\_EXPOSE**

```
public final static int WINDOW_EXPOSE
```

The expose window event.

o **WINDOW\_ICONIFY**

```
public final static int WINDOW_ICONIFY
```

The iconify window event.

o **WINDOW\_DEICONIFY**

```
public final static int WINDOW_DEICONIFY
```

The de-iconify window event.

o **WINDOW\_MOVED**

```
public final static int WINDOW_MOVED
```

The move window event.

o **KEY\_PRESS**

```
public final static int KEY_PRESS
```

The key press keyboard event.

o **KEY\_RELEASE**

```
public final static int KEY_RELEASE
```

The key release keyboard event.

o **KEY\_ACTION**

```
public final static int KEY_ACTION
```

The key action keyboard event.

o **KEY\_ACTION\_RELEASE**

```
public final static int KEY_ACTION_RELEASE
```

The key action keyboard event.

o **MOUSE\_DOWN**

```
public final static int MOUSE_DOWN
```

The mouse down event.

o **MOUSE\_UP**

```
public final static int MOUSE_UP
```

The mouse up event.

#### o **MOUSE\_MOVE**

```
public final static int MOUSE_MOVE
```

The mouse move event.

#### o **MOUSE\_ENTER**

```
public final static int MOUSE_ENTER
```

The mouse enter event.

#### o **MOUSE\_EXIT**

```
public final static int MOUSE_EXIT
```

The mouse exit event.

#### o **MOUSE\_DRAG**

```
public final static int MOUSE_DRAG
```

The mouse drag event.

#### o **SCROLL\_LINE\_UP**

```
public final static int SCROLL_LINE_UP
```

The line up scroll event.

#### o **SCROLL\_LINE\_DOWN**

```
public final static int SCROLL_LINE_DOWN
```

The line down scroll event.

#### o **SCROLL\_PAGE\_UP**

```
public final static int SCROLL_PAGE_UP
```

The page up scroll event.

#### o **SCROLL\_PAGE\_DOWN**

```
public final static int SCROLL_PAGE_DOWN
```

The page down scroll event.

#### o **SCROLL\_ABSOLUTE**

```
public final static int SCROLL_ABSOLUTE
```

The absolute scroll event.

#### o **LIST\_SELECT**

```
public final static int LIST_SELECT
```

#### o **LIST\_DESELECT**

```
public final static int LIST_DESELECT
```

#### o **ACTION\_EVENT**

```
public final static int ACTION_EVENT
```

An action event.

#### o **LOAD\_FILE**

```
public final static int LOAD_FILE
```

A file loading event.

#### o **SAVE\_FILE**

```
public final static int SAVE_FILE
```

A file saving event.

#### o **GOT\_FOCUS**

```
public final static int GOT_FOCUS
```

A component gained the focus.

#### o **LOST\_FOCUS**

```
public final static int LOST_FOCUS
```

A component lost the focus.

#### o **target**

```
public Object target
```

The target component.

**o when**

```
public long when
```

The time stamp.

**o id**

```
public int id
```

The type of this event.

**o x**

```
public int x
```

The x coordinate of the event.

**o y**

```
public int y
```

The y coordinate of the event.

**o key**

```
public int key
```

The key that was pressed in a keyboard event.

**o modifiers**

```
public int modifiers
```

The state of the modifier keys.

**o arg**

```
public Object arg
```

An arbitrary argument.

**o evt**

```
public Event evt
```

The next event. Used when putting events into a linked list.

## Constructors

### o Event

```
public Event(Object target,  
            long when,  
            int id,  
            int x,  
            int y,  
            int key,  
            int modifiers,  
            Object arg)
```

Constructs an event with the specified target component, time stamp, event type, x and y coordinates, keyboard key, state of the modifier keys and argument.

**Parameters:**

target – the target component  
when – the time stamp  
id – the event type  
x – the x coordinate  
y – the y coordinate  
key – the key pressed in a keyboard event  
modifiers – the state of the modifier keys  
arg – the specified argument

### o Event

```
public Event(Object target,  
            long when,  
            int id,  
            int x,  
            int y,  
            int key,  
            int modifiers)
```

Constructs an event with the specified target component, time stamp, event type, x and y coordinates, keyboard key, state of the modifier keys and an argument set to null.

**Parameters:**

target – the target component  
when – the time stamp  
id – the event type  
x – the x coordinate  
y – the y coordinate  
key – the key pressed in a keyboard event  
modifiers – the state of the modifier keys

## o Event

```
public Event(Object target,  
            int id,  
            Object arg)
```

Constructs an event with the specified target component, event type, and argument.

**Parameters:**

target – the target component

id – the event type

arg – the specified argument

## Methods

### o translate

```
public void translate(int x,  
                    int y)
```

Translates an event relative to the given component. This involves at a minimum translating the coordinates so they make sense within the given component. It may also involve translating a region in the case of an expose event.

**Parameters:**

x – the x coordinate

y – the y coordinate

### o shiftDown

```
public boolean shiftDown()
```

Checks if the shift key is down.

**See Also:**

[modifiers](#), [controlDown](#), [metaDown](#)

### o controlDown

```
public boolean controlDown()
```

Checks if the control key is down.

**See Also:**

[modifiers](#), [shiftDown](#), [metaDown](#)

### o metaDown

```
public boolean metaDown()
```

Checks if the meta key is down.

**See Also:**

[modifiers](#), [shiftDown](#), [controlDown](#)

o **paramString**

```
protected String paramString()
```

Returns the parameter String of this Event.

o **toString**

```
public String toString()
```

Returns the String representation of this Event's values.

**Overrides:**

[toString](#) in class [Object](#)

---

[All Packages](#)

[This Package](#)

[Previous](#)

[Next](#)