| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DragGestureEvent.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/dnd/DnDConstants.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/dnd/DragGestureListener.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/dnd/DragGestureEvent.html)    [**NO FRAMES**](http://docs.google.com/DragGestureEvent.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#3rdcrjn) |

## **java.awt.dnd**

Class DragGestureEvent

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 [java.util.EventObject](http://docs.google.com/java/util/EventObject.html)  
 **java.awt.dnd.DragGestureEvent**

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

public class **DragGestureEvent**extends [EventObject](http://docs.google.com/java/util/EventObject.html)

A DragGestureEvent is passed to DragGestureListener's dragGestureRecognized() method when a particular DragGestureRecognizer detects that a platform dependent drag initiating gesture has occurred on the Component that it is tracking.

**See Also:**[DragGestureRecognizer](http://docs.google.com/java/awt/dnd/DragGestureRecognizer.html), [DragGestureListener](http://docs.google.com/java/awt/dnd/DragGestureListener.html), [DragSource](http://docs.google.com/java/awt/dnd/DragSource.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.awt.dnd.DragGestureEvent)

| **Field Summary** | |
| --- | --- |

| **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** | |
| --- | --- |
| [**DragGestureEvent**](http://docs.google.com/java/awt/dnd/DragGestureEvent.html#DragGestureEvent(java.awt.dnd.DragGestureRecognizer,%20int,%20java.awt.Point,%20java.util.List))([DragGestureRecognizer](http://docs.google.com/java/awt/dnd/DragGestureRecognizer.html) dgr, int act, [Point](http://docs.google.com/java/awt/Point.html) ori, [List](http://docs.google.com/java/util/List.html)<? extends [InputEvent](http://docs.google.com/java/awt/event/InputEvent.html)> evs)            Construct a DragGestureEvent given the DragGestureRecognizer firing this event, an int representing the user's preferred action, a Point indicating the origin of the drag, and a List of events that comprise the gesture. |

| **Method Summary** | |
| --- | --- |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getComponent**](http://docs.google.com/java/awt/dnd/DragGestureEvent.html#getComponent())()            Returns the Component associated with this DragGestureEvent. |
| int | [**getDragAction**](http://docs.google.com/java/awt/dnd/DragGestureEvent.html#getDragAction())()            Returns an int representing the action selected by the user. |
| [Point](http://docs.google.com/java/awt/Point.html) | [**getDragOrigin**](http://docs.google.com/java/awt/dnd/DragGestureEvent.html#getDragOrigin())()            Returns a Point in the coordinates of the Component over which the drag originated. |
| [DragSource](http://docs.google.com/java/awt/dnd/DragSource.html) | [**getDragSource**](http://docs.google.com/java/awt/dnd/DragGestureEvent.html#getDragSource())()            Returns the DragSource. |
| [DragGestureRecognizer](http://docs.google.com/java/awt/dnd/DragGestureRecognizer.html) | [**getSourceAsDragGestureRecognizer**](http://docs.google.com/java/awt/dnd/DragGestureEvent.html#getSourceAsDragGestureRecognizer())()            Returns the source as a DragGestureRecognizer. |
| [InputEvent](http://docs.google.com/java/awt/event/InputEvent.html) | [**getTriggerEvent**](http://docs.google.com/java/awt/dnd/DragGestureEvent.html#getTriggerEvent())()            Returns the initial event that triggered the gesture. |
| [Iterator](http://docs.google.com/java/util/Iterator.html)<[InputEvent](http://docs.google.com/java/awt/event/InputEvent.html)> | [**iterator**](http://docs.google.com/java/awt/dnd/DragGestureEvent.html#iterator())()            Returns an Iterator for the events comprising the gesture. |
| void | [**startDrag**](http://docs.google.com/java/awt/dnd/DragGestureEvent.html#startDrag(java.awt.Cursor,%20java.awt.Image,%20java.awt.Point,%20java.awt.datatransfer.Transferable,%20java.awt.dnd.DragSourceListener))([Cursor](http://docs.google.com/java/awt/Cursor.html) dragCursor, [Image](http://docs.google.com/java/awt/Image.html) dragImage, [Point](http://docs.google.com/java/awt/Point.html) imageOffset, [Transferable](http://docs.google.com/java/awt/datatransfer/Transferable.html) transferable, [DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html) dsl)            Start the drag given the initial Cursor to display, a drag Image, the offset of the Image, the Transferable object, and the DragSourceListener to use. |
| void | [**startDrag**](http://docs.google.com/java/awt/dnd/DragGestureEvent.html#startDrag(java.awt.Cursor,%20java.awt.datatransfer.Transferable))([Cursor](http://docs.google.com/java/awt/Cursor.html) dragCursor, [Transferable](http://docs.google.com/java/awt/datatransfer/Transferable.html) transferable)            Starts the drag operation given the Cursor for this drag operation and the Transferable representing the source data for this drag operation. |
| void | [**startDrag**](http://docs.google.com/java/awt/dnd/DragGestureEvent.html#startDrag(java.awt.Cursor,%20java.awt.datatransfer.Transferable,%20java.awt.dnd.DragSourceListener))([Cursor](http://docs.google.com/java/awt/Cursor.html) dragCursor, [Transferable](http://docs.google.com/java/awt/datatransfer/Transferable.html) transferable, [DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html) dsl)            Starts the drag given the initial Cursor to display, the Transferable object, and the DragSourceListener to use. |
| [Object](http://docs.google.com/java/lang/Object.html)[] | [**toArray**](http://docs.google.com/java/awt/dnd/DragGestureEvent.html#toArray())()            Returns an Object array of the events comprising the drag gesture. |
| [Object](http://docs.google.com/java/lang/Object.html)[] | [**toArray**](http://docs.google.com/java/awt/dnd/DragGestureEvent.html#toArray(java.lang.Object%5B%5D))([Object](http://docs.google.com/java/lang/Object.html)[] array)            Returns an array of the events comprising the drag gesture. |

| **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()), [toString](http://docs.google.com/java/util/EventObject.html#toString()) |

| **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)) |

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

### DragGestureEvent

public **DragGestureEvent**([DragGestureRecognizer](http://docs.google.com/java/awt/dnd/DragGestureRecognizer.html) dgr,  
 int act,  
 [Point](http://docs.google.com/java/awt/Point.html) ori,  
 [List](http://docs.google.com/java/util/List.html)<? extends [InputEvent](http://docs.google.com/java/awt/event/InputEvent.html)> evs)

Construct a DragGestureEvent given the DragGestureRecognizer firing this event, an int representing the user's preferred action, a Point indicating the origin of the drag, and a List of events that comprise the gesture.

**Parameters:**dgr - The DragGestureRecognizer firing this eventact - The the user's preferred actionori - The origin of the dragevs - The List of events that comprise the gesture

**Throws:** IllegalArgumentException - if input parameters are null

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

### getSourceAsDragGestureRecognizer

public [DragGestureRecognizer](http://docs.google.com/java/awt/dnd/DragGestureRecognizer.html) **getSourceAsDragGestureRecognizer**()

Returns the source as a DragGestureRecognizer.

**Returns:**the source as a DragGestureRecognizer

### getComponent

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

Returns the Component associated with this DragGestureEvent.

**Returns:**the Component

### getDragSource

public [DragSource](http://docs.google.com/java/awt/dnd/DragSource.html) **getDragSource**()

Returns the DragSource.

**Returns:**the DragSource

### getDragOrigin

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

Returns a Point in the coordinates of the Component over which the drag originated.

**Returns:**the Point where the drag originated in Component coords.

### iterator

public [Iterator](http://docs.google.com/java/util/Iterator.html)<[InputEvent](http://docs.google.com/java/awt/event/InputEvent.html)> **iterator**()

Returns an Iterator for the events comprising the gesture.

**Returns:**an Iterator for the events comprising the gesture

### toArray

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

Returns an Object array of the events comprising the drag gesture.

**Returns:**an array of the events comprising the gesture

### toArray

public [Object](http://docs.google.com/java/lang/Object.html)[] **toArray**([Object](http://docs.google.com/java/lang/Object.html)[] array)

Returns an array of the events comprising the drag gesture.

**Parameters:**array - the array of EventObject sub(types)

**Returns:**an array of the events comprising the gesture

### getDragAction

public int **getDragAction**()

Returns an int representing the action selected by the user.

**Returns:**the action selected by the user

### getTriggerEvent

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

Returns the initial event that triggered the gesture.

**Returns:**the first "triggering" event in the sequence of the gesture

### startDrag

public void **startDrag**([Cursor](http://docs.google.com/java/awt/Cursor.html) dragCursor,  
 [Transferable](http://docs.google.com/java/awt/datatransfer/Transferable.html) transferable)  
 throws [InvalidDnDOperationException](http://docs.google.com/java/awt/dnd/InvalidDnDOperationException.html)

Starts the drag operation given the Cursor for this drag operation and the Transferable representing the source data for this drag operation.

If a null Cursor is specified no exception will be thrown and default drag cursors will be used instead.

If a null Transferable is specified NullPointerException will be thrown.

**Parameters:**dragCursor - The Cursor for this drag operationtransferable - The Transferable representing the source data for this drag operation. **Throws:** InvalidDnDOperationException - if the Drag and Drop system is unable to initiate a drag operation, or if the user attempts to start a drag while an existing drag operation is still executing. NullPointerException - if the Transferable is null. [InvalidDnDOperationException](http://docs.google.com/java/awt/dnd/InvalidDnDOperationException.html)**Since:** 1.4

### startDrag

public void **startDrag**([Cursor](http://docs.google.com/java/awt/Cursor.html) dragCursor,  
 [Transferable](http://docs.google.com/java/awt/datatransfer/Transferable.html) transferable,  
 [DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html) dsl)  
 throws [InvalidDnDOperationException](http://docs.google.com/java/awt/dnd/InvalidDnDOperationException.html)

Starts the drag given the initial Cursor to display, the Transferable object, and the DragSourceListener to use.

**Parameters:**dragCursor - The initial drag Cursortransferable - The source's Transferabledsl - The source's DragSourceListener

**Throws:** InvalidDnDOperationException - if the Drag and Drop system is unable to initiate a drag operation, or if the user attempts to start a drag while an existing drag operation is still executing. [InvalidDnDOperationException](http://docs.google.com/java/awt/dnd/InvalidDnDOperationException.html)

### startDrag

public void **startDrag**([Cursor](http://docs.google.com/java/awt/Cursor.html) dragCursor,  
 [Image](http://docs.google.com/java/awt/Image.html) dragImage,  
 [Point](http://docs.google.com/java/awt/Point.html) imageOffset,  
 [Transferable](http://docs.google.com/java/awt/datatransfer/Transferable.html) transferable,  
 [DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html) dsl)  
 throws [InvalidDnDOperationException](http://docs.google.com/java/awt/dnd/InvalidDnDOperationException.html)

Start the drag given the initial Cursor to display, a drag Image, the offset of the Image, the Transferable object, and the DragSourceListener to use.

**Parameters:**dragCursor - The initial drag CursordragImage - The source's dragImageimageOffset - The dragImage's offsettransferable - The source's Transferabledsl - The source's DragSourceListener

**Throws:** InvalidDnDOperationException - if the Drag and Drop system is unable to initiate a drag operation, or if the user attempts to start a drag while an existing drag operation is still executing. [InvalidDnDOperationException](http://docs.google.com/java/awt/dnd/InvalidDnDOperationException.html)

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DragGestureEvent.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/dnd/DnDConstants.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/dnd/DragGestureListener.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/dnd/DragGestureEvent.html)    [**NO FRAMES**](http://docs.google.com/DragGestureEvent.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#2et92p0) | [CONSTR](#tyjcwt) | [METHOD](#3dy6vkm) | DETAIL: FIELD | [CONSTR](#2s8eyo1) | [METHOD](#3rdcrjn) |

[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).