| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DropTargetAdapter.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/DropTarget.DropTargetAutoScroller.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/dnd/DropTargetContext.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/dnd/DropTargetAdapter.html)    [**NO FRAMES**](http://docs.google.com/DropTargetAdapter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#2s8eyo1) |

## **java.awt.dnd**

Class DropTargetAdapter

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

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

public abstract class **DropTargetAdapter**extends [Object](http://docs.google.com/java/lang/Object.html)implements [DropTargetListener](http://docs.google.com/java/awt/dnd/DropTargetListener.html)

An abstract adapter class for receiving drop target events. The methods in this class are empty. This class exists only as a convenience for creating listener objects.

Extend this class to create a DropTargetEvent listener and override the methods for the events of interest. (If you implement the DropTargetListener interface, you have to define all of the methods in it. This abstract class defines a null implementation for every method except drop(DropTargetDropEvent), so you only have to define methods for events you care about.) You must provide an implementation for at least drop(DropTargetDropEvent). This method cannot have a null implementation because its specification requires that you either accept or reject the drop, and, if accepted, indicate whether the drop was successful.

Create a listener object using the extended class and then register it with a DropTarget. When the drag enters, moves over, or exits the operable part of the drop site for that DropTarget, when the drop action changes, and when the drop occurs, the relevant method in the listener object is invoked, and the DropTargetEvent is passed to it.

The operable part of the drop site for the DropTarget is the part of the associated Component's geometry that is not obscured by an overlapping top-level window or by another Component higher in the Z-order that has an associated active DropTarget.

During the drag, the data associated with the current drag operation can be retrieved by calling getTransferable() on DropTargetDragEvent instances passed to the listener's methods.

Note that getTransferable() on the DropTargetDragEvent instance should only be called within the respective listener's method and all the necessary data should be retrieved from the returned Transferable before that method returns.

**Since:** 1.4 **See Also:**[DropTargetEvent](http://docs.google.com/java/awt/dnd/DropTargetEvent.html), [DropTargetListener](http://docs.google.com/java/awt/dnd/DropTargetListener.html)

| **Constructor Summary** | |
| --- | --- |
| [**DropTargetAdapter**](http://docs.google.com/java/awt/dnd/DropTargetAdapter.html#DropTargetAdapter())() |

| **Method Summary** | |
| --- | --- |
| void | [**dragEnter**](http://docs.google.com/java/awt/dnd/DropTargetAdapter.html#dragEnter(java.awt.dnd.DropTargetDragEvent))([DropTargetDragEvent](http://docs.google.com/java/awt/dnd/DropTargetDragEvent.html) dtde)            Called while a drag operation is ongoing, when the mouse pointer enters the operable part of the drop site for the DropTarget registered with this listener. |
| void | [**dragExit**](http://docs.google.com/java/awt/dnd/DropTargetAdapter.html#dragExit(java.awt.dnd.DropTargetEvent))([DropTargetEvent](http://docs.google.com/java/awt/dnd/DropTargetEvent.html) dte)            Called while a drag operation is ongoing, when the mouse pointer has exited the operable part of the drop site for the DropTarget registered with this listener. |
| void | [**dragOver**](http://docs.google.com/java/awt/dnd/DropTargetAdapter.html#dragOver(java.awt.dnd.DropTargetDragEvent))([DropTargetDragEvent](http://docs.google.com/java/awt/dnd/DropTargetDragEvent.html) dtde)            Called when a drag operation is ongoing, while the mouse pointer is still over the operable part of the drop site for the DropTarget registered with this listener. |
| void | [**dropActionChanged**](http://docs.google.com/java/awt/dnd/DropTargetAdapter.html#dropActionChanged(java.awt.dnd.DropTargetDragEvent))([DropTargetDragEvent](http://docs.google.com/java/awt/dnd/DropTargetDragEvent.html) dtde)            Called if the user has modified the current drop gesture. |

| **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)) |

| **Methods inherited from interface java.awt.dnd.**[**DropTargetListener**](http://docs.google.com/java/awt/dnd/DropTargetListener.html) |
| --- |
| [drop](http://docs.google.com/java/awt/dnd/DropTargetListener.html#drop(java.awt.dnd.DropTargetDropEvent)) |

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

### DropTargetAdapter

public **DropTargetAdapter**()

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

### dragEnter

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

Called while a drag operation is ongoing, when the mouse pointer enters the operable part of the drop site for the DropTarget registered with this listener.

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

### dragOver

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

Called when a drag operation is ongoing, while the mouse pointer is still over the operable part of the drop site for the DropTarget registered with this listener.

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

### dropActionChanged

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

Called if the user has modified the current drop gesture.

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

### dragExit

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

Called while a drag operation is ongoing, when the mouse pointer has exited the operable part of the drop site for the DropTarget registered with this listener.

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

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DropTargetAdapter.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/DropTarget.DropTargetAutoScroller.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/dnd/DropTargetContext.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/dnd/DropTargetAdapter.html)    [**NO FRAMES**](http://docs.google.com/DropTargetAdapter.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#2s8eyo1) |

[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).