| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/InputMethodEvent.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/InputEvent.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/event/InputMethodListener.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/event/InputMethodEvent.html)    [**NO FRAMES**](http://docs.google.com/InputMethodEvent.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#3rdcrjn) | [CONSTR](#44sinio) | [METHOD](#1y810tw) |

## **java.awt.event**

Class InputMethodEvent

[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.InputMethodEvent**

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

public class **InputMethodEvent**extends [AWTEvent](http://docs.google.com/java/awt/AWTEvent.html)

Input method events contain information about text that is being composed using an input method. Whenever the text changes, the input method sends an event. If the text component that's currently using the input method is an active client, the event is dispatched to that component. Otherwise, it is dispatched to a separate composition window.

The text included with the input method event consists of two parts: committed text and composed text. Either part may be empty. The two parts together replace any uncommitted composed text sent in previous events, or the currently selected committed text. Committed text should be integrated into the text component's persistent data, it will not be sent again. Composed text may be sent repeatedly, with changes to reflect the user's editing operations. Committed text always precedes composed text.

**Since:** 1.2 **See Also:**[Serialized Form](http://docs.google.com/serialized-form.html#java.awt.event.InputMethodEvent)

| **Field Summary** | |
| --- | --- |
| static int | [**CARET\_POSITION\_CHANGED**](http://docs.google.com/java/awt/event/InputMethodEvent.html#CARET_POSITION_CHANGED)            The event type indicating a changed insertion point in input method text. |
| static int | [**INPUT\_METHOD\_FIRST**](http://docs.google.com/java/awt/event/InputMethodEvent.html#INPUT_METHOD_FIRST)            Marks the first integer id for the range of input method event ids. |
| static int | [**INPUT\_METHOD\_LAST**](http://docs.google.com/java/awt/event/InputMethodEvent.html#INPUT_METHOD_LAST)            Marks the last integer id for the range of input method event ids. |
| static int | [**INPUT\_METHOD\_TEXT\_CHANGED**](http://docs.google.com/java/awt/event/InputMethodEvent.html#INPUT_METHOD_TEXT_CHANGED)            The event type indicating changed input method text. |

| **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** | |
| --- | --- |
| [**InputMethodEvent**](http://docs.google.com/java/awt/event/InputMethodEvent.html#InputMethodEvent(java.awt.Component,%20int,%20java.text.AttributedCharacterIterator,%20int,%20java.awt.font.TextHitInfo,%20java.awt.font.TextHitInfo))([Component](http://docs.google.com/java/awt/Component.html) source, int id, [AttributedCharacterIterator](http://docs.google.com/java/text/AttributedCharacterIterator.html) text, int committedCharacterCount, [TextHitInfo](http://docs.google.com/java/awt/font/TextHitInfo.html) caret, [TextHitInfo](http://docs.google.com/java/awt/font/TextHitInfo.html) visiblePosition)            Constructs an InputMethodEvent with the specified source component, type, text, caret, and visiblePosition. |
| [**InputMethodEvent**](http://docs.google.com/java/awt/event/InputMethodEvent.html#InputMethodEvent(java.awt.Component,%20int,%20long,%20java.text.AttributedCharacterIterator,%20int,%20java.awt.font.TextHitInfo,%20java.awt.font.TextHitInfo))([Component](http://docs.google.com/java/awt/Component.html) source, int id, long when, [AttributedCharacterIterator](http://docs.google.com/java/text/AttributedCharacterIterator.html) text, int committedCharacterCount, [TextHitInfo](http://docs.google.com/java/awt/font/TextHitInfo.html) caret, [TextHitInfo](http://docs.google.com/java/awt/font/TextHitInfo.html) visiblePosition)            Constructs an InputMethodEvent with the specified source component, type, time, text, caret, and visiblePosition. |
| [**InputMethodEvent**](http://docs.google.com/java/awt/event/InputMethodEvent.html#InputMethodEvent(java.awt.Component,%20int,%20java.awt.font.TextHitInfo,%20java.awt.font.TextHitInfo))([Component](http://docs.google.com/java/awt/Component.html) source, int id, [TextHitInfo](http://docs.google.com/java/awt/font/TextHitInfo.html) caret, [TextHitInfo](http://docs.google.com/java/awt/font/TextHitInfo.html) visiblePosition)            Constructs an InputMethodEvent with the specified source component, type, caret, and visiblePosition. |

| **Method Summary** | |
| --- | --- |
| void | [**consume**](http://docs.google.com/java/awt/event/InputMethodEvent.html#consume())()            Consumes this event so that it will not be processed in the default manner by the source which originated it. |
| [TextHitInfo](http://docs.google.com/java/awt/font/TextHitInfo.html) | [**getCaret**](http://docs.google.com/java/awt/event/InputMethodEvent.html#getCaret())()            Gets the caret. |
| int | [**getCommittedCharacterCount**](http://docs.google.com/java/awt/event/InputMethodEvent.html#getCommittedCharacterCount())()            Gets the number of committed characters in the text. |
| [AttributedCharacterIterator](http://docs.google.com/java/text/AttributedCharacterIterator.html) | [**getText**](http://docs.google.com/java/awt/event/InputMethodEvent.html#getText())()            Gets the combined committed and composed text. |
| [TextHitInfo](http://docs.google.com/java/awt/font/TextHitInfo.html) | [**getVisiblePosition**](http://docs.google.com/java/awt/event/InputMethodEvent.html#getVisiblePosition())()            Gets the position that's most important to be visible. |
| long | [**getWhen**](http://docs.google.com/java/awt/event/InputMethodEvent.html#getWhen())()            Returns the time stamp of when this event occurred. |
| boolean | [**isConsumed**](http://docs.google.com/java/awt/event/InputMethodEvent.html#isConsumed())()            Returns whether or not this event has been consumed. |
| [String](http://docs.google.com/java/lang/String.html) | [**paramString**](http://docs.google.com/java/awt/event/InputMethodEvent.html#paramString())()            Returns a parameter string identifying this event. |

| **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** |
| --- |

### INPUT\_METHOD\_FIRST

public static final int **INPUT\_METHOD\_FIRST**

Marks the first integer id for the range of input method event ids.

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

### INPUT\_METHOD\_TEXT\_CHANGED

public static final int **INPUT\_METHOD\_TEXT\_CHANGED**

The event type indicating changed input method text. This event is generated by input methods while processing input.

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

### CARET\_POSITION\_CHANGED

public static final int **CARET\_POSITION\_CHANGED**

The event type indicating a changed insertion point in input method text. This event is generated by input methods while processing input if only the caret changed.

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

### INPUT\_METHOD\_LAST

public static final int **INPUT\_METHOD\_LAST**

Marks the last integer id for the range of input method event ids.

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

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

### InputMethodEvent

public **InputMethodEvent**([Component](http://docs.google.com/java/awt/Component.html) source,  
 int id,  
 long when,  
 [AttributedCharacterIterator](http://docs.google.com/java/text/AttributedCharacterIterator.html) text,  
 int committedCharacterCount,  
 [TextHitInfo](http://docs.google.com/java/awt/font/TextHitInfo.html) caret,  
 [TextHitInfo](http://docs.google.com/java/awt/font/TextHitInfo.html) visiblePosition)

Constructs an InputMethodEvent with the specified source component, type, time, text, caret, and visiblePosition.

The offsets of caret and visiblePosition are relative to the current composed text; that is, the composed text within text if this is an INPUT\_METHOD\_TEXT\_CHANGED event, the composed text within the text of the preceding INPUT\_METHOD\_TEXT\_CHANGED event otherwise.

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

**Parameters:**source - the object where the event originatedid - the event typewhen - a long integer that specifies the time the event occurredtext - the combined committed and composed text, committed text first; must be null when the event type is CARET\_POSITION\_CHANGED; may be null for INPUT\_METHOD\_TEXT\_CHANGED if there's no committed or composed textcommittedCharacterCount - the number of committed characters in the textcaret - the caret (a.k.a. insertion point); null if there's no caret within current composed textvisiblePosition - the position that's most important to be visible; null if there's no recommendation for a visible position within current composed text **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if id is not in the range INPUT\_METHOD\_FIRST..INPUT\_METHOD\_LAST; or if id is CARET\_POSITION\_CHANGED and text is not null; or if committedCharacterCount is not in the range 0..(text.getEndIndex() - text.getBeginIndex()) [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if source is null**Since:** 1.4

### InputMethodEvent

public **InputMethodEvent**([Component](http://docs.google.com/java/awt/Component.html) source,  
 int id,  
 [AttributedCharacterIterator](http://docs.google.com/java/text/AttributedCharacterIterator.html) text,  
 int committedCharacterCount,  
 [TextHitInfo](http://docs.google.com/java/awt/font/TextHitInfo.html) caret,  
 [TextHitInfo](http://docs.google.com/java/awt/font/TextHitInfo.html) visiblePosition)

Constructs an InputMethodEvent with the specified source component, type, text, caret, and visiblePosition.

The offsets of caret and visiblePosition are relative to the current composed text; that is, the composed text within text if this is an INPUT\_METHOD\_TEXT\_CHANGED event, the composed text within the text of the preceding INPUT\_METHOD\_TEXT\_CHANGED event otherwise. The time stamp for this event is initialized by invoking [EventQueue.getMostRecentEventTime()](http://docs.google.com/java/awt/EventQueue.html#getMostRecentEventTime()).

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

**Parameters:**source - the object where the event originatedid - the event typetext - the combined committed and composed text, committed text first; must be null when the event type is CARET\_POSITION\_CHANGED; may be null for INPUT\_METHOD\_TEXT\_CHANGED if there's no committed or composed textcommittedCharacterCount - the number of committed characters in the textcaret - the caret (a.k.a. insertion point); null if there's no caret within current composed textvisiblePosition - the position that's most important to be visible; null if there's no recommendation for a visible position within current composed text **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if id is not in the range INPUT\_METHOD\_FIRST..INPUT\_METHOD\_LAST; or if id is CARET\_POSITION\_CHANGED and text is not null; or if committedCharacterCount is not in the range 0..(text.getEndIndex() - text.getBeginIndex()) [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if source is null

### InputMethodEvent

public **InputMethodEvent**([Component](http://docs.google.com/java/awt/Component.html) source,  
 int id,  
 [TextHitInfo](http://docs.google.com/java/awt/font/TextHitInfo.html) caret,  
 [TextHitInfo](http://docs.google.com/java/awt/font/TextHitInfo.html) visiblePosition)

Constructs an InputMethodEvent with the specified source component, type, caret, and visiblePosition. The text is set to null, committedCharacterCount to 0.

The offsets of caret and visiblePosition are relative to the current composed text; that is, the composed text within the text of the preceding INPUT\_METHOD\_TEXT\_CHANGED event if the event being constructed as a CARET\_POSITION\_CHANGED event. For an INPUT\_METHOD\_TEXT\_CHANGED event without text, caret and visiblePosition must be null. The time stamp for this event is initialized by invoking [EventQueue.getMostRecentEventTime()](http://docs.google.com/java/awt/EventQueue.html#getMostRecentEventTime()).

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

**Parameters:**source - the object where the event originatedid - the event typecaret - the caret (a.k.a. insertion point); null if there's no caret within current composed textvisiblePosition - the position that's most important to be visible; null if there's no recommendation for a visible position within current composed text **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if id is not in the range INPUT\_METHOD\_FIRST..INPUT\_METHOD\_LAST [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if source is null

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

### getText

public [AttributedCharacterIterator](http://docs.google.com/java/text/AttributedCharacterIterator.html) **getText**()

Gets the combined committed and composed text. Characters from index 0 to index getCommittedCharacterCount() - 1 are committed text, the remaining characters are composed text.

**Returns:**the text. Always null for CARET\_POSITION\_CHANGED; may be null for INPUT\_METHOD\_TEXT\_CHANGED if there's no composed or committed text.

### getCommittedCharacterCount

public int **getCommittedCharacterCount**()

Gets the number of committed characters in the text.

### getCaret

public [TextHitInfo](http://docs.google.com/java/awt/font/TextHitInfo.html) **getCaret**()

Gets the caret.

The offset of the caret is relative to the current composed text; that is, the composed text within getText() if this is an INPUT\_METHOD\_TEXT\_CHANGED event, the composed text within getText() of the preceding INPUT\_METHOD\_TEXT\_CHANGED event otherwise.

**Returns:**the caret (a.k.a. insertion point). Null if there's no caret within current composed text.

### getVisiblePosition

public [TextHitInfo](http://docs.google.com/java/awt/font/TextHitInfo.html) **getVisiblePosition**()

Gets the position that's most important to be visible.

The offset of the visible position is relative to the current composed text; that is, the composed text within getText() if this is an INPUT\_METHOD\_TEXT\_CHANGED event, the composed text within getText() of the preceding INPUT\_METHOD\_TEXT\_CHANGED event otherwise.

**Returns:**the position that's most important to be visible. Null if there's no recommendation for a visible position within current composed text.

### consume

public void **consume**()

Consumes this event so that it will not be processed in the default manner by the source which originated it.

**Overrides:**[consume](http://docs.google.com/java/awt/AWTEvent.html#consume()) in class [AWTEvent](http://docs.google.com/java/awt/AWTEvent.html)

### isConsumed

public boolean **isConsumed**()

Returns whether or not this event has been consumed.

**Overrides:**[isConsumed](http://docs.google.com/java/awt/AWTEvent.html#isConsumed()) in class [AWTEvent](http://docs.google.com/java/awt/AWTEvent.html) **See Also:**[consume()](http://docs.google.com/java/awt/event/InputMethodEvent.html#consume())

### getWhen

public long **getWhen**()

Returns the time stamp of when this event occurred.

**Returns:**this event's timestamp**Since:** 1.4

### 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. It contains the event ID in text form, the characters of the committed and composed text separated by "+", the number of committed characters, the caret, and the visible position.

**Overrides:**[paramString](http://docs.google.com/java/awt/AWTEvent.html#paramString()) in class [AWTEvent](http://docs.google.com/java/awt/AWTEvent.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/InputMethodEvent.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/InputEvent.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/event/InputMethodListener.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/event/InputMethodEvent.html)    [**NO FRAMES**](http://docs.google.com/InputMethodEvent.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#3dy6vkm) | [METHOD](#1t3h5sf) | DETAIL: [FIELD](#3rdcrjn) | [CONSTR](#44sinio) | [METHOD](#1y810tw) |

[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).