

```
#import <appkit/appkit.h>
#import "MemoController.h"
#import "MailComposer.h"
#import "TurboTFCell.h"
```

@implementation MemoController

- **appDidInit:sender**

```
{
    [self initialize:self];

    return self;
}
```

- **deliver:sender**

```
{
```

```
char *to, *from, *org, *phone;
int i;
char *msg = (char *)malloc(2048*sizeof(char));

if(strlen([toField stringValue])<3)
{
    NXRunAlertPanel("Alert", "Must specify a recipient.", "Ok", 0, 0);
    return nil;
}

to = (char *)malloc(30*(sizeof(char)));
from = (char *)malloc(50*(sizeof(char)));
org = (char *)malloc(50*(sizeof(char)));
phone = (char *)malloc(40*(sizeof(char)));

strcpy(to, [toField stringValue]);
strcpy(from, [fromField stringValue]?[fromField stringValue]:" ");
```

```
strcpy(org,[companyField stringValue]?[companyField stringValue]:" ");  
strcpy(phone,[phoneField stringValue]?[phoneField stringValue]:" ");
```

```
for(i=0;i<=strlen(to);i++)  
{  
    if(to[i]==0x0a)  
        to[i]='\0';  
}
```

```
for(i=0;i<=strlen(from);i++)  
{  
    if(from[i]==0x0a)  
        from[i]='\0';  
}
```

```
for(i=0;i<=strlen(org);i++)  
{
```

```
    if(org[i]==0x0a)
        org[i]='\0';
}

for(i=0;i<=strlen(phone);i++)
{
    if(phone[i]==0x0a)
        phone[i]='\0';
}
```

```
[toField setValue:to];
```

```
    sprintf(msg, "\nCaller:   %s from %s %s.\nPriority:   %s\nPhone:     %s\nAction: (%s)/( %s)\n\nMessage:   %s", (strlen(from)>2)?from:"Mojo Nixon"
, (strlen(org)>2)?org:"Nowhere", [rs state]?[rs title]:"called", [us state]?
"URGENT":"normal", (strlen(phone)>2)?phone:"none", [pcs state]?[pcs
title]:"", [wcs state]?[wcs title]:"", [bodyField stringValue]?
```

```
[bodyField stringValue:@"No message."];
```

```
[msgField setStringValue:msg];  
[mailComposer takeRemoteToStringValueFrom:toField];  
[mailComposer takeRemoteSubjectStringValueFrom:subField];  
[mailComposer takeRemoteBodyStringValueFrom:msgField];  
[mailComposer remoteComposeAndDeliver:self];
```

```
NX_FREE(msg);  
NX_FREE(to);  
NX_FREE(org);  
NX_FREE(from);  
NX_FREE(phone);
```

```
[self setMemoDirty:FALSE];  
[self new:self];  
return self;
```

```
}
```

```
- new:sender
```

```
{
```

```
    int i, j, rows,cols;
```

```
    if([self memoDirty])
```

```
    {
```

```
        if(!NXRunAlertPanel("Alert",  
                             "You have a memo in progress. Destroy?",  
                             "Ok","Cancel",0))
```

```
        {
```

```
            [memoPanel makeKeyAndOrderFront:self];  
            return self;
```

```
        }
```

```
    }
```

```
    [self setMemoDirty:NO];
```

```
[switchMatrix getNumRows:&rows numCols:&cols];  
for (i=0;i<rows;i++)  
    for (j=0;j<cols;j++)  
        [[switchMatrix cellAt:i :j] setIntValue:0];
```

```
[toField setStringValue:NULL];  
[fromField setStringValue:NULL];  
[bodyField setStringValue:NULL];  
[companyField setStringValue:NULL];  
[phoneField setStringValue:NULL];  
if (![mailComposer])  
    mailComposer = [[MailComposer alloc] init];  
[memoPanel display];  
[memoPanel makeKeyAndOrderFront:self];  
[toField selectText:self];  
return self;
```

```
}
```

- addSliders

{

```
float r,g,b;  
NXRect rRect;  
NXColor aColor;  
id rField,gField,bField;  
id aFont;  
const char *rValue, *gValue, *bValue;
```

```
rValue = NXGetDefaultValue([NXApp appName], "rValue");  
if(rValue) r=atof(rValue); else r=0.8;  
gValue = NXGetDefaultValue([NXApp appName], "gValue");  
if(gValue) g=atof(gValue); else g=0.8;  
bValue = NXGetDefaultValue([NXApp appName], "bValue");  
if(bValue) b=atof(bValue); else b=0.8;  
aColor = NXConvertRGBToColor(r,g,b);
```



```
[memoPanel setBackgroundColor:aColor];
```

```
rRect.origin.x = 236;  
rRect.origin.y = 220;  
rRect.size.height = 10;  
rRect.size.width = 50;
```

```
aFont = [Font newFont:"Helvetica" size:7.0];  
rSlider = [[[[[[[Slider alloc] initWithFrame:&rRect] setKnobThickness:4]  
    setMinValue:0.0] setMaxValue:1.0] setFloatValue:r]  
    setAction:@selector(setWindowColor:)] setContinuous:NO]  
    setTag:1];  
rRect.origin.x+=54;  
rField = [[[[[TextField alloc] initWithFrame:&rRect] setStringValue:"red"]  
    setFont:aFont] setBezeled:NO] setSelectable:NO];  
rRect.origin.x-=54;  
rRect.origin.y += 14;
```

```
gSlider = [[[[[[[Slider alloc] initWithFrame:&rRect] setKnobThickness:4]
    setMinValue:0.0] setMaxValue:1.0] setFloatValue:g]
    setAction:@selector(setWindowColor:)] setContinuous:NO]
    setTag:2];
rRect.origin.x+=54;
gField = [[[[[TextField alloc] initWithFrame:&rRect] setStringValue:"green"]
    setFont:aFont] setBezeled:NO] setSelectable:NO];
rRect.origin.x-=54;
rRect.origin.y += 14;
bSlider = [[[[[[[Slider alloc] initWithFrame:&rRect] setKnobThickness:4]
    setMinValue:0.0] setMaxValue:1.0] setFloatValue:b]
    setAction:@selector(setWindowColor:)] setContinuous:NO]
    setTag:3];
rRect.origin.x+=54;
bField = [[[[[TextField alloc] initWithFrame:&rRect] setStringValue:"blue"]
    setFont:aFont] setBezeled:NO] setSelectable:NO];
[[memoPanel contentView] addSubview:rSlider];
```

```
[[memoPanel contentView] addSubview:gSlider];  
[[memoPanel contentView] addSubview:bSlider];  
[[memoPanel contentView] addSubview:rField];  
[[memoPanel contentView] addSubview:gField];  
[[memoPanel contentView] addSubview:bField];
```

```
return self;
```

```
}
```

- free

```
{
```

```
[mailComposer free];  
[memoPanel free];
```

```
return [super free];
```

```
}
```

- initialize:sender

```
{  
    [bodyField setAcceptsReturn:TRUE];  
    [self addSliders];  
    [self new:self];  
    return self;  
}
```

- **setWindowColor:sender**

```
{  
    NXColor    aColor;  
    float      r,g,b;  
    char       str[16];  
  
    aColor = [memoPanel backgroundColor];  
    NXConvertColorToRGB(aColor,&r,&g,&b);  
  
    switch([sender tag])
```

```
{  
    case 1:  
        r=[sender floatValue];  
        sprintf(str,"%f",r);  
        NXWriteDefault([NXApp appName],"rValue",str);  
        break;  
    case 2:  
        g=[sender floatValue];  
        sprintf(str,"%f",g);  
        NXWriteDefault([NXApp appName],"gValue",str);  
        break;  
    case 3:  
        b=[sender floatValue];  
        sprintf(str,"%f",b);  
        NXWriteDefault([NXApp appName],"bValue",str);  
        break;  
}
```

```
aColor = NXConvertRGBToColor(r,g,b);
```

```
[memoPanel setBackgroundColor:aColor];  
[memoPanel display];
```

```
return self;
```

```
}  
- quit:sender  
{
```

```
    if([self memoDirty])  
        if(NXRunAlertPanel("Alert", "You have an undelivered memo",  
                           "Close", "Cancel", 0) == NX_CANCELTAG)  
            return nil;
```

```
[self setMemoDirty:NO];  
[NXApp terminate:self];  
return self;
```

```
}
```

```
- setMemoDirty:(BOOL)flag
{
    memoDirty = flag;
    [memoPanel setDocEdited:flag];
    return self;
}
- (BOOL)memoDirty { return memoDirty; }
```

@end