| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DropTarget.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/DragSourceMotionListener.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/dnd/DropTarget.DropTargetAutoScroller.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/dnd/DropTarget.html)    [**NO FRAMES**](http://docs.google.com/DropTarget.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#lnxbz9) |

## **java.awt.dnd**

Class DropTarget

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

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

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

The DropTarget is associated with a Component when that Component wishes to accept drops during Drag and Drop operations.

Each DropTarget is associated with a FlavorMap. The default FlavorMap hereafter designates the FlavorMap returned by SystemFlavorMap.getDefaultFlavorMap().

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

| **Nested Class Summary** | |
| --- | --- |
| protected static class | [**DropTarget.DropTargetAutoScroller**](http://docs.google.com/java/awt/dnd/DropTarget.DropTargetAutoScroller.html)            this protected nested class implements autoscrolling |

| **Constructor Summary** | |
| --- | --- |
| [**DropTarget**](http://docs.google.com/java/awt/dnd/DropTarget.html#DropTarget())()            Creates a DropTarget. |
| [**DropTarget**](http://docs.google.com/java/awt/dnd/DropTarget.html#DropTarget(java.awt.Component,%20java.awt.dnd.DropTargetListener))([Component](http://docs.google.com/java/awt/Component.html) c, [DropTargetListener](http://docs.google.com/java/awt/dnd/DropTargetListener.html) dtl)            Creates a DropTarget given the Component to associate itself with, and the DropTargetListener to handle event processing. |
| [**DropTarget**](http://docs.google.com/java/awt/dnd/DropTarget.html#DropTarget(java.awt.Component,%20int,%20java.awt.dnd.DropTargetListener))([Component](http://docs.google.com/java/awt/Component.html) c, int ops, [DropTargetListener](http://docs.google.com/java/awt/dnd/DropTargetListener.html) dtl)            Creates a DropTarget given the Component to associate itself with, an int representing the default acceptable action(s) to support, and a DropTargetListener to handle event processing. |
| [**DropTarget**](http://docs.google.com/java/awt/dnd/DropTarget.html#DropTarget(java.awt.Component,%20int,%20java.awt.dnd.DropTargetListener,%20boolean))([Component](http://docs.google.com/java/awt/Component.html) c, int ops, [DropTargetListener](http://docs.google.com/java/awt/dnd/DropTargetListener.html) dtl, boolean act)            Creates a DropTarget given the Component to associate itself with, an int representing the default acceptable action(s) to support, a DropTargetListener to handle event processing, and a boolean indicating if the DropTarget is currently accepting drops. |
| [**DropTarget**](http://docs.google.com/java/awt/dnd/DropTarget.html#DropTarget(java.awt.Component,%20int,%20java.awt.dnd.DropTargetListener,%20boolean,%20java.awt.datatransfer.FlavorMap))([Component](http://docs.google.com/java/awt/Component.html) c, int ops, [DropTargetListener](http://docs.google.com/java/awt/dnd/DropTargetListener.html) dtl, boolean act, [FlavorMap](http://docs.google.com/java/awt/datatransfer/FlavorMap.html) fm)            Creates a new DropTarget given the Component to associate itself with, an int representing the default acceptable action(s) to support, a DropTargetListener to handle event processing, a boolean indicating if the DropTarget is currently accepting drops, and a FlavorMap to use (or null for the default FlavorMap). |

| **Method Summary** | |
| --- | --- |
| void | [**addDropTargetListener**](http://docs.google.com/java/awt/dnd/DropTarget.html#addDropTargetListener(java.awt.dnd.DropTargetListener))([DropTargetListener](http://docs.google.com/java/awt/dnd/DropTargetListener.html) dtl)            Adds a new DropTargetListener (UNICAST SOURCE). |
| void | [**addNotify**](http://docs.google.com/java/awt/dnd/DropTarget.html#addNotify(java.awt.peer.ComponentPeer))(java.awt.peer.ComponentPeer peer)            Notify the DropTarget that it has been associated with a Component This method is usually called from java.awt.Component.addNotify() of the Component associated with this DropTarget to notify the DropTarget that a ComponentPeer has been associated with that Component. |
| protected  void | [**clearAutoscroll**](http://docs.google.com/java/awt/dnd/DropTarget.html#clearAutoscroll())()            clear autoscrolling |
| protected  [DropTarget.DropTargetAutoScroller](http://docs.google.com/java/awt/dnd/DropTarget.DropTargetAutoScroller.html) | [**createDropTargetAutoScroller**](http://docs.google.com/java/awt/dnd/DropTarget.html#createDropTargetAutoScroller(java.awt.Component,%20java.awt.Point))([Component](http://docs.google.com/java/awt/Component.html) c, [Point](http://docs.google.com/java/awt/Point.html) p)            create an embedded autoscroller |
| protected  [DropTargetContext](http://docs.google.com/java/awt/dnd/DropTargetContext.html) | [**createDropTargetContext**](http://docs.google.com/java/awt/dnd/DropTarget.html#createDropTargetContext())()            Creates the DropTargetContext associated with this DropTarget. |
| void | [**dragEnter**](http://docs.google.com/java/awt/dnd/DropTarget.html#dragEnter(java.awt.dnd.DropTargetDragEvent))([DropTargetDragEvent](http://docs.google.com/java/awt/dnd/DropTargetDragEvent.html) dtde)            Calls dragEnter on the registered DropTargetListener and passes it the specified DropTargetDragEvent. |
| void | [**dragExit**](http://docs.google.com/java/awt/dnd/DropTarget.html#dragExit(java.awt.dnd.DropTargetEvent))([DropTargetEvent](http://docs.google.com/java/awt/dnd/DropTargetEvent.html) dte)            Calls dragExit on the registered DropTargetListener and passes it the specified DropTargetEvent. |
| void | [**dragOver**](http://docs.google.com/java/awt/dnd/DropTarget.html#dragOver(java.awt.dnd.DropTargetDragEvent))([DropTargetDragEvent](http://docs.google.com/java/awt/dnd/DropTargetDragEvent.html) dtde)            Calls dragOver on the registered DropTargetListener and passes it the specified DropTargetDragEvent. |
| void | [**drop**](http://docs.google.com/java/awt/dnd/DropTarget.html#drop(java.awt.dnd.DropTargetDropEvent))([DropTargetDropEvent](http://docs.google.com/java/awt/dnd/DropTargetDropEvent.html) dtde)            Calls drop on the registered DropTargetListener and passes it the specified DropTargetDropEvent if this DropTarget is active. |
| void | [**dropActionChanged**](http://docs.google.com/java/awt/dnd/DropTarget.html#dropActionChanged(java.awt.dnd.DropTargetDragEvent))([DropTargetDragEvent](http://docs.google.com/java/awt/dnd/DropTargetDragEvent.html) dtde)            Calls dropActionChanged on the registered DropTargetListener and passes it the specified DropTargetDragEvent. |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getComponent**](http://docs.google.com/java/awt/dnd/DropTarget.html#getComponent())()            Gets the Component associated with this DropTarget. |
| int | [**getDefaultActions**](http://docs.google.com/java/awt/dnd/DropTarget.html#getDefaultActions())()            Gets an int representing the current action(s) supported by this DropTarget. |
| [DropTargetContext](http://docs.google.com/java/awt/dnd/DropTargetContext.html) | [**getDropTargetContext**](http://docs.google.com/java/awt/dnd/DropTarget.html#getDropTargetContext())()            Gets the DropTargetContext associated with this DropTarget. |
| [FlavorMap](http://docs.google.com/java/awt/datatransfer/FlavorMap.html) | [**getFlavorMap**](http://docs.google.com/java/awt/dnd/DropTarget.html#getFlavorMap())()            Gets the FlavorMap associated with this DropTarget. |
| protected  void | [**initializeAutoscrolling**](http://docs.google.com/java/awt/dnd/DropTarget.html#initializeAutoscrolling(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) p)            initialize autoscrolling |
| boolean | [**isActive**](http://docs.google.com/java/awt/dnd/DropTarget.html#isActive())()            Reports whether or not this DropTarget is currently active (ready to accept drops). |
| void | [**removeDropTargetListener**](http://docs.google.com/java/awt/dnd/DropTarget.html#removeDropTargetListener(java.awt.dnd.DropTargetListener))([DropTargetListener](http://docs.google.com/java/awt/dnd/DropTargetListener.html) dtl)            Removes the current DropTargetListener (UNICAST SOURCE). |
| void | [**removeNotify**](http://docs.google.com/java/awt/dnd/DropTarget.html#removeNotify(java.awt.peer.ComponentPeer))(java.awt.peer.ComponentPeer peer)            Notify the DropTarget that it has been disassociated from a Component This method is usually called from java.awt.Component.removeNotify() of the Component associated with this DropTarget to notify the DropTarget that a ComponentPeer has been disassociated with that Component. |
| void | [**setActive**](http://docs.google.com/java/awt/dnd/DropTarget.html#setActive(boolean))(boolean isActive)            Sets the DropTarget active if true, inactive if false. |
| void | [**setComponent**](http://docs.google.com/java/awt/dnd/DropTarget.html#setComponent(java.awt.Component))([Component](http://docs.google.com/java/awt/Component.html) c)            Note: this interface is required to permit the safe association of a DropTarget with a Component in one of two ways, either: component.setDropTarget(droptarget); or droptarget.setComponent(component); |
| void | [**setDefaultActions**](http://docs.google.com/java/awt/dnd/DropTarget.html#setDefaultActions(int))(int ops)            Sets the default acceptable actions for this DropTarget |
| void | [**setFlavorMap**](http://docs.google.com/java/awt/dnd/DropTarget.html#setFlavorMap(java.awt.datatransfer.FlavorMap))([FlavorMap](http://docs.google.com/java/awt/datatransfer/FlavorMap.html) fm)            Sets the FlavorMap associated with this DropTarget. |
| protected  void | [**updateAutoscroll**](http://docs.google.com/java/awt/dnd/DropTarget.html#updateAutoscroll(java.awt.Point))([Point](http://docs.google.com/java/awt/Point.html) dragCursorLocn)            update autoscrolling with current cursor locn |

| **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)) |

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

### DropTarget

public **DropTarget**([Component](http://docs.google.com/java/awt/Component.html) c,  
 int ops,  
 [DropTargetListener](http://docs.google.com/java/awt/dnd/DropTargetListener.html) dtl,  
 boolean act,  
 [FlavorMap](http://docs.google.com/java/awt/datatransfer/FlavorMap.html) fm)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Creates a new DropTarget given the Component to associate itself with, an int representing the default acceptable action(s) to support, a DropTargetListener to handle event processing, a boolean indicating if the DropTarget is currently accepting drops, and a FlavorMap to use (or null for the default FlavorMap).

The Component will receive drops only if it is enabled.

**Parameters:**c - The Component with which this DropTarget is associatedops - The default acceptable actions for this DropTargetdtl - The DropTargetListener for this DropTargetact - Is the DropTarget accepting drops.fm - The FlavorMap to use, or null for the default FlavorMap **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### DropTarget

public **DropTarget**([Component](http://docs.google.com/java/awt/Component.html) c,  
 int ops,  
 [DropTargetListener](http://docs.google.com/java/awt/dnd/DropTargetListener.html) dtl,  
 boolean act)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Creates a DropTarget given the Component to associate itself with, an int representing the default acceptable action(s) to support, a DropTargetListener to handle event processing, and a boolean indicating if the DropTarget is currently accepting drops.

The Component will receive drops only if it is enabled.

**Parameters:**c - The Component with which this DropTarget is associatedops - The default acceptable actions for this DropTargetdtl - The DropTargetListener for this DropTargetact - Is the DropTarget accepting drops. **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### DropTarget

public **DropTarget**()  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Creates a DropTarget.

**Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### DropTarget

public **DropTarget**([Component](http://docs.google.com/java/awt/Component.html) c,  
 [DropTargetListener](http://docs.google.com/java/awt/dnd/DropTargetListener.html) dtl)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Creates a DropTarget given the Component to associate itself with, and the DropTargetListener to handle event processing.

The Component will receive drops only if it is enabled.

**Parameters:**c - The Component with which this DropTarget is associateddtl - The DropTargetListener for this DropTarget **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

### DropTarget

public **DropTarget**([Component](http://docs.google.com/java/awt/Component.html) c,  
 int ops,  
 [DropTargetListener](http://docs.google.com/java/awt/dnd/DropTargetListener.html) dtl)  
 throws [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html)

Creates a DropTarget given the Component to associate itself with, an int representing the default acceptable action(s) to support, and a DropTargetListener to handle event processing.

The Component will receive drops only if it is enabled.

**Parameters:**c - The Component with which this DropTarget is associatedops - The default acceptable actions for this DropTargetdtl - The DropTargetListener for this DropTarget **Throws:** [HeadlessException](http://docs.google.com/java/awt/HeadlessException.html) - if GraphicsEnvironment.isHeadless() returns true**See Also:**[GraphicsEnvironment.isHeadless()](http://docs.google.com/java/awt/GraphicsEnvironment.html#isHeadless())

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

### setComponent

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

Note: this interface is required to permit the safe association of a DropTarget with a Component in one of two ways, either: component.setDropTarget(droptarget); or droptarget.setComponent(component);

The Component will receive drops only if it is enabled.

**Parameters:**c - The new Component this DropTarget is to be associated with.

### getComponent

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

Gets the Component associated with this DropTarget.

**Returns:**the current Component

### setDefaultActions

public void **setDefaultActions**(int ops)

Sets the default acceptable actions for this DropTarget

**Parameters:**ops - the default actions

**See Also:**[DnDConstants](http://docs.google.com/java/awt/dnd/DnDConstants.html)

### getDefaultActions

public int **getDefaultActions**()

Gets an int representing the current action(s) supported by this DropTarget.

**Returns:**the current default actions

### setActive

public void **setActive**(boolean isActive)

Sets the DropTarget active if true, inactive if false.

**Parameters:**isActive - sets the DropTarget (in)active.

### isActive

public boolean **isActive**()

Reports whether or not this DropTarget is currently active (ready to accept drops).

**Returns:**true if active, false if not

### addDropTargetListener

public void **addDropTargetListener**([DropTargetListener](http://docs.google.com/java/awt/dnd/DropTargetListener.html) dtl)  
 throws [TooManyListenersException](http://docs.google.com/java/util/TooManyListenersException.html)

Adds a new DropTargetListener (UNICAST SOURCE).

**Parameters:**dtl - The new DropTargetListener

**Throws:** TooManyListenersException - if a DropTargetListener is already added to this DropTarget. [TooManyListenersException](http://docs.google.com/java/util/TooManyListenersException.html)

### removeDropTargetListener

public void **removeDropTargetListener**([DropTargetListener](http://docs.google.com/java/awt/dnd/DropTargetListener.html) dtl)

Removes the current DropTargetListener (UNICAST SOURCE).

**Parameters:**dtl - the DropTargetListener to deregister.

### dragEnter

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

Calls dragEnter on the registered DropTargetListener and passes it the specified DropTargetDragEvent. Has no effect if this DropTarget is not active.

**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 **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if this DropTarget is active and dtde is null**See Also:**[isActive()](http://docs.google.com/java/awt/dnd/DropTarget.html#isActive())

### dragOver

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

Calls dragOver on the registered DropTargetListener and passes it the specified DropTargetDragEvent. Has no effect if this DropTarget is not active.

**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 **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if this DropTarget is active and dtde is null**See Also:**[isActive()](http://docs.google.com/java/awt/dnd/DropTarget.html#isActive())

### dropActionChanged

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

Calls dropActionChanged on the registered DropTargetListener and passes it the specified DropTargetDragEvent. Has no effect if this DropTarget is not active.

**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 **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if this DropTarget is active and dtde is null**See Also:**[isActive()](http://docs.google.com/java/awt/dnd/DropTarget.html#isActive())

### dragExit

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

Calls dragExit on the registered DropTargetListener and passes it the specified DropTargetEvent. Has no effect if this DropTarget is not active.

This method itself does not throw any exception for null parameter but for exceptions thrown by the respective method of the 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**See Also:**[isActive()](http://docs.google.com/java/awt/dnd/DropTarget.html#isActive())

### drop

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

Calls drop on the registered DropTargetListener and passes it the specified DropTargetDropEvent if this DropTarget is active.

**Specified by:**[drop](http://docs.google.com/java/awt/dnd/DropTargetListener.html#drop(java.awt.dnd.DropTargetDropEvent)) in interface [DropTargetListener](http://docs.google.com/java/awt/dnd/DropTargetListener.html) **Parameters:**dtde - the DropTargetDropEvent **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if dtde is null and at least one of the following is true: this DropTarget is not active, or there is no a DropTargetListener registered.**See Also:**[isActive()](http://docs.google.com/java/awt/dnd/DropTarget.html#isActive())

### getFlavorMap

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

Gets the FlavorMap associated with this DropTarget. If no FlavorMap has been set for this DropTarget, it is associated with the default FlavorMap.

**Returns:**the FlavorMap for this DropTarget

### setFlavorMap

public void **setFlavorMap**([FlavorMap](http://docs.google.com/java/awt/datatransfer/FlavorMap.html) fm)

Sets the FlavorMap associated with this DropTarget.

**Parameters:**fm - the new FlavorMap, or null to associate the default FlavorMap with this DropTarget.

### addNotify

public void **addNotify**(java.awt.peer.ComponentPeer peer)

Notify the DropTarget that it has been associated with a Component This method is usually called from java.awt.Component.addNotify() of the Component associated with this DropTarget to notify the DropTarget that a ComponentPeer has been associated with that Component. Calling this method, other than to notify this DropTarget of the association of the ComponentPeer with the Component may result in a malfunction of the DnD system.

**Parameters:**peer - The Peer of the Component we are associated with!

### removeNotify

public void **removeNotify**(java.awt.peer.ComponentPeer peer)

Notify the DropTarget that it has been disassociated from a Component This method is usually called from java.awt.Component.removeNotify() of the Component associated with this DropTarget to notify the DropTarget that a ComponentPeer has been disassociated with that Component. Calling this method, other than to notify this DropTarget of the disassociation of the ComponentPeer from the Component may result in a malfunction of the DnD system.

**Parameters:**peer - The Peer of the Component we are being disassociated from!

### getDropTargetContext

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

Gets the DropTargetContext associated with this DropTarget.

**Returns:**the DropTargetContext associated with this DropTarget.

### createDropTargetContext

protected [DropTargetContext](http://docs.google.com/java/awt/dnd/DropTargetContext.html) **createDropTargetContext**()

Creates the DropTargetContext associated with this DropTarget. Subclasses may override this method to instantiate their own DropTargetContext subclass. This call is typically \*only\* called by the platform's DropTargetContextPeer as a drag operation encounters this DropTarget. Accessing the Context while no Drag is current has undefined results.

### createDropTargetAutoScroller

protected [DropTarget.DropTargetAutoScroller](http://docs.google.com/java/awt/dnd/DropTarget.DropTargetAutoScroller.html) **createDropTargetAutoScroller**([Component](http://docs.google.com/java/awt/Component.html) c,  
 [Point](http://docs.google.com/java/awt/Point.html) p)

create an embedded autoscroller

**Parameters:**c - the Componentp - the Point

### initializeAutoscrolling

protected void **initializeAutoscrolling**([Point](http://docs.google.com/java/awt/Point.html) p)

initialize autoscrolling

**Parameters:**p - the Point

### updateAutoscroll

protected void **updateAutoscroll**([Point](http://docs.google.com/java/awt/Point.html) dragCursorLocn)

update autoscrolling with current cursor locn

**Parameters:**dragCursorLocn - the Point

### clearAutoscroll

protected void **clearAutoscroll**()

clear autoscrolling

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/DropTarget.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/DragSourceMotionListener.html)   [**NEXT CLASS**](http://docs.google.com/java/awt/dnd/DropTarget.DropTargetAutoScroller.html) | [**FRAMES**](http://docs.google.com/index.html?java/awt/dnd/DropTarget.html)    [**NO FRAMES**](http://docs.google.com/DropTarget.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: [NESTED](#3znysh7) | FIELD | [CONSTR](#2et92p0) | [METHOD](#tyjcwt) | DETAIL: FIELD | [CONSTR](#1t3h5sf) | [METHOD](#lnxbz9) |

[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).