Custom File Dialog

```
// Custom File Dialog
// This is an example of how to access a file within the dlgHook
// of an SFPGetFile call.  This technique would work for SFPPutFile
// as well, of course.  Under System 7, you should use CustomGetFile
// and CustomPutFile for this purpose.

// This program uses a copy of the standard file dialog box from the
// System file (DLOG and DITL resource id's -4000) with a button and
// a static text item added.  When a user selects a file in the
// standard file dialog box and clicks the button, the first 30 characters
// in the file show up in the static text item.


#include <StandardFile.h>
// Assumes inclusion of <MacHeaders>

#define CONTENTS_BUTTON         11
#define FILE_FIELD              12

SFReply             gReply;
Boolean             gDone;

void ToolBoxInit (void);
pascal short myDlgProc (short theItem, DialogPtr theDialog);


void ToolBoxInit ()
{
    InitGraf (&thePort);
    InitFonts ();
    InitWindows ();
    InitMenus ();
    TEInit ();
    InitDialogs (nil);
    InitCursor ();
}


pascal short myDlgProc (short theItem, DialogPtr theDialog)
{
    short           fRefNum;
    OSErr           err;
    long            inOutCount = 30;
    char            bufPtr[31];
    short           wdRefNum;
    short           iType;
    Handle  iHandle;
    Rect            iRect;


    if (theItem == CONTENTS_BUTTON)
    {
            // curDirStore is a low-memory global which stores the
            // hard directory id of the last directory used by
```

```
        // standard file
        err = OpenWD ( gReply.vRefNum, CurDirStore, 0, &wdRefNum );
        err = FSOpen (gReply.fName, wdRefNum, &fRefNum);
        err = FSRead ( fRefNum, &inOutCount, bufPtr );
        if (!err) {
            bufPtr[30] = 0;     // add null character to make a C String
            GetDItem ( theDialog, FILE_FIELD, &iType, &iHandle, &iRect );
            SetIText (iHandle, CtoPstr (bufPtr));
            FSClose (fRefNum);
        }
        err = CloseWD ( wdRefNum );
    }
    return theItem;
}

main ()
{
    static Point    where;
    SFTypeList      typeList;


    ToolBoxInit();
    typeList[0] = 'TEXT';
    SFPGetFile ( where, "\p", nil, 1, typeList, myDlgProc,
                        &gReply, 128, nil );
}

/* Rez description file for DLOG and DITL for custom file dialog */

#include "Types.r"

resource 'DITL' (128, purgeable) {
    {   /* array DITLarray: 12 elements */
        /* [1] */
        {138, 256, 156, 336},
        Button {
            enabled,
            "Open"
        },
        /* [2] */
        {1152, 59, 1232, 77},
        Button {
            enabled,
            ""
        },
        /* [3] */
        {163, 256, 181, 336},
        Button {
            enabled,
            "Cancel"
        },
        /* [4] */
        {39, 232, 59, 347},
        UserItem {
            disabled
        },
			/* [5] */
			{68, 256, 86, 336},
			Button {
				enabled,
				"Eject"
			},
			/* [6] */
			{93, 256, 111, 336},
			Button {
				enabled,
				"Drive"
			},
			/* [7] */
			{39, 12, 185, 230},
			UserItem {
				enabled
			},
			/* [8] */
			{39, 229, 185, 246},
			UserItem {
				enabled
			},
			/* [9] */
			{124, 252, 125, 340},
			UserItem {
				disabled
			},
			/* [10] */
			{1044, 20, 1145, 116},
			StaticText {
				disabled,
				""
			},
			/* [11] */
			{191, 256, 209, 336},
			Button {
				enabled,
				"Contents"
			},
			/* [12] */
			{191, 13, 211, 247},
			StaticText {
				disabled,
				""
			}
		}
};

resource 'DLOG' (128, purgeable) {
		{0, 0, 224, 359},
		dBoxProc,
		invisible,
		noGoAway,
		0x0,
		128,
		""
```

```
};
```