

```
//
// FILENAME: eText.eTDoc.m
// SUMMARY: Implementation of interactions between eText and eTDoc
// CATEGORY: eTDoc
// PROTOCOLS: None
// INTERFACE: None
// AUTHOR: Rohit Khare
// COPYRIGHT: ©1993,94 California Institute of Technology, eText Project
//
// Implementation Comments
//      undoManager is little more than a "local cache".
//      Dragging out an etfLink is only in the margin-- cross-ref this to
//      code in Pasteboard that checks it when dropping files. This prevents the
//      click-and-drop scenario.
//      Dragging out should lazily create the dragged .etfLink file, but...
//
// History
// 10/17/94: Cleaned up for eText5.
// 08/05/94: Completely Rearchitected for 5.0. RK
//
// Imported Interfaces
//
//      #import "eText.eTDoc.h"

@implementation eText(eTDoc)
//
// Accessors
//
- setDoc:newDoc {
    etDoc = newDoc;
    undoManager = [etDoc undoManager];
    [undoManager addUndoDelegate:self];
    if (![self textLength]) { // if it's a virgin document, set the font.
        id font = [[etDoc docInfo] tagFont:BODY];
        if ([font matrix] != NX_FLIPPEDMATRIX)
            font = [Font newFont:[font name] size:[font pointSize]
                    matrix:NX_FLIPPEDMATRIX];
        [self setFont:font];
    }
    return self;
}

- etDoc {return etDoc;}
- undoManager {return undoManager;}
- touch {return [etDoc touch];}
- touch:sender {return [etDoc touch];}
```

[illegible]

```
offset.x = offset.y = 0.0;
[self dragImage:[NXImage findImageNamed:"NXLinkButton"]
  at:&org offset:&offset
  event:e pasteboard:dragPasteboard
  source:self slideBack: YES];
return self;
}
return [super mouseDown:e];
}
@end
```