

```

/*****
CATEGORY:          MiscDragView (Delegate)
* Copyright (C) 1995 Robert Todd Thomas
* Use is governed by the MiscKit license

    See the header for more information on this category.
*****/

```

```

#import <appkit/appkit.h>
#import "MiscDragView-BackwardCompat.h"
#import "MiscDragView-Delegate.h"

```

**@implementation MiscDragView (Delegate)**

```

- delegate
{
    return delegate;
}

```

```
}
```

```
// The delegate will only be sent messages that it can respond to.
```

**- setDelegate: (id)theDelegate**

```
{
    delegate = theDelegate;
    if ([delegate respondsToSelector:@selector(image)])
        [self setImage:[delegate image]];
    [self initDragTypes];
    return self;
}
```

```
// Sent from "mouseDown" just before the dragging starts to take place
```

**- \_sourceDragInitiated: sender**

```
{  
    // here for backward compatibility, please use new delegate  
    // methods  
    if ([delegate respondsToSelector: @selector(sourceDragInitiated:)])  
        [delegate sourceDragInitiated: sender];  
  
    // newly renamed delegate methods  
    if ([delegate respondsToSelector: @selector(didInitiateSourceDrag:)])  
        [delegate didInitiateSourceDrag: sender];  
  
    return self;  
}
```

// Sent when "draggedImage: endedAt: screenPoint: deposited:" is given

```
// control and success is whether the image was deposited in another  
// view somewhere.
```

### **-\_sourceDragFinished: (BOOL)successful**

```
{  
    // here for backward compatibility, please use new delegate  
    // methods  
    if ([delegate respondsToSelector: @selector(sourceDragFinished:)])  
        [delegate sourceDragFinished: successful];  
  
    // new delegate method  
    if ([delegate respondsToSelector: @selector(didFinishSourceDrag:)])  
        [delegate didFinishSourceDrag: successful];  
  
    return self;  
}
```

```
// Sent when "draggingEntered:" is given control.
```

```
-_destinationDragInitiated: sender
```

```
{  
    // here for backward compatibility, please use new delegate  
    // methods  
    if ([delegate respondsToSelector: @selector(destinationDragInitiated:)])  
        [delegate destinationDragInitiated: sender];  
  
    // new delegate method  
    if ([delegate respondsToSelector: @selector(didInitiateDestinationDrag:)])  
        [delegate didInitiateDestinationDrag: sender];  
  
    return self;  
}
```

```
// Sent when "concludeDragOperation:" is given control
```

```
-_destinationDragFinished: (BOOL)successful
```

```
{  
    // here for backward compatibility, please use new delegate  
    // methods  
    if ([delegate respondsToSelector: @selector(destinationDragFinished:)])  
        [delegate destinationDragFinished: successful];  
  
    // new delegate method  
    if ([delegate respondsToSelector: @selector(didFinishDestinationDrag:)])  
        [delegate didFinishDestinationDrag: successful];  
  
    return self;  
}
```

```
@end
```

