| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TransferHandler.TransferSupport.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/javax/swing/TransferHandler.DropLocation.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/UIDefaults.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/TransferHandler.TransferSupport.html)    [**NO FRAMES**](http://docs.google.com/TransferHandler.TransferSupport.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

## **javax.swing**

Class TransferHandler.TransferSupport

[java.lang.Object](http://docs.google.com/java/lang/Object.html)  
 **javax.swing.TransferHandler.TransferSupport**

**Enclosing class:**[TransferHandler](http://docs.google.com/javax/swing/TransferHandler.html)

public static final class **TransferHandler.TransferSupport**extends [Object](http://docs.google.com/java/lang/Object.html)

This class encapsulates all relevant details of a clipboard or drag and drop transfer, and also allows for customizing aspects of the drag and drop experience.

The main purpose of this class is to provide the information needed by a developer to determine the suitability of a transfer or to import the data contained within. But it also doubles as a controller for customizing properties during drag and drop, such as whether or not to show the drop location, and which drop action to use.

Developers typically need not create instances of this class. Instead, they are something provided by the DnD implementation to certain methods in TransferHandler.

**Since:** 1.6 **See Also:**[TransferHandler.canImport(TransferHandler.TransferSupport)](http://docs.google.com/javax/swing/TransferHandler.html#canImport(javax.swing.TransferHandler.TransferSupport)), [TransferHandler.importData(TransferHandler.TransferSupport)](http://docs.google.com/javax/swing/TransferHandler.html#importData(javax.swing.TransferHandler.TransferSupport))

| **Constructor Summary** | |
| --- | --- |
| [**TransferHandler.TransferSupport**](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#TransferHandler.TransferSupport(java.awt.Component,%20java.awt.datatransfer.Transferable))([Component](http://docs.google.com/java/awt/Component.html) component, [Transferable](http://docs.google.com/java/awt/datatransfer/Transferable.html) transferable)            Create a TransferSupport with isDrop() false for the given component and Transferable. |

| **Method Summary** | |
| --- | --- |
| [Component](http://docs.google.com/java/awt/Component.html) | [**getComponent**](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#getComponent())()            Returns the target component of this transfer. |
| [DataFlavor](http://docs.google.com/java/awt/datatransfer/DataFlavor.html)[] | [**getDataFlavors**](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#getDataFlavors())()            Returns the data flavors for this transfer. |
| int | [**getDropAction**](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#getDropAction())()            Returns the action chosen for the drop, when this TransferSupport represents a drop. |
| [TransferHandler.DropLocation](http://docs.google.com/javax/swing/TransferHandler.DropLocation.html) | [**getDropLocation**](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#getDropLocation())()            Returns the current (non-null) drop location for the component, when this TransferSupport represents a drop. |
| int | [**getSourceDropActions**](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#getSourceDropActions())()            Returns the drag source's supported drop actions, when this TransferSupport represents a drop. |
| [Transferable](http://docs.google.com/java/awt/datatransfer/Transferable.html) | [**getTransferable**](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#getTransferable())()            Returns the Transferable associated with this transfer. |
| int | [**getUserDropAction**](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#getUserDropAction())()            Returns the user drop action for the drop, when this TransferSupport represents a drop. |
| boolean | [**isDataFlavorSupported**](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#isDataFlavorSupported(java.awt.datatransfer.DataFlavor))([DataFlavor](http://docs.google.com/java/awt/datatransfer/DataFlavor.html) df)            Returns whether or not the given data flavor is supported. |
| boolean | [**isDrop**](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#isDrop())()            Returns whether or not this TransferSupport represents a drop operation. |
| void | [**setDropAction**](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#setDropAction(int))(int dropAction)            Sets the drop action for the transfer - which must represent a drop - to the given action, instead of the default user drop action. |
| void | [**setShowDropLocation**](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#setShowDropLocation(boolean))(boolean showDropLocation)            Sets whether or not the drop location should be visually indicated for the transfer - which must represent a drop. |

| **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** |
| --- |

### TransferHandler.TransferSupport

public **TransferHandler.TransferSupport**([Component](http://docs.google.com/java/awt/Component.html) component,  
 [Transferable](http://docs.google.com/java/awt/datatransfer/Transferable.html) transferable)

Create a TransferSupport with isDrop() false for the given component and Transferable.

**Parameters:**component - the target componenttransferable - the transferable **Throws:** [NullPointerException](http://docs.google.com/java/lang/NullPointerException.html) - if either parameter is null

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

### isDrop

public boolean **isDrop**()

Returns whether or not this TransferSupport represents a drop operation.

**Returns:**true if this is a drop operation, false otherwise.

### getComponent

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

Returns the target component of this transfer.

**Returns:**the target component

### getDropLocation

public [TransferHandler.DropLocation](http://docs.google.com/javax/swing/TransferHandler.DropLocation.html) **getDropLocation**()

Returns the current (non-null) drop location for the component, when this TransferSupport represents a drop.

Note: For components with built-in drop support, this location will be a subclass of DropLocation of the same type returned by that component's getDropLocation method.

This method is only for use with drag and drop transfers. Calling it when isDrop() is false results in an IllegalStateException.

**Returns:**the drop location **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this is not a drop**See Also:**[isDrop](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#isDrop)

### setShowDropLocation

public void **setShowDropLocation**(boolean showDropLocation)

Sets whether or not the drop location should be visually indicated for the transfer - which must represent a drop. This is applicable to those components that automatically show the drop location when appropriate during a drag and drop operation). By default, the drop location is shown only when the TransferHandler has said it can accept the import represented by this TransferSupport. With this method you can force the drop location to always be shown, or always not be shown.

This method is only for use with drag and drop transfers. Calling it when isDrop() is false results in an IllegalStateException.

**Parameters:**showDropLocation - whether or not to indicate the drop location **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this is not a drop**See Also:**[isDrop](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#isDrop)

### setDropAction

public void **setDropAction**(int dropAction)

Sets the drop action for the transfer - which must represent a drop - to the given action, instead of the default user drop action. The action must be supported by the source's drop actions, and must be one of COPY, MOVE or LINK.

This method is only for use with drag and drop transfers. Calling it when isDrop() is false results in an IllegalStateException.

**Parameters:**dropAction - the drop action **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this is not a drop [IllegalArgumentException](http://docs.google.com/java/lang/IllegalArgumentException.html) - if an invalid action is specified**See Also:**[getDropAction()](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#getDropAction()), [getUserDropAction()](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#getUserDropAction()), [getSourceDropActions()](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#getSourceDropActions()), [isDrop](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#isDrop)

### getDropAction

public int **getDropAction**()

Returns the action chosen for the drop, when this TransferSupport represents a drop.

Unless explicitly chosen by way of setDropAction, this returns the user drop action provided by getUserDropAction.

You may wish to query this in TransferHandler's importData method to customize processing based on the action.

This method is only for use with drag and drop transfers. Calling it when isDrop() is false results in an IllegalStateException.

**Returns:**the action chosen for the drop **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this is not a drop**See Also:**[setDropAction(int)](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#setDropAction(int)), [getUserDropAction()](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#getUserDropAction()), [isDrop](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#isDrop)

### getUserDropAction

public int **getUserDropAction**()

Returns the user drop action for the drop, when this TransferSupport represents a drop.

The user drop action is chosen for a drop as described in the documentation for [DropTargetDragEvent](http://docs.google.com/java/awt/dnd/DropTargetDragEvent.html) and [DropTargetDropEvent](http://docs.google.com/java/awt/dnd/DropTargetDropEvent.html). A different action may be chosen as the drop action by way of the setDropAction method.

You may wish to query this in TransferHandler's canImport method when determining the suitability of a drop or when deciding on a drop action to explicitly choose.

This method is only for use with drag and drop transfers. Calling it when isDrop() is false results in an IllegalStateException.

**Returns:**the user drop action **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this is not a drop**See Also:**[setDropAction(int)](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#setDropAction(int)), [getDropAction()](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#getDropAction()), [isDrop](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#isDrop)

### getSourceDropActions

public int **getSourceDropActions**()

Returns the drag source's supported drop actions, when this TransferSupport represents a drop.

The source actions represent the set of actions supported by the source of this transfer, and are represented as some bitwise-OR combination of COPY, MOVE and LINK. You may wish to query this in TransferHandler's canImport method when determining the suitability of a drop or when deciding on a drop action to explicitly choose. To determine if a particular action is supported by the source, bitwise-AND the action with the source drop actions, and then compare the result against the original action. For example:

boolean copySupported = (COPY & getSourceDropActions()) == COPY;

This method is only for use with drag and drop transfers. Calling it when isDrop() is false results in an IllegalStateException.

**Returns:**the drag source's supported drop actions **Throws:** [IllegalStateException](http://docs.google.com/java/lang/IllegalStateException.html) - if this is not a drop**See Also:**[isDrop](http://docs.google.com/javax/swing/TransferHandler.TransferSupport.html#isDrop)

### getDataFlavors

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

Returns the data flavors for this transfer.

**Returns:**the data flavors for this transfer

### isDataFlavorSupported

public boolean **isDataFlavorSupported**([DataFlavor](http://docs.google.com/java/awt/datatransfer/DataFlavor.html) df)

Returns whether or not the given data flavor is supported.

**Parameters:**df - the DataFlavor to test **Returns:**whether or not the given flavor is supported.

### getTransferable

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

Returns the Transferable associated with this transfer.

Note: Unless it is necessary to fetch the Transferable directly, use one of the other methods on this class to inquire about the transfer. This may perform better than fetching the Transferable and asking it directly.

**Returns:**the Transferable associated with this transfer

| | [**Overview**](http://docs.google.com/overview-summary.html) | [**Package**](http://docs.google.com/package-summary.html) | **Class** | [**Use**](http://docs.google.com/class-use/TransferHandler.TransferSupport.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/javax/swing/TransferHandler.DropLocation.html)   [**NEXT CLASS**](http://docs.google.com/javax/swing/UIDefaults.html) | [**FRAMES**](http://docs.google.com/index.html?javax/swing/TransferHandler.TransferSupport.html)    [**NO FRAMES**](http://docs.google.com/TransferHandler.TransferSupport.html)     [**All Classes**](http://docs.google.com/allclasses-noframe.html) |
| SUMMARY: NESTED | FIELD | [CONSTR](#3znysh7) | [METHOD](#2et92p0) | DETAIL: FIELD | [CONSTR](#3dy6vkm) | [METHOD](#4d34og8) |

[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).