| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/java/awt/datatransfer/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/java/awt/event/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/dnd/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

## Package java.awt.dnd

Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI.

**See:**

[**Description**](#3znysh7)

| **Interface Summary** | |
| --- | --- |
| [**Autoscroll**](http://docs.google.com/java/awt/dnd/Autoscroll.html) | During DnD operations it is possible that a user may wish to drop the subject of the operation on a region of a scrollable GUI control that is not currently visible to the user. |
| [**DragGestureListener**](http://docs.google.com/java/awt/dnd/DragGestureListener.html) | This interface is sourced from a DragGestureRecognizer and is invoked when an object of that (sub)class detects a drag initiating gesture. |
| [**DragSourceListener**](http://docs.google.com/java/awt/dnd/DragSourceListener.html) | The DragSourceListener defines the event interface for originators of Drag and Drop operations to track the state of the user's gesture, and to provide appropriate "drag over" feedback to the user throughout the Drag and Drop operation. |
| [**DragSourceMotionListener**](http://docs.google.com/java/awt/dnd/DragSourceMotionListener.html) | A listener interface for receiving mouse motion events during a drag operation. |
| [**DropTargetListener**](http://docs.google.com/java/awt/dnd/DropTargetListener.html) | The DropTargetListener interface is the callback interface used by the DropTarget class to provide notification of DnD operations that involve the subject DropTarget. |

| **Class Summary** | |
| --- | --- |
| [**DnDConstants**](http://docs.google.com/java/awt/dnd/DnDConstants.html) | This class contains constant values representing the type of action(s) to be performed by a Drag and Drop operation. |
| [**DragGestureEvent**](http://docs.google.com/java/awt/dnd/DragGestureEvent.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. |
| [**DragGestureRecognizer**](http://docs.google.com/java/awt/dnd/DragGestureRecognizer.html) | The DragGestureRecognizer is an abstract base class for the specification of a platform-dependent listener that can be associated with a particular Component in order to identify platform-dependent drag initiating gestures. |
| [**DragSource**](http://docs.google.com/java/awt/dnd/DragSource.html) | The DragSource is the entity responsible for the initiation of the Drag and Drop operation, and may be used in a number of scenarios: 1 default instance per JVM for the lifetime of that JVM. |
| [**DragSourceAdapter**](http://docs.google.com/java/awt/dnd/DragSourceAdapter.html) | An abstract adapter class for receiving drag source events. |
| [**DragSourceContext**](http://docs.google.com/java/awt/dnd/DragSourceContext.html) | The DragSourceContext class is responsible for managing the initiator side of the Drag and Drop protocol. |
| [**DragSourceDragEvent**](http://docs.google.com/java/awt/dnd/DragSourceDragEvent.html) | The DragSourceDragEvent is delivered from the DragSourceContextPeer, via the DragSourceContext, to the DragSourceListener registered with that DragSourceContext and with its associated DragSource. |
| [**DragSourceDropEvent**](http://docs.google.com/java/awt/dnd/DragSourceDropEvent.html) | The DragSourceDropEvent is delivered from the DragSourceContextPeer, via the DragSourceContext, to the dragDropEnd method of DragSourceListeners registered with that DragSourceContext and with its associated DragSource. |
| [**DragSourceEvent**](http://docs.google.com/java/awt/dnd/DragSourceEvent.html) | This class is the base class for DragSourceDragEvent and DragSourceDropEvent. |
| [**DropTarget**](http://docs.google.com/java/awt/dnd/DropTarget.html) | The DropTarget is associated with a Component when that Component wishes to accept drops during Drag and Drop operations. |
| [**DropTarget.DropTargetAutoScroller**](http://docs.google.com/java/awt/dnd/DropTarget.DropTargetAutoScroller.html) | this protected nested class implements autoscrolling |
| [**DropTargetAdapter**](http://docs.google.com/java/awt/dnd/DropTargetAdapter.html) | An abstract adapter class for receiving drop target events. |
| [**DropTargetContext**](http://docs.google.com/java/awt/dnd/DropTargetContext.html) | A DropTargetContext is created whenever the logical cursor associated with a Drag and Drop operation coincides with the visible geometry of a Component associated with a DropTarget. |
| [**DropTargetDragEvent**](http://docs.google.com/java/awt/dnd/DropTargetDragEvent.html) | The DropTargetDragEvent is delivered to a DropTargetListener via its dragEnter() and dragOver() methods. |
| [**DropTargetDropEvent**](http://docs.google.com/java/awt/dnd/DropTargetDropEvent.html) | The DropTargetDropEvent is delivered via the DropTargetListener drop() method. |
| [**DropTargetEvent**](http://docs.google.com/java/awt/dnd/DropTargetEvent.html) | The DropTargetEvent is the base class for both the DropTargetDragEvent and the DropTargetDropEvent. |
| [**MouseDragGestureRecognizer**](http://docs.google.com/java/awt/dnd/MouseDragGestureRecognizer.html) | This abstract subclass of DragGestureRecognizer defines a DragGestureRecognizer for mouse-based gestures. |

| **Exception Summary** | |
| --- | --- |
| [**InvalidDnDOperationException**](http://docs.google.com/java/awt/dnd/InvalidDnDOperationException.html) | This exception is thrown by various methods in the java.awt.dnd package. |

## Package java.awt.dnd Description

Drag and Drop is a direct manipulation gesture found in many Graphical User Interface systems that provides a mechanism to transfer information between two entities logically associated with presentation elements in the GUI. Normally driven by a physical gesture of a human user using an appropriate input device, Drag and Drop provides both a mechanism to enable continuous feedback regarding the possible outcome of any subsequent data transfer to the user during navigation over the presentation elements in the GUI, and the facilities to provide for any subsequent data negotiation and transfer.

This package defines the classes and interfaces necessary to perform Drag and Drop operations in Java. It defines classes for the drag-source and the drop-target, as well as events for transferring the data being dragged. This package also provides a means for giving visual feedback to the user throughout the duration of the Drag and Drop operation.

A typical Drag and Drop operation can be decomposed into the following states (not entirely sequentially):

* A DragSource comes into existence, associated with some presentation element (Component) in the GUI, to initiate a Drag and Drop of some potentially Transferable data.
* 1 or more DropTarget(s) come into/go out of existence, associated with presentation elements in the GUI (Components), potentially capable of consuming Transferable data types.
* A DragGestureRecognizer is obtained from the DragSource and is associated with a Component in order to track and identify any Drag initiating gesture by the user over the Component.
* A user makes a Drag gesture over the Component, which the registered DragGestureRecognizer detects, and notifies its DragGestureListener of.  
  Note: Although this API consistently refers to the stimulus for a drag and drop operation being a physical gesture by a human user, this does not preclude a programmatically driven DnD operation given the appropriate implementation of a DragSource. This package contains the abstract class MouseDragGestureRecognizer for recognizing mouse device gestures. Other abstract subclasses may be provided by the platform to support other input devices or particular Component class semantics.
* The DragGestureListener causes the DragSource to initiate the Drag and Drop operation on behalf of the user, perhaps animating the GUI Cursor and/or rendering an Image of the item(s) that are the subject of the operation.
* As the user gestures navigate over Component(s) in the GUI with associated DropTarget(s), the DragSource receives notifications in order to provide "Drag Over" feedback effects, and the DropTarget(s) receive notifications in order to provide "Drag Under" feedback effects based upon the operation(s) supported and the data type(s) involved.

The gesture itself moves a logical cursor across the GUI hierarchy, intersecting the geometry of GUI Component(s), possibly resulting in the logical "Drag" cursor entering, crossing, and subsequently leaving Component(s) and associated DropTarget(s).

The DragSource object manifests "Drag Over" feedback to the user, in the typical case by animating the GUI Cursor associated with the logical cursor.

DropTarget objects manifest "Drag Under" feedback to the user, in the typical case, by rendering animations into their associated GUI Component(s) under the GUI Cursor.

The determination of the feedback effects, and the ultimate success or failure of the data transfer, should one occur, is parameterized as follows:

* By the transfer "operation" selected by the user, and supported by both the DragSource and DropTarget: Copy, Move or Reference(link).
* By the intersection of the set of data types provided by the DragSource and the set of data types comprehensible by the DropTarget.
* When the user terminates the drag operation, normally resulting in a successful Drop, both the DragSource and DropTarget receive notifications that include, and result in the type negotiation and transfer of, the information associated with the DragSource via a Transferable object.

**Since:** 1.2

| | [**Overview**](http://docs.google.com/overview-summary.html) | **Package** | Class | [**Use**](http://docs.google.com/package-use.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 PACKAGE**](http://docs.google.com/java/awt/datatransfer/package-summary.html)   [**NEXT PACKAGE**](http://docs.google.com/java/awt/event/package-summary.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/dnd/package-summary.html)    [**NO FRAMES**](http://docs.google.com/package-summary.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |

[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).