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

    SampleList.cp

*****************************************************************************/

/*
    A sample list for the Starter application.
*/

/********** Includes */
#include <Dialogs.h>
#include "Fn_Prototypes.h"
#include "MainWindow.h"

/********** Defines */
#define LIST_DLOG         510
#define NIL_PTR           0L
#define ALLOCATE_MEM      0
#define IN_FRONT          (WindowPtr)-1L
#define OK_BUTTON         1
#define LIST_ITEM         2
#define RETURN_KEY        13
#define ENTER_KEY         3
#define VISUAL_DELAY      8   // standard is 8 ticks

ListHandle theList;

/********** Prototypes */
Boolean           MyListDialog( void );
pascal Boolean MyListEventFilter(   DialogPtr       theDialog,
                                    EventRecord     *theEvent,
                                    short           *itemHit );
void              MyListDoContent(    WindowPtr       w,
                                    EventRecord     *e,
                                    short           *itemHit );
void              MyFrameList(        WindowPtr       w,
                                    ListHandle      *myList );

/********** MyListDialog */

Boolean MyListDialog( void )

    WindowPtr       docWindow;
    DialogPtr       dialog;
    Boolean         done;
    Boolean         result;
    short           itemHit;
    short           itemType;
    Handle          itemHandle;
    Rect            itemRect;
    EventRecord     theEvent;
    long            finalTicks;

    ListHandle      myList;
    Rect            myDataBounds;
    Point           myCellSize;
    Rect            myRect;
    const Boolean kDoDraw = TRUE;
    const Boolean kNoGrow = FALSE;
    const           kIncludeScrollBar = TRUE;
    const           kScrollBarWidth = 15;
```

```
const             kColumnsInList = 1;
const             kTextLDEF = 0;
int               myRowNum;
int               myIndex;
Cell              myCell;
int               i;
char              *c;

Str255            myStrArray[21] =
   "\pZero",
   "\pOne",
   "\pTwo",
   "\pThree",
   "\pFour",
   "\pFive",
   "\pSix",
   "\pSeven",
   "\pEight",
   "\pNine",
   "\pTen",
   "\pEleven",
   "\pTwelve",
   "\pThirteen",
   "\pFourteen",
   "\pFifteen",
   "\pSixteen",
   "\pSeventeen",
   "\pEighteen",
   "\pNineteen",
   "\pTwenty" ;


result = FALSE;

docWindow = FrontWindow();
if( docWindow != NIL_PTR )
      MyDoDeactivateWindow( docWindow );

dialog = GetNewDialog( LIST_DLOG, ALLOCATE_MEM, IN_FRONT );

if( dialog == NIL_PTR )
      return( result );

/* AdjustMenus_(); */

ShowWindow( dialog );
FnMisc_FrameButton( dialog, OK_BUTTON );

GetDItem( dialog, LIST_ITEM, &itemType, &itemHandle, &itemRect );
myRect = itemRect;

/* set-up list */

/* start with a list with one column and no rows */
SetRect( &myDataBounds, 0, 0, kColumnsInList, 0 );

/* let list manager calculate size of a cell */
SetPt( &myCellSize, 0, 0 );

/* adjust the rectangle to leave room for scroll bar */
myRect.right = myRect.right - kScrollBarWidth;
```

```
    /* create the list */
    myList = LNew( &myRect, &myDataBounds, myCellSize,
        kTextLDEF, (WindowPtr)dialog, kDoDraw, kNoGrow,
        !kIncludeScrollBar, kIncludeScrollBar );
    (**myList).selFlags = lOnlyOne;
    theList = myList;

    /* frame list */
    MyFrameList( (WindowPtr)dialog, &myList );

    /* add strings to list */
    LDoDraw( FALSE, myList );
    myRowNum = (**myList).dataBounds.bottom;
    myIndex = 1;
    for( i = 1; i <= 20; i++ )

        myRowNum = LAddRow( 1, myRowNum, myList );
        SetPt( &myCell, 0, myRowNum );
        c = (char *)myStrArray[myIndex];
        c = c + 1;
        LSetCell( c, StrLength(myStrArray[myIndex]), myCell, myList );
        myRowNum = myRowNum + 1;
        myIndex = myIndex + 1;

    LDoDraw( TRUE, myList );
    LUpdate( dialog->visRgn, myList );

    done = FALSE;
    while( done == FALSE )

        ModalDialog( &MyListEventFilter, &itemHit );

        switch( itemHit )

            case OK_BUTTON:
                result = TRUE;
                done = TRUE;
                break;
            case LIST_ITEM:
                result = TRUE;
                break;
            default:
                break;



    DisposDialog( dialog );

    return( result );



/********** PASCAL MyListEventFilter */

pascal Boolean MyListEventFilter( DialogPtr      theDialog,
                                  EventRecord   *theEvent,
                                  short         *itemHit )

    short           thePart;
    char             key;
    short           itemType;
    Handle           itemHandle;
```

```
Rect            itemRect;
long            finalTicks;
Rect            dragRect;
Boolean         result;
Point           theLocation;
WindowPtr       theWindow;

result = FALSE;
dragRect = screenBits.bounds;

    switch( (*theEvent).what )

        case mouseDown:
            thePart = FindWindow( (*theEvent).where, &theWindow );
            if( theWindow == theDialog )

                switch( thePart )

                    case inDrag:
                        DragWindow( theDialog,
                                    (*theEvent).where,
                                    &dragRect );
                        result = TRUE;
                        break;
                    case inContent:
                        if( theDialog == FrontWindow() )

                            MyListDoContent( (WindowPtr)theDialog,
                                theEvent, itemHit );
                            result = TRUE;

                        break;
                    default:
                        break;


            break;
        case keyDown:
        case autoKey:
            key = (*theEvent).message & charCodeMask;
            if( (key == RETURN_KEY) || (key == ENTER_KEY) )

                GetDItem( theDialog,
                          OK_BUTTON,
                          &itemType,
                          &itemHandle,
                          &itemRect );
                HiliteControl( (ControlHandle)itemHandle,
                               inButton );
                Delay( VISUAL_DELAY, &finalTicks );
                HiliteControl( (ControlHandle)itemHandle, 0 );
                *itemHit = OK_BUTTON;
                result = TRUE;

            /* Handle other keyboard equivalents here */
            break;
        case updateEvt:
            if( (WindowPtr)(*theEvent).message != theDialog )

                MyDoUpdateWindow( (WindowPtr)(*theEvent).message );

            else
```

```
                    MyFrameList( (WindowPtr)theDialog, &theList );
                    LUpdate( theDialog->visRgn, theList );
                    FnMisc_FrameButton( theDialog, OK_BUTTON );

                break;
            case activateEvt:
                if( (WindowPtr)(*theEvent).message != theDialog )

                    /*
                    DoActivate_( (WindowPtr)(*theEvent).message,
                                ((*theEvent).modifiers & activeFlag),
                                *theEvent );
                    */

                 break;
            default:
                 break;


    return( result );



/********** MyListDoContent */

void MyListDoContent( WindowPtr       w,
                      EventRecord     *e,
                      short           *itemHit )
/*
    Use this procedure when user clicks in content region of TE window.
    Uses the TnTE_ScrollProc below to take care of scroll bar function.
    Also uses FnTE_AdjustText function.
*/

    int                 cntlCode;
    ControlHandle       cntl;
    GrafPtr             savePort;

    short           itemType;
    Handle          itemHandle;
    Rect            itemRect;
    long            finalTicks;

    GetPort(&savePort);
    SetPort(w);
    *itemHit = 0;
    GlobalToLocal(&e->where);
    if( (cntlCode = FindControl(e->where, w, &cntl)) == inButton )

        if( TrackControl( cntl, e->where, 0L ) )
            *itemHit = OK_BUTTON;

    else

        GetDItem( (DialogPtr)w, LIST_ITEM, &itemType, &itemHandle,
            &itemRect );
        if( PtInRect( e->where, &itemRect ) )

            *itemHit = LIST_ITEM;
            if( LClick( e->where, e->modifiers, theList ) )
```

```
                GetDItem( (DialogPtr)w,
                        OK_BUTTON,
                        &itemType,
                        &itemHandle,
                        &itemRect );
                HiliteControl( (ControlHandle)itemHandle,
                                inButton );
                Delay( VISUAL_DELAY, &finalTicks );
                HiliteControl( (ControlHandle)itemHandle, 0 );
                *itemHit = OK_BUTTON;



    SetPort(savePort);


/********** MyFrameList */

void MyFrameList( WindowPtr w, ListHandle *myList )

    Rect        myBorder;
    PenState myPenState;
    GrafPtr   savePort;

    savePort = thePort;
    SetPort( w );
    myBorder = (***myList).rView;
    GetPenState( &myPenState );
    PenNormal();
    PenSize(1,1);
    InsetRect( &myBorder, -1, -1 );
    FrameRect( &myBorder );
    SetPenState( &myPenState );
    SetPort( savePort );



/*********************************************************************

    SampleList.r

*********************************************************************/

/* THINK Rez resource format */
#include <Types.r>

resource 'DLOG' (510, "List Dialog")

45, 10, 317, 205,

movableDBoxProc, /* movableDBoxProc or noGrowDocProc */

invisible,

goAway,

0x0,

510,
```

```
"List Dialog"
;

resource 'DITL' (510, "List Dialog")

/* array DITLarray: 1 elements */
        /* [1] */

242, 68, 262, 128,

Button


enabled,


"OK"


,

/* [2] */

5, 5, 229, 190,

UserItem


disabled




;

/*****************************************************************

    SampleList.h

*****************************************************************/

/********** Prototypes */
extern Boolean MyListDialog( void );

// End of File
```