| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DragSourceContext.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/DragSourceAdapter.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/dnd/DragSourceDragEvent.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/dnd/DragSourceContext.html)    [**NO FRAMES**](http://docs.google.com/DragSourceContext.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

## **java.awt.dnd**

Class DragSourceContext

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

**All Implemented Interfaces:** [DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html), [DragSourceMotionListener](http://docs.google.com/java/awt/dnd/DragSourceMotionListener.html), [Serializable](http://docs.google.com/java/io/Serializable.html), [EventListener](http://docs.google.com/java/util/EventListener.html)

public class **DragSourceContext**extends [Object](http://docs.google.com/java/lang/Object.html)implements [DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html), [DragSourceMotionListener](http://docs.google.com/java/awt/dnd/DragSourceMotionListener.html), [Serializable](http://docs.google.com/java/io/Serializable.html)

The DragSourceContext class is responsible for managing the initiator side of the Drag and Drop protocol. In particular, it is responsible for managing drag event notifications to the DragSourceListeners and DragSourceMotionListeners, and providing the Transferable representing the source data for the drag operation.

Note that the DragSourceContext itself implements the DragSourceListener and DragSourceMotionListener interfaces. This is to allow the platform peer (the DragSourceContextPeer instance) created by the DragSource to notify the DragSourceContext of state changes in the ongoing operation. This allows the DragSourceContext to interpose itself between the platform and the listeners provided by the initiator of the drag operation.

**Since:** 1.2 **See Also:**[DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html), [DragSourceMotionListener](http://docs.google.com/java/awt/dnd/DragSourceMotionListener.html), [Serialized Form](http://docs.google.com/serialized-form.html#java.awt.dnd.DragSourceContext)

| **Field Summary** | |
| --- | --- |
| protected static int | [**CHANGED**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#CHANGED)            An int used by updateCurrentCursor() indicating that the user operation has changed. |
| protected static int | [**DEFAULT**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#DEFAULT)            An int used by updateCurrentCursor() indicating that the Cursor should change to the default (no drop) Cursor. |
| protected static int | [**ENTER**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#ENTER)            An int used by updateCurrentCursor() indicating that the Cursor has entered a DropTarget. |
| protected static int | [**OVER**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#OVER)            An int used by updateCurrentCursor() indicating that the Cursor is over a DropTarget. |

| **Constructor Summary** | |
| --- | --- |
| [**DragSourceContext**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#DragSourceContext(java.awt.dnd.peer.DragSourceContextPeer,%20java.awt.dnd.DragGestureEvent,%20java.awt.Cursor,%20java.awt.Image,%20java.awt.Point,%20java.awt.datatransfer.Transferable,%20java.awt.dnd.DragSourceListener))(java.awt.dnd.peer.DragSourceContextPeer dscp, [DragGestureEvent](http://docs.google.com/java/awt/dnd/DragGestureEvent.html) trigger, [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) offset, [Transferable](http://docs.google.com/java/awt/datatransfer/Transferable.html) t, [DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html) dsl)            Called from DragSource, this constructor creates a new DragSourceContext given the DragSourceContextPeer for this Drag, the DragGestureEvent that triggered the Drag, the initial Cursor to use for the Drag, an (optional) Image to display while the Drag is taking place, the offset of the Image origin from the hotspot at the instant of the triggering event, the Transferable subject data, and the DragSourceListener to use during the Drag and Drop operation. |

| **Method Summary** | |
| --- | --- |
| void | [**addDragSourceListener**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#addDragSourceListener(java.awt.dnd.DragSourceListener))([DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html) dsl)            Add a DragSourceListener to this DragSourceContext if one has not already been added. |
| void | [**dragDropEnd**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#dragDropEnd(java.awt.dnd.DragSourceDropEvent))([DragSourceDropEvent](http://docs.google.com/java/awt/dnd/DragSourceDropEvent.html) dsde)            Calls dragDropEnd on the DragSourceListeners registered with this DragSourceContext and with the associated DragSource, and passes them the specified DragSourceDropEvent. |
| void | [**dragEnter**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#dragEnter(java.awt.dnd.DragSourceDragEvent))([DragSourceDragEvent](http://docs.google.com/java/awt/dnd/DragSourceDragEvent.html) dsde)            Calls dragEnter on the DragSourceListeners registered with this DragSourceContext and with the associated DragSource, and passes them the specified DragSourceDragEvent. |
| void | [**dragExit**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#dragExit(java.awt.dnd.DragSourceEvent))([DragSourceEvent](http://docs.google.com/java/awt/dnd/DragSourceEvent.html) dse)            Calls dragExit on the DragSourceListeners registered with this DragSourceContext and with the associated DragSource, and passes them the specified DragSourceEvent. |
| void | [**dragMouseMoved**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#dragMouseMoved(java.awt.dnd.DragSourceDragEvent))([DragSourceDragEvent](http://docs.google.com/java/awt/dnd/DragSourceDragEvent.html) dsde)            Calls dragMouseMoved on the DragSourceMotionListeners registered with the DragSource associated with this DragSourceContext, and them passes the specified DragSourceDragEvent. |
| void | [**dragOver**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#dragOver(java.awt.dnd.DragSourceDragEvent))([DragSourceDragEvent](http://docs.google.com/java/awt/dnd/DragSourceDragEvent.html) dsde)            Calls dragOver on the DragSourceListeners registered with this DragSourceContext and with the associated DragSource, and passes them the specified DragSourceDragEvent. |
| void | [**dropActionChanged**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#dropActionChanged(java.awt.dnd.DragSourceDragEvent))([DragSourceDragEvent](http://docs.google.com/java/awt/dnd/DragSourceDragEvent.html) dsde)            Calls dropActionChanged on the DragSourceListeners registered with this DragSourceContext and with the associated DragSource, and passes them the specified DragSourceDragEvent. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getComponent**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#getComponent())()            Returns the Component associated with this DragSourceContext. |
| [Cursor](http://docs.google.com/java/awt/Cursor.html) | [**getCursor**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#getCursor())()            Returns the current drag Cursor. |
| [DragSource](http://docs.google.com/java/awt/dnd/DragSource.html) | [**getDragSource**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#getDragSource())()            Returns the DragSource that instantiated this DragSourceContext. |
| int | [**getSourceActions**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#getSourceActions())()            Returns a bitwise mask of DnDConstants that represent the set of drop actions supported by the drag source for the drag operation associated with this DragSourceContext. |
| [Transferable](http://docs.google.com/java/awt/datatransfer/Transferable.html) | [**getTransferable**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#getTransferable())()            Returns the Transferable associated with this DragSourceContext. |
| [DragGestureEvent](http://docs.google.com/java/awt/dnd/DragGestureEvent.html) | [**getTrigger**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#getTrigger())()            Returns the DragGestureEvent that initially triggered the drag. |
| void | [**removeDragSourceListener**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#removeDragSourceListener(java.awt.dnd.DragSourceListener))([DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html) dsl)            Removes the specified DragSourceListener from this DragSourceContext. |
| void | [**setCursor**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#setCursor(java.awt.Cursor))([Cursor](http://docs.google.com/java/awt/Cursor.html) c)            Sets the cursor for this drag operation to the specified Cursor. |
| void | [**transferablesFlavorsChanged**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#transferablesFlavorsChanged())()            Notifies the peer that the Transferable's DataFlavors have changed. |
| protected  void | [**updateCurrentCursor**](http://docs.google.com/java/awt/dnd/DragSourceContext.html#updateCurrentCursor(int,%20int,%20int))(int sourceAct, int targetAct, int status)            If the default drag cursor behavior is active, this method sets the default drag cursor for the specified actions supported by the drag source, the drop target action, and status, otherwise this method does nothing. |

| **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()), [toString](http://docs.google.com/java/lang/Object.html#toString()), [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** |
| --- |

### DEFAULT

protected static final int **DEFAULT**

An int used by updateCurrentCursor() indicating that the Cursor should change to the default (no drop) Cursor.

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

### ENTER

protected static final int **ENTER**

An int used by updateCurrentCursor() indicating that the Cursor has entered a DropTarget.

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

### OVER

protected static final int **OVER**

An int used by updateCurrentCursor() indicating that the Cursor is over a DropTarget.

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

### CHANGED

protected static final int **CHANGED**

An int used by updateCurrentCursor() indicating that the user operation has changed.

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

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

### DragSourceContext

public **DragSourceContext**(java.awt.dnd.peer.DragSourceContextPeer dscp,  
 [DragGestureEvent](http://docs.google.com/java/awt/dnd/DragGestureEvent.html) trigger,  
 [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) offset,  
 [Transferable](http://docs.google.com/java/awt/datatransfer/Transferable.html) t,  
 [DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html) dsl)

Called from DragSource, this constructor creates a new DragSourceContext given the DragSourceContextPeer for this Drag, the DragGestureEvent that triggered the Drag, the initial Cursor to use for the Drag, an (optional) Image to display while the Drag is taking place, the offset of the Image origin from the hotspot at the instant of the triggering event, the Transferable subject data, and the DragSourceListener to use during the Drag and Drop operation.

If DragSourceContextPeer is null NullPointerException is thrown.

If DragGestureEvent is null NullPointerException is thrown.

If Cursor is null no exception is thrown and the default drag cursor behavior is activated for this drag operation.

If Image is null no exception is thrown.

If Image is not null and the offset is null NullPointerException is thrown.

If Transferable is null NullPointerException is thrown.

If DragSourceListener is null no exception is thrown.

**Parameters:**dscp - the DragSourceContextPeer for this dragtrigger - the triggering eventdragCursor - the initial CursordragImage - the Image to drag (or null)offset - the offset of the image origin from the hotspot at the instant of the triggering eventt - the Transferabledsl - the DragSourceListener **Throws:** [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the Component associated with the trigger event is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the DragSource for the trigger event is null. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the drag action for the trigger event is DnDConstants.ACTION\_NONE. [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if the source actions for the DragGestureRecognizer associated with the trigger event are equal to DnDConstants.ACTION\_NONE. [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if dscp, trigger, or t are null, or if dragImage is non-null and offset is null

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

### getDragSource

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

Returns the DragSource that instantiated this DragSourceContext.

**Returns:**the DragSource that instantiated this DragSourceContext

### getComponent

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

Returns the Component associated with this DragSourceContext.

**Returns:**the Component that started the drag

### getTrigger

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

Returns the DragGestureEvent that initially triggered the drag.

**Returns:**the Event that triggered the drag

### getSourceActions

public int **getSourceActions**()

Returns a bitwise mask of DnDConstants that represent the set of drop actions supported by the drag source for the drag operation associated with this DragSourceContext.

**Returns:**the drop actions supported by the drag source

### setCursor

public void **setCursor**([Cursor](http://docs.google.com/java/awt/Cursor.html) c)

Sets the cursor for this drag operation to the specified Cursor. If the specified Cursor is null, the default drag cursor behavior is activated for this drag operation, otherwise it is deactivated.

**Parameters:**c - the Cursor to display, or null to activate the default drag cursor behavior

### getCursor

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

Returns the current drag Cursor.

**Returns:**the current drag Cursor

### addDragSourceListener

public void **addDragSourceListener**([DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html) dsl)  
 throws [TooManyListenersException](http://docs.google.com/java/util/TooManyListenersException.html)

Add a DragSourceListener to this DragSourceContext if one has not already been added. If a DragSourceListener already exists, this method throws a TooManyListenersException.

**Parameters:**dsl - the DragSourceListener to add. Note that while null is not prohibited, it is not acceptable as a parameter.

**Throws:** TooManyListenersException - if a DragSourceListener has already been added [TooManyListenersException](http://docs.google.com/java/util/TooManyListenersException.html)

### removeDragSourceListener

public void **removeDragSourceListener**([DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html) dsl)

Removes the specified DragSourceListener from this DragSourceContext.

**Parameters:**dsl - the DragSourceListener to remove; note that while null is not prohibited, it is not acceptable as a parameter

### transferablesFlavorsChanged

public void **transferablesFlavorsChanged**()

Notifies the peer that the Transferable's DataFlavors have changed.

### dragEnter

public void **dragEnter**([DragSourceDragEvent](http://docs.google.com/java/awt/dnd/DragSourceDragEvent.html) dsde)

Calls dragEnter on the DragSourceListeners registered with this DragSourceContext and with the associated DragSource, and passes them the specified DragSourceDragEvent.

**Specified by:**[dragEnter](http://docs.google.com/java/awt/dnd/DragSourceListener.html#dragEnter(java.awt.dnd.DragSourceDragEvent)) in interface [DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html) **Parameters:**dsde - the DragSourceDragEvent

### dragOver

public void **dragOver**([DragSourceDragEvent](http://docs.google.com/java/awt/dnd/DragSourceDragEvent.html) dsde)

Calls dragOver on the DragSourceListeners registered with this DragSourceContext and with the associated DragSource, and passes them the specified DragSourceDragEvent.

**Specified by:**[dragOver](http://docs.google.com/java/awt/dnd/DragSourceListener.html#dragOver(java.awt.dnd.DragSourceDragEvent)) in interface [DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html) **Parameters:**dsde - the DragSourceDragEvent

### dragExit

public void **dragExit**([DragSourceEvent](http://docs.google.com/java/awt/dnd/DragSourceEvent.html) dse)

Calls dragExit on the DragSourceListeners registered with this DragSourceContext and with the associated DragSource, and passes them the specified DragSourceEvent.

**Specified by:**[dragExit](http://docs.google.com/java/awt/dnd/DragSourceListener.html#dragExit(java.awt.dnd.DragSourceEvent)) in interface [DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html) **Parameters:**dse - the DragSourceEvent

### dropActionChanged

public void **dropActionChanged**([DragSourceDragEvent](http://docs.google.com/java/awt/dnd/DragSourceDragEvent.html) dsde)

Calls dropActionChanged on the DragSourceListeners registered with this DragSourceContext and with the associated DragSource, and passes them the specified DragSourceDragEvent.

**Specified by:**[dropActionChanged](http://docs.google.com/java/awt/dnd/DragSourceListener.html#dropActionChanged(java.awt.dnd.DragSourceDragEvent)) in interface [DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html) **Parameters:**dsde - the DragSourceDragEvent

### dragDropEnd

public void **dragDropEnd**([DragSourceDropEvent](http://docs.google.com/java/awt/dnd/DragSourceDropEvent.html) dsde)

Calls dragDropEnd on the DragSourceListeners registered with this DragSourceContext and with the associated DragSource, and passes them the specified DragSourceDropEvent.

**Specified by:**[dragDropEnd](http://docs.google.com/java/awt/dnd/DragSourceListener.html#dragDropEnd(java.awt.dnd.DragSourceDropEvent)) in interface [DragSourceListener](http://docs.google.com/java/awt/dnd/DragSourceListener.html) **Parameters:**dsde - the DragSourceDropEvent

### dragMouseMoved

public void **dragMouseMoved**([DragSourceDragEvent](http://docs.google.com/java/awt/dnd/DragSourceDragEvent.html) dsde)

Calls dragMouseMoved on the DragSourceMotionListeners registered with the DragSource associated with this DragSourceContext, and them passes the specified DragSourceDragEvent.

**Specified by:**[dragMouseMoved](http://docs.google.com/java/awt/dnd/DragSourceMotionListener.html#dragMouseMoved(java.awt.dnd.DragSourceDragEvent)) in interface [DragSourceMotionListener](http://docs.google.com/java/awt/dnd/DragSourceMotionListener.html) **Parameters:**dsde - the DragSourceDragEvent**Since:** 1.4

### getTransferable

public [Transferable](http://docs.google.com/java/awt/datatransfer/Transferable.html) **getTransferable**()

Returns the Transferable associated with this DragSourceContext.

**Returns:**the Transferable

### updateCurrentCursor

protected void **updateCurrentCursor**(int sourceAct,  
 int targetAct,  
 int status)

If the default drag cursor behavior is active, this method sets the default drag cursor for the specified actions supported by the drag source, the drop target action, and status, otherwise this method does nothing.

**Parameters:**sourceAct - the actions supported by the drag sourcetargetAct - the drop target actionstatus - one of the fields DEFAULT, ENTER, OVER, CHANGED

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DragSourceContext.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/DragSourceAdapter.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/dnd/DragSourceDragEvent.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/dnd/DragSourceContext.html)    [**NO FRAMES**](http://docs.google.com/DragSourceContext.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | [FIELD](#3znysh7) | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: [FIELD](#1t3h5sf) | [CONSTR](#26in1rg) | [METHOD](#35nkun2) |

[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).