MiscDragView

Inherits From: View : Responder : Object

Declared In: misckit/MiscDragView.h

Protocols: NXDraggingSource, NXDraggingDestination

Class Description

MiscDragView is an abstract class that implements the methods needed for both source and destination dragging, includes methods to set what types of dragging are allowed, along with the addition of a delegate that is notified when a drag session (both source and destination) has begun or ended. The dragging options (see below for a list) can be set and changed at any time (unlike, for example, view's **setFlipped:** which should be done only when the view is created) without any ill effects.

The minimum needed to create a subclass that allows simple drag and drop (source and destination) is as follows:

±Override **initFrame:** and register the pasteboard types you will accept (needed for destination dragging only).
±Override **setupSourceForDragging** in order to set the dragImage and put the relevant data on the pasteboard (needed for source dragging only).
±Override **performDragOperation:** to take the data from the pasteboard, when a successful dragging session

has occurred (needed for destination dragging only).

Unless you would like the most simple of drag views, you will find it necessary to override other methods, mostly the ones that are in the two NXDragging protocols. Though I have them listed in the MiscDragView documentation, I will only list any changes I made to make the dragging options and delegation operational. For more information refer to /NextLibrary/Documentation/ NextDev/GeneralRef/02_ApplicationKit/Protocols/NXDraggingDestination.rtf and /NextLibrary/Documentation/ NextDev/GeneralRef/02_ApplicationKit/Protocols/NXDraggingSource.rtf.

Instance Variables

BOOL **allowSourceDragging**;
BOOL **allowDestinationDragging**;
BOOL **acceptForeignDrag**;
BOOL **acceptLocalDrag**;
BOOL **acceptSelfDrag**;
BOOL **retainData**;

id **delegate**;
id **dragImage**;

allowSourceDragging	YES if we are allowed to initiate a dragging session.
allowDestinationDragging	YES if we are allowed to be the destination of a dragging session.
acceptForeignDrag	YES if we are allowed to be the destination of nonlocal dragging sessions.
acceptLocalDrag	YES if we are allowed to be the destination of local drag sessions

acceptSelfDrag	YES if we are allowed to be both the source and the destination in a single drag.
retainData	YES if the data is not destroyed after it is dragged out of the view.
delegate	The object that accepts notification messages.
dragImage	The image that is displayed during the dragging.

Method Types

Initializing a MiscDragView	± initFrame: ± awake
Dragging options	± setAllowSourceDragging: ± allowSourceDragging ± setAllowDestinationDragging: ± allowDestinationDragging ± setAcceptForeignDrag: ± acceptForeignDrag ± setAcceptLocalDrag: ± acceptLocalDrag ± setAcceptSelfDrag: ± acceptSelfDrag ± setRetainData: ± retainData
Source dragging methods	± mouseDown:

	± setupForSourceDrag ± calculateDragPoint:: ± draggingSourceOperationMaskForLocal ± draggedImage: beganAt: ± draggedImage: endedAt: deposited:
Destination dragging methods	± draggingEntered: ± draggingUpdated: ± draggingExited: ± prepareForDragOperation: ± performDragOperation: ± concludeDragOperation:
Delegate methods	± delegate ± setDelegate: ± sourceDragInitiated: ± sourceDragFinished: ± destinationDragInitiated: ± destinationDragFinished
Archiving	± read: ± write:

Instance Methods

acceptForeignDrag

- (BOOL)**acceptForeignDrag**

Returns YES if the view will accept a drag that did not initiate from within the same application.

See also: **-setAcceptForeignDrag**

acceptLocalDrag
- (BOOL)**acceptLocalDrag**

Returns YES if the view will accept a drag that initiated from within the same application.

See also: **-setAcceptLocalDrag**

acceptSelfDrag
- (BOOL)**acceptLocalDrag**

Returns YES if the view will accept a drag that initiated from itself.

See also: **-setAcceptSelfDrag**

allowDestinationDragging
- (BOOL)**allowDestinationDragging**

Returns YES if the view is allowed to be the destination of a dragging session.

See also: **-setAllowDestinationDragging**

allowSourceDragging
- (BOOL)**allowSourceDragging**

Returns YES if the view is allowed to initiate a dragging session.

See also: **-setAllowSourceDragging**

awake

- **awake**

Used when the object is unarchived to initialize the variables and register for dragging. Returns **self**.

See also: **-initFrame**

calculateDragPoint: andOffset:

- **calculateDragPoint:**(NXPoint *)*dragPoint* **andOffset:** (NXPoint *)*offset*

This method does nothing in MiscDragView, but is there for subclasses that would like more control of where the drag image first appears in relation to the mouse. The *dragPoint* is the NXImage's location given in the view's coordinate system. The *offset* is the current mouse location relative to the mouse down location that started the drag. So far I have just found that adjusting the *dragPoint* can be useful as to center the dragged image relative to the cursor. Returns **self.**

concludeDragOperation:

- **concludeDragOperation:** *sender*

This method is part of the NXDraggingSource protocol. If you subclass this method, make sure to call [super **concludeDragOperation:** sender] because this method sends the **sourceDragFinished:** message to the delegate. Returns **self**.

See also: **-prepareForDragOperation, -performDragOperation**

delegate

- **delegate**

Returns the current delegate, and *nil* if no delegate is currently set.

See also: **-setDelegate, -destinationDragInitiated, -destinationDragFinished, -sourceDragInitiated, -sourceDragFinished**

destinationDragInitiated:
- destinationDragInitiated: *view*

This message will be sent to the delegate, if it can respond, when the **draggingEntered:** method is called. Returns **self**.

See also: **-destinationDragFinished, -draggingEntered**

destinationDragFinished:
- destinationDragFinished: (BOOL)*success*

This message is sent to the delegate from the **concludeDragOperation:** method, with *success* set to YES (since the destination drag was successful). On the other hand, this message can also be sent from the **draggingExited:** method with *success* set to NO (since the destination drag was not a success).

See also: **-destinationDragInitiated, -concludeDragOperation**

draggedImage: beganAt:
- draggedImage: (NXImage *)*image* **beganAt:** (NXPoint *)*screenPoint*

This method is part of the NXDraggingSource protocol.

See also: **-draggedImage: endedAt:**

draggedImage: endedAt:
- draggedImage: (NXImage *)*image* **endedAt:** (NXPoint *)*screenPoint*

This method is part of the NXDraggingSource protocol. If you override this method in a subclass, make sure to call the super class's method so the **sourceDragFinished:** message is sent to the delegate (if it reponds to it). Returns **self**.

See also: **-draggedImage: beganAt:**

draggingEntered:

-(NXDragOperation)**draggingEntered:** *sender*

This method is part of the NXDraggingDestination protocol. Since this method contains most all the testing for types of destination dragging that are allowed, if you plan to override it, do so in the following way:

```
(NXDragOperation)draggingEntered: sender
{
   ( your code here )
   return [super draggingEntered: sender];
 }
```

See also: **-draggingUpdated, -draggingExited**

draggingExited:

-draggingExited: *sender*

This method is part of the NXDraggingDestination protocol. This method passes the message **destinationDragFinished:** NO to the delegate, since the dragging image exited the view, and the dragging did end (as far as the destination view is concerned). If you override this method later on, make to include a call to the super's method so the delegate method will be sent. Returns **self**.

See also: **-draggingUpdated, -draggingEntered**

draggingSourceOperationMaskForLocal:

- (NXDragOperation)**draggingSourceOperationMaskForLocal:** (BOOL)*flag*

This method is part of the NXDraggingSource protocol. The default value returned, as determined by this view is NXDragOperationAll.

draggingUpdated:

-(NXDragOperation)**draggingUpdated:** *sender*

This method is part of the NXDraggingDestination protocol. By default it returns the same value that **draggingEntered:** returned when the view was first entered by the dragged image.

See also: **-draggingExited, -draggingEntered**

initFrame:

-initFrame: (const NXRect *)*frameRect*

First calls superclass's initFrame, then initializes all the dragging options (all YES by default). Returns **self**.

See also: **-awake**

mouseDown:

- mouseDown: (NXEvent *)*theEvent*

mouseDown is overridden in order to begin a drag session. First, if source dragging is allowed, **setupForSourceDrag** is called to put the data on the pasteboard and choose an image for dragging. If **setupForSourceDrag** returns YES, **calculateDragPoint: andOffset:** is called to allow each subclass some finer control on where the image appears when a drag begins. As long as the drag image is not **nil, sourceDragInitiated:** is sent to the delegate, and the source dragging begins. Be careful when subclassing this method (I suppose you do have the source code though).

performDragOperation:

- (BOOL)**performDragOperation:** *sender*

This method is part of the NXDraggingDestination protocol. The default (for this class) is to return YES.

See also: **-prepareForDragOperation, -concludeDragOperation**

prepareForDragOperation:

- (BOOL)**prepareForDragOperation:** *sender*

This method is part of the NXDraggingDestination protocol. The default (for this class) is to return YES.

See also: **-performDragOperation, -concludeDragOperation**

read:

- **read:** (NXTypedStream *)*stream*

Archives a MiscDragView object. Returns **self**.

See also: **-write**

retainData

- (BOOL)**retainData**

Returns YES if the view keeps the data after a source drag has been completed.

See also: **-setRetainData**

setAcceptForeignDrag

- **setAcceptForeignDrag:** (BOOL)*aBool*

Sets whether to accept a drag that was not initiated from within the current application. Returns **self**.

See also: **-acceptForeignDrag**

setAcceptLocalDrag
- setAcceptLocalDrag: (BOOL)*aBool*

Sets whether to accept a drag that was initiated from within the current application. Returns **self**.

See also: **-acceptLocalDrag**

setAcceptSelfDrag
- setAcceptSelfDrag: (BOOL)*aBool*

Sets whether to accept a drag that was initiated from within itself. Returns **self**.

See also: **-acceptSelfDrag**

setAllowDestinationDragging
- setAllowDestinationDragging: (BOOL)*aBool*

Sets whether the view is allowed to be the destination of a dragging session. If set to NO, the view will not accept any dragged images. Returns **self.**

See also: **-allowDestinationDragging**

setAllowSourceDragging
- setAllowSourceDragging: (BOOL)*aBool*

Sets whether the view is allowed to initiate a dragging session. Returns **self.**

See also: **-allowSourceDragging**

setDelegate

- **setDelegate:** *theDelegate*

Sets *theDelegate* to be the object to receive the dragging notification messages.

See also: **-delegate, -destinationDragInitiated, -destinationDragFinished, -sourceDragInitiated, -sourceDragFinished**

setRetainData:

- **setRetainData:** (BOOL)*aBool*

Can be used to determine whether the view is will keep the data available after a source drag has been initiated. Within this class it is only used to determine whether the dragged image should slide back to the source view if the drag is not successful. Returns **self**.

See also: **-retainData**

setupForSourceDrag:

- (BOOL)**setupForSourceDrag**

This method does nothing in MiscDragView, but is meant to be overridden in subclasses. The overridden method should take care of putting it's data on the pasteboard and choosing the image to be dragged. This method is called from within **mouseDown:** just before the **dragImage::::** method is called. You should return YES if the data was put on the pasteboard successfully. If you return a NO, the drag session will not start. Returns **self**.

See also: **-mouseDown**

sourceDragFinished:

- **sourceDragFinished:** (BOOL)*success*

This method sends a message of the same name to the delegate if it can respond. It is called from **draggedImage: endedAt: deposited:** and returns the value from **deposited:** which is a YES if the dragging session was successful (the data was deposited in another view), or NO if the dragging session was stopped (a mouse up) before a view accepted it. Returns **self**.

See also: **-sourceDragInitiated**

sourceDragInitiated:
- sourceDragInitiated: *view*

This method sends a message of the same name to the delegate if it can respond. It is called from **mouseDown:** just before the dragging actually starts taking place, but after anything could possibly stop it (for instance returning a NO from **setupForSourceDrag**). Returns **self**.

See also: **-sourceDragFinished**

write:
-write: (NXTypedStream *)*stream*

Write a MiscDragView object and it's current state to an archive.

See also: **-read**