

```
/******
```

```
MainWindow.cp
```

```
*****/
```

```
#include "Fn_Prototypes.h"
```

```
#include "SampleDialog.h"
```

```
#include "SampleList.h"
```

```
#define DRAG_THRESHOLD 30
```

```
#define DIALOG_BUTTON 400
```

```
#define LIST_BUTTON 401
```

```
#define AE_BUTTON 402
```

```
#define ERR_BUTTON 403
```

```
#define TBD_BUTTON 404
```

```
extern int gNewWindowLeft;
```

```
extern int gNewWindowTop;
```

```
extern struct SysConfigRec
```

```
Boolean hasGestalt;
```

```
Boolean hasWNE;
```

```
Boolean hasColorQD;
```

```
Boolean hasAppleEvents;
```

```
Boolean hasEditionMgr;
```

```
Boolean hasHelpMgr;
```

```
long sysVersion;
```

```
;
```

```
extern struct SysConfigRec gSysConfig;
```

```
/****** Prototypes */
```

```
void MyCreateWindow( short winID,  
Ptr winRec,  
WindowPtr behind,  
int top,  
int left,  
int offset,  
int errID );
```

```
void MyDoContent( WindowPtr w,  
EventRecord *e );
```

```
void MyDoUpdateWindow( WindowPtr w );
```

```
void MyDoActivateWindow( WindowPtr w );
```

```
void MyDoDeactivateWindow( WindowPtr w );
```

```
/****** CreateWindow */
```

```
void MyCreateWindow( short winID,  
Ptr winRec,  
WindowPtr behind,  
int top,  
int left,  
int offset,  
int errID )
```

```
WindowPtr w;
```

```
ControlHandle c;
```

```

if( gSysConfig.hasColorQD )

    if ( ( w = GetNewCWindow( winID, winRec, behind ) ) == nil )
        FnErr_DisplayStrID( errID, TRUE );

    else

        FnErr_DisplayStr( "\pColor QuickDraw not supported! ",
                        "\pPlease upgrade your machine!",
                        "\p",
                        "\p",
                        TRUE );

    if (((qd.screenBits.bounds.right - gNewWindowLeft) < DRAG_THRESHOLD) ||
        ((qd.screenBits.bounds.bottom - gNewWindowTop) < DRAG_THRESHOLD))

        gNewWindowLeft = left;
        gNewWindowTop = top;

    MoveWindow( w, gNewWindowLeft, gNewWindowTop, FALSE );
    gNewWindowLeft += offset;
    gNewWindowTop += offset;

    c = GetNewControl( DIALOG_BUTTON, w );
    (**c).ctrlRfCon = DIALOG_BUTTON;
    c = GetNewControl( LIST_BUTTON, w );
    (**c).ctrlRfCon = LIST_BUTTON;
    c = GetNewControl( AE_BUTTON, w );
    (**c).ctrlRfCon = AE_BUTTON;
    c = GetNewControl( ERR_BUTTON, w );
    (**c).ctrlRfCon = ERR_BUTTON;
    c = GetNewControl( TBD_BUTTON, w );
    (**c).ctrlRfCon = TBD_BUTTON;

    ShowWindow( w );
    DrawControls( w );

/***** DoContent */

void MyDoContent( WindowPtr w, EventRecord *e )

    ControlHandle c;
    GrafPtr      savePort;
    Point        theLocation;
    short        theClick;
    int          theResponse;

    StandardFileReply mySFR;
    SFTYPEList      myTypeList;
    // OSErr        myErr;
    // Str255        myApp = "\pEditor";
    // FSSpec        mySpec;

    savePort = qd.thePort;
    SetPort( w );
    theLocation = (*e).where;
    GlobalToLocal( &theLocation );
    theClick = FindControl( theLocation, w, &c );

```

```

if( theClick == inButton )

    theResponse = TrackControl( c, theLocation, nil );
    if( theResponse != 0 )

        switch( (**c).ctrlRfCon )

            case DIALOG_BUTTON:
                MySampleDialog();
                break;
            case LIST_BUTTON:
                MyListDialog();
                break;
            case AE_BUTTON:
                myTypeList[0] = 'TEXT';
                myTypeList[1] = 'APPL';
                StandardGetFile(nil,2,myTypeList,&mySFR);
                if( mySFR.sfGood )
                    FnAE_SendOpenAE(&(mySFR.sfFile));
                //
                // option
                //
                // myErr = FSMakeFSSpec(0,0,myApp,&mySpec);
                // if( myErr == noErr )
                //     FnAE_SendOpenAE(&mySpec);
                //
                break;
            case ERR_BUTTON:
                FnErr_DisplayStr(
                    "\pThis is a test ",
                    "\pto see if this error ",
                    "\pmessage is displayed...",
                    "\p",
                    FALSE );
                break;
            case TBD_BUTTON:
                break;
            default:
                break;

```

```

SetPort( savePort );

```

```

/***** MyDoUpdateWindow */

```

```

void MyDoUpdateWindow( WindowPtr w )

```

```

    GrafPtr savePort;

    GetPort( &savePort );
    SetPort( w );
    BeginUpdate( w );
        DrawControls( w );
    EndUpdate( w );
    SetPort( savePort );

```

```

/***** MyDoActivateWindow */

```

```

void MyDoActivateWindow( WindowPtr w )

    GrafPtr      savePort;
    ControlHandle cntl;

    GetPort( &savePort );
    SetPort( w );
    cntl = ((WindowPeek)(w))->controlList;
    while( cntl != 0 )

        HiliteControl( cntl, 0 );
        cntl = (*cntl)->nextControl;

    DrawControls( w );
    SetPort( savePort );

```

/****** MyDoDeactivateWindow */

```

void MyDoDeactivateWindow( WindowPtr w )

```

```

    GrafPtr      savePort;
    ControlHandle cntl;

    GetPort( &savePort );
    SetPort( w );
    cntl = ((WindowPeek)(w))->controlList;
    while( cntl != 0 )

        HiliteControl( cntl, 255 );
        cntl = (*cntl)->nextControl;

    DrawControls( w );
    SetPort( savePort );

```

/******

MainWindow.r

*****/

```

#include <Types.r>

```

```

resource 'WIND' (400, "Main Window")

```

```

50, 10, 223, 125,

```

```

noGrowDocProc,

```

```

invisible,

```

```

goAway,

```

```

0x0,

```

```

"Window"

```

```

;
```

resource 'CNTL' (400, "Dialog")

13, 30, 33, 90,

0,

visible,

0,

0,

pushButProc,

0,

"Dialog"

;

resource 'CNTL' (401, "List")

46, 30, 66, 90,

0,

visible,

0,

0,

pushButProc,

0,

"List"

;

resource 'CNTL' (402, "AEvent")

79, 30, 99, 90,

0,

visible,

0,

0,

pushButProc,

0,

"AEvent"

;

resource 'CNTL' (403, "Error")

112, 30, 132, 90,

0,

```
visible,  
  
0,  
  
0,  
  
pushButProc,  
  
0,  
  
"Error"  
;  
  
resource 'CNTL' (404, "TBD")  
  
145, 30, 165, 90,  
  
0,  
  
visible,  
  
0,  
  
0,  
  
pushButProc,  
  
0,  
  
"???"  
;  
  
// End of File
```