The Closest Thing to PRFect!

ADAM L. MENKES

Bypassing the Print menu to create your own .PRF files

Currently the only way to create or modify a print form file (.PRF) is through the Control Center using the **Print: Save settings to print form** option. You could, of course, set the printer system variables (_padvance, _pageno, _pbpage, _pcopies, _pdriver, _pecode, _peject, _pepage, _plength, _ploffset, _ppitch, _pquality, _pscode, _pspacing and _pwait) from the dot prompt or within a program, then SET PRINTER to the appropriate DOS DEVICE. But since these system memory variables can't be saved to a .MEM file, you must proceed to the Control Center and save them to a print form file through the Print menu in the design surfaces, program or memo editor. Moreover, in RunTime, you would not easily be able to allow users to create their own .PRF files.

The UDF MakePRF will create or modify an existing .PRF file based on the current system settings. The structure of the .PRF file is listed below (all 698 bytes). Just prior to running the UDF set the _pageno variable to 1. This variable will increase every time you do any streaming output commands, including DISPLAY STATUS, and may store an unexpected number to the .PRF file and show up on the resulting printed document.

MakePRF() requires three other UDFs, GetInfo(), TLineNo(), and TLine(), which are published in the April '92 issue of *TechNotes/dBASE IV*. These get information about the printer destination and read the CONFIG.DB file to determine printer numbers.

Structure of a .PRF Binary File

Byte	Contents	Meaning
0-1	16 bit number	dBASE version - integer (2 bytes)
2	1 byte	Destination : 0) Printer 1) DOS file
3-83	80 bytes	Print to File <Filename>, includes <Path>
84-85	16-bit number	Printer # from CONFIG.DB; DEFAULT is 0
86-166	80 bytes	Printer driver : <Filename> (..PR2, .PR3)
167	1 byte	Echo to screen : 0) No 1) Yes
168	1 byte	Text pitch : _ppitch -0) Pica 1) Elite 2) Condensed 3) DEFAULT
169	1 byte	Quality print : _pquality - 0) Yes 1) No 2) DEFAULT
170	1 byte	Eject page : _peject - 0) After 1) Before 2) Both 3) None
171	1 byte	Pause between pages : pwait - 0) .F. 1) .T.
172	1 byte	Advance page using : _padvance - 0) Formfeed 1) Linefeeds
173-427	254 Bytes	Starting control codes with null terminator.
428-682	254 Bytes	Ending control codes with null terminator.
683-684	16-bit number	Begin on page : _pbpage
685-686	16-bit number	End on page : _pepage
687-688	16-bit number	Page number : _pageno
689	1 byte	Spacing : _pspacing -1) Single 2) Double 3) Triple
690-691	16-bit number	Number of copies : _pcopies
692-693	16-bit number	Page length : _plength
694-695	16-bit number	Width of label * number of labels across Irrelevant since the settings are read from the .LBL file. May fill with NULLs CHR(0) (or any number) instead.
696-697	16-bit number	Page left offset : _ploffset

Function: MakePRF()

```
FUNCTION MakePRF
  * - Author: Adam L. Menkes
  PARAMETERS cPRF
  *-- Requires GETINFO(), TLINENO(), TLINE()
  *-- Differences and Assumptions :
  *-- 1) No DEFAULT settings are honored (i.e. Quality Yes/No ONLY)
  *-- 2) dBASE started with CONFIG.DB configuration file.
  *-- 3) ECHO TO SCREEN uses setting of SET ECHO.
  *-- 4) Uses <File> as specified by SET PRINTER TO <File>
  *--    instead of <Report>.PRT or .TXT.

  cTmpForm = _pform
  _pform = ""
  cPRF   = LTRIM(RTRIM(cPRF))
  cPRF   = cPRF + IIF(RIGHT(cPRF, 4) = ".PRF", "", ".PRF")
  lnOpen = IIF(FILE(cPRF), FOPEN(cPRF, "RW"), FCREATE(cPRF, "RW"))
  nPage  = _pageno
  cPort  = GetInfo("PRINT")     && UDF to get current printer port
  nDest  = IIF(LEFT(cPort, 3) $ 'PRN, LPT, COM, AUX, NUL', 0, 1)

  nPLineno = TLineNo(HOME() + "Config.DB", "PRINTER 1", .F.)
  nPrinter = 1
  DO WHILE nPrinter <= 4
    nPLine = TLINE(HOME() + "Config.DB", nPLineno + nPrinter - 1)
    IF _pdriver $ nPLine .AND. cPort $ nPLine
      EXIT
    ENDIF
    nPrinter = nPrinter + 1
  ENDDO

  lnWrite = FWRITE(lnOpen, CHR(4))                    && Byte 0
  lnWrite = FWRITE(lnOpen, CHR(0))                    && Byte 1
  lnWrite = FWRITE(lnOpen, CHR(nDest))                && Byte 2
  cFile   = IIF(nDest = 0, CHR(0), cPort)

  lnWrite = FWRITE(lnOpen, cFile)                     && Bytes 3 - 83
  x = IIF(LEN(cFile) = 0, 1, 0)
  DO WHILE x <= 80 - LEN(cFile)
    lnWrite = FWRITE(lnOpen, CHR(0))
    x = x + 1
  ENDDO

  lnWrite = FWRITE(lnOpen, CHR(nPrinter))             && Byte 84
  lnWrite = FWRITE(lnOpen, CHR(0))                    && Byte 85

  lnWrite = FWRITE(lnOpen, _pdriver)                  && Bytes 86 - 166
  x = 0
  DO WHILE x <= 80 - LEN(_pdriver)
    lnWrite = FWRITE(lnOpen, CHR(0))
    x = x + 1
  ENDDO

  *-- No system variable for Echo to screen.          && Byte 167
  lnWrite = FWRITE(lnOpen, CHR(IIF(SET("ECHO") = "ON", 1, 0)))
  *--                                                 && Byte 168
```

Function: MakePRF() *continued*

```
    lnWrite = FWRITE(lnOpen, CHR(AT(LEFT(_pPITCH, 1), "PECD") - 1))
    *-- Can not check for DEFAULT, only Yes/No                    && Byte 169
    lnWrite = FWRITE(lnOpen, IIF(_pQUALITY, CHR(0), CHR(1)))
    *--
    lnWrite = FWRITE(lnOpen, CHR(VAL(SUBSTR(;
         "0-AFTER 1-BEFORE 2-BOTH 3-NONE", AT(_pEJECT, ;
         "0-AFTER 1-BEFORE 2-BOTH 3-NONE") - 2, 1))))             && Byte 170
    lnWrite = FWRITE(lnOpen, IIF(_pwait, CHR(1), CHR(0)))         && Byte 171
    lnWrite = FWRITE(lnOpen, IIF(_padvance = "FORMFEED", ;
           CHR(0), CHR(1)))                                       && Byte 172
    lnWrite = FWRITE(lnOpen, _pscode)                             && Bytes 173- 427
    x = 0
    DO WHILE x <= (254 - LEN(_pscode))
      lnWrite = FWRITE(lnOpen, CHR(0))
      x = x + 1
    ENDDO

    lnWrite = FWRITE(lnOpen, _pecode)                             && Bytes 428- 682
    x = 0
    DO WHILE x <= (254 - LEN(_pecode))
      lnWrite = FWRITE(lnOpen, CHR(0))
      x = x + 1
    ENDDO

    lnWrite = FWRITE(lnOpen, CHR(MOD(_pbpage, 256)))              && Byte 683 -
    lnWrite = FWRITE(lnOpen, CHR(INT(_pbpage / 256)))             && Byte 684
    lnWrite = FWRITE(lnOpen, CHR(MOD(_pepage, 256)))              && Byte 685 -
    lnWrite = FWRITE(lnOpen, CHR(INT(_pepage / 256)))             && Byte 686
    lnWrite = FWRITE(lnOpen, CHR(MOD(nPage, 256)))                && Byte 687 -
    lnWrite = FWRITE(lnOpen, CHR(INT(nPage / 256)))               && Byte 688
    lnWrite = FWRITE(lnOpen, CHR(_pspacing - 1))                  && Byte 689
    lnWrite = FWRITE(lnOpen, CHR(MOD(_pcopies, 256)))             && Byte 690 -
    lnWrite = FWRITE(lnOpen, CHR(INT(_pcopies / 256)))            && Byte 691
    lnWrite = FWRITE(lnOpen, CHR(MOD(_plength, 256)))             && Byte 692 -
    lnWrite = FWRITE(lnOpen, CHR(INT(_plength / 256)))            && Byte 693
    lnWrite = FWRITE(lnOpen, CHR(0))                              && Byte 694 -
    lnWrite = FWRITE(lnOpen, CHR(0))                              && Byte 695
    lnWrite = FWRITE(lnOpen, CHR(MOD(_ploffset, 256)))            && Byte 696 -
    lnWrite = FWRITE(lnOpen, CHR(INT(_ploffset / 256)))           && Byte 697
    llClose = FCLOSE(lnOpen)

    mSAFETY = SET("SAFETY")
    SET SAFETY OFF
    _pform = cTmpForm
    SET SAFETY &mSafety
  RETURN (FSIZE(cPRF) = 698)
```