

## 4. To Do Tutorial

# Subclass Example: Overriding Behavior (SelectionNotifMatrix)

You can often achieve significant gains in object behavior by making a subclass that adds only a small amount of code to its superclass. Such is the case with the subclass you'll create in this section: SelectionNotifMatrix.

The need for this class is this: An instance of NSMatrix is a control and thus can send action messages to its cell's targets; but when it contains NSTextFieldCells, action messages are sent only when users press the Return key in a cell. You want the inspector to synchronize its displays when the user selects a new item by clicking a text field. To do this, you will *override* the method in NSMatrix that is invoked when users click the matrix; in your implementation, you'll invoke the superclass method, detect the selected row, and then post a notification to interested observers.

### 1 Create template source-code files and add to the project.

Choose File arrow.eps ↩ New In Project.

In the New File In ToDo panel, select the Class suitcase, turn on the Create header switch, and type "SelectionNotifMatrix" after Name.

### 2 Add declarations to the header file.

```
#import <AppKit/AppKit.h>

extern NSString *SelectionInMatrixNotification = /* 1 */
    @"SelectionInMatrixNotification";
```

```

@interface SelectionNotifMatrix : NSMatrix
{
}

- (void)mouseDown:(NSEvent *)theEvent; /* 2 */

@end

```

These declarations do the following:

1. Declares a string constant identifying the notification that will be posted.
2. Declares **mouseDown:**, the method implemented by the superclass that SelectionNotifMatrix overrides.

### 3 Override mouseDown:

In **SelectionNotifMatrix.m**, implement **mouseDown:** as shown here.

```

- (void)mouseDown:(NSEvent *)theEvent
{
    int row;
    [super mouseDown:theEvent]; /* 1 */

    row = [self selectedRow]; /* 2 */
    if (row != -1) {
        [[NSNotificationCenter defaultCenter]
         postNotificationName:@"SelectionInMatrixNotification"
         object:self userInfo:[NSDictionary dictionaryWithObjectsAndKeys:
         [NSNumber numberWithInt:row], @"ItemIndex", nil]];
    }
}

```

```
}
```

This override of **mouseDown:** does the following:

1. Invokes NSMatrix's implementation of **mouseDown:** to allow the normal processing of this event.
2. Gets the row of the cell clicked and, if it's a valid row, creates a **userInfo** dictionary containing the clicked row, and posts the SelectionInMatrixNotification.

Now that you've created the SelectionNotifMatrix class, you must re-assign the class membership of the object in the interface. You can do this easily in Interface Builder.

#### 4 Replace the class of the matrix object.

*In Interface Builder:*

Open **ToDoDoc.nib**.

Select the matrix of editable text cells.

Open the inspector and choose Custom Class from the pop-up menu.

Select SelectionNotifMatrix in the browser of compatible classes.

TD\_OverridingBehavior.eps ⇐

**Related Concept:** ;ToDoConcepts.rtf;linkMarkername EventsandtheEventCycle;, Events and the Event Cycle