Printing a Text File

```
/*
 * Printing a text file
 * This is a very simple demonstration that shows how to use the
 * Print Manager to print a text file.
 */

// Assumes inclusion of <MacHeaders>
#include <PrintTraps.h>
#include <stdio.h>
#include <pascal.h>


/* Do you want to prompt for the page setup? */
Boolean       DoPageSetUp  = FALSE;

/* Globals */
FILE *aRandomFile;

/* Prototypes */
void Init(void);
Boolean DoneDrawingPagesOfATextFile(FILE *someFile);
void PrintStuff(void);
FILE *GotATextFile(void);

void
Init(void)
{
      InitGraf(&thePort);
      InitFonts();
      InitWindows();
      TEInit();
      InitDialogs(nil);
      InitCursor();
      MaxApplZone();
}

Boolean DoneDrawingPagesOfATextFile(FILE *someFile)
{
      short            i;
      Str255           aStringOfText;

      TextSize(10);
      TextFont(courier);
      for (i = 1; i <= 50; ++i) {
            if (feof(someFile))
                  return TRUE;
            fgets((char *)aStringOfText, 255, someFile);
            CtoPstr((char *)aStringOfText);
            /*
             * Carrige return characters mess up DrawString, so they are removed.
             * Also, tabs are not handled correctly.  They are left as an exercise
             * for the programmer!
             */
            if (aStringOfText[aStringOfText[0]] == '\n')
```

```
                aStringOfText[aStringOfText[0]] = ' ';
            MoveTo(10, 14 * i);
            DrawString(aStringOfText);
        }
        return FALSE;
}

void PrintStuff(void)
{
        GrafPtr         savedPort;
        TPrStatus       prStatus;
        TPPrPort        printPort;
        THPrint         hPrint;
        Boolean         ok;

        GetPort(&savedPort);
        PrOpen();
        hPrint = (THPrint)NewHandle(sizeof(TPrint));/* *not* sizeof(THPrint) */
        PrintDefault(hPrint);
        if (DoPageSetUp)
            ok = PrStlDialog(hPrint);
        else
            ok = PrValidate(hPrint);
        ok = PrJobDialog(hPrint);
        if (ok)
        {
            printPort = PrOpenDoc(hPrint, nil, nil);
            SetPort(&printPort->gPort);
            PrOpenPage(printPort, nil);
            /*
             * Now draw a single page.  You decide how many pages are drawn, and
             * what is drawn in each page.  QuickDraw commands are automatically
             * translated into commands to the printer. You could draw multiple pages
             * like this.
             */
            while (!DoneDrawingPagesOfATextFile(aRandomFile)) {
                PrClosePage(printPort);   /* Close the currently open page. */
                PrOpenPage(printPort, nil);       /* and open a new one. */
            }
            PrClosePage(printPort);
            PrCloseDoc(printPort);
            /* Print spooled document, if any. */
            if ((**hPrint).prJob.bJDocLoop == bSpoolLoop && PrError() ==
                noErr)
                PrPicFile(hPrint, nil, nil, nil, &prStatus);
        }
        else {
            /* You will want to add error handling here... */
            SysBeep(5);
        }
        PrClose();
        SetPort(savedPort);
}

/* Open any text file */
FILE * GotATextFile(void)
```

```
{
	SFReply		theReply;
	SFTypeList	theTypeList = {'TEXT'};
	Point		where = {40, 60};
	OSErr		PotentialErr;
	FILE		*someFile;

	SFGetFile(where, "\p", nil, 1, theTypeList, nil, &theReply);
	if (theReply.good) {
		PotentialErr  = SetVol(nil, theReply.vRefNum);
		someFile = fopen(PtoCstr((char *)theReply.fName), "r");
		return someFile;
	}
	return nil;
}

main()
{
	Init();
	if (aRandomFile = GotATextFile()) {
		PrintStuff();
		fclose(aRandomFile);
	}
}
```