PostScript Quick Tips: Printing Font Samples

Apple's LaserWriter Utility includes a Print Font Samples function. In this month's Quick Tip, we'll provide you with a better tool. The PostScript code described here has the following advantages over Apple's tool:

- Our PostScript code is platform independent. It can be used on a Mac, a DOS or Windows machine, or a Unix workstation.
- Our code is more robust. Occasionally, corrupted fonts may be found on a printer's hard disk. Apple's utility will abort if it encounters a bad font, without reporting which font caused the problem. Our code will print a message and continue if any defective fonts are found.
- Our code includes the Version number (when available) for all fonts. This information can be useful for diagnosing discrepancies between multiple printers.
- For Multiple Master fonts, our code displays each of the master designs. Multiple Master fonts with two design axes have four master designs; Multiple Master fonts with three design axes have eight master designs.
- Our code doesn't include the reencoded fonts produced by earlier versions of the LaserWriter driver on the Mac.

```
%! Copyright (C) 1986-1993 Herb Weiner.
%! All rights reserved.

/Top 724 def /y Top def /Left 54 def /Bottom 72 def
/TitleHeight 740 def /Width 512 def /Middle 264 def
/vx 224 def /PageNumber 0 def /Pointsize 12 def
/Leading 18 def /myString 100 string def /pn () def
/fl 25000 array def /fi 0 def /version () def /text
  (The quick brown fox jumps over the lazy dog.) def
/Helvetica-Bold findfont dup 12 scalefont
  /TF exch def 10 scalefont /FF exch def
/Helvetica findfont 8 scalefont /BF exch def
/statusdict where {pop statusdict /printername known
  {/pn 100 string statusdict begin printername end
  def} if} if
/insert {dup length string copy fi 0 eq
  {fl exch 0 exch put} {fi 0 1 fi 1 sub {fl 1 index
  get 3 index gt {exch pop exit} {pop} ifelse} for
  fi 1 sub -1 2 index {fl exch dup 1 add exch fl
  exch get put} for fl 3 1 roll exch put} ifelse
  /fi fi 1 add def} def
/PrintTitle {TF setfont userdict /PageNumber
  PageNumber 1 add put Left 18 add TitleHeight
  moveto (Printer Font Catalog) show pn length 0 gt
  {( for \252) show pn show (\272) show} if
  PageNumber myString cvs dup stringwidth pop
  (Page ) stringwidth pop add Width 18 sub exch sub
  Left add TitleHeight moveto (Page ) show show
  newpath Left TitleHeight 9 sub moveto Width 0
  rlineto 0 30 rlineto Width neg 0 rlineto
  closepath stroke} bind def
/PrintPage {PrintTitle showpage userdict
  /y Top put} bind def
/newLine {y Bottom le {PrintPage} if userdict
  /y y Leading sub put} bind def
/showLine {Left y moveto show} bind def
/showSample {Middle y moveto {currentdict Pointsize
  scalefont setfont text show} stopped FontType 4 ne
  and {TF setfont (*** DEFECTIVE FONT ***) show} if
  clear} bind def
/showFont {newLine dup myString cvs FF setfont
  showLine /mySave save def {findfont dup /FontInfo
  known {dup /FontInfo get begin} if begin
  BF setfont vx y moveto version show showSample
  FontType 4 eq { {userdict /nMasters
  BlendDesignPositions length put userdict /nAxes
  BlendAxisTypes length put 0 1 nMasters 1 sub
  {userdict exch /i exch put newLine save Left
  12 add y moveto BF setfont 0 1 nAxes 1 sub
  {BlendAxisTypes 1 index get myString cvs show ( )
  show BlendDesignPositions i get 1 index get exch
  BlendDesignMap exch get {dup 1 get 2 index eq
  {0 get myString cvs show ( ) show exit} {pop}
  ifelse} forall pop} for Middle y moveto
  currentdict [1.0 nMasters 1 sub {0.0} repeat
  nMasters i roll] makeblendedfont /myFont exch
  definefont Pointsize scalefont setfont text show
  y exch PageNumber exch restore userdict exch
  /PageNumber exch put userdict exch /y exch put}
  for} stopped {clear} if} if} stopped
  cleardictstack {clear Middle y moveto TF setfont
  (*** DEFECTIVE FONT ***) show} if y PageNumber
  mySave restore /PageNumber exch def /y exch def}
  bind def
newLine TF setfont (Fonts in printer's memory)
  showLine /y y 6 sub def
```

```
FontDirectory {dup /FontName known
  {/FontName get 1 index eq {myString cvs insert}
  {pop} ifelse} {pop pop} ifelse} forall
  0 1 fi 1 sub {fl exch get dup showFont} for
  /fi 0 def statusdict /diskstatus known
  {statusdict /diskonline known {statusdict
  /diskonline get exec} {true} ifelse
  {PrintPage newLine TF setfont
  (Fonts on printer's disk(s)) showLine
  /y y 6 sub def (fonts/*) {dup length 6 sub 6 exch
  getinterval insert} myString filenameforall
  0 1 fi 1 sub {fl exch get dup showFont} for} if}
  if PrintPage
```

The figure below illustrates an interesting portion of the output. Adobe's Minion Multiple Master font has eight master designs, each of which is included in the sample output. (The Minion Multiple Master package also includes MinionMM-It, which has eight master designs as well.) This output helps to demonstrate the huge effort required to design a three axis Multiple Master font. The "001.000" indicates that we are using version 1.0 of this font.

How to Use it

- Use any text editor to create a text file containing the PostScript code.
- Download the PostScript code to any PostScrpt printer to print the font samples. If your printer has a hard disk attached, this program prints samples for all fonts on the printer's hard disk as well as those in your printer's ROM. Otherwise, this program only prints fonts in your printer's memory (fonts in ROM and fonts which have been downloaded into your printer's RAM). Note that it takes several minutes per page to print font samples from the printer's hard disk.
- If you wish, you can change the sample phrase. For example, in place of "The quick brown fox ...," you may prefer "The five boxing wizards jump quickly," which covers the entire alphabet using fewer letters.

How it Works

Apple's LaserWriter Utility requires a bidirectional communication channel to the printer. It first downloads a PostScript program to obtain a list of fonts from the printer. The response is transmitted back to the LaserWriter Utility, which then creates a document containing the font samples.

In contrast, our code performs all computation and formatting on the printer. This makes our PostScript program somewhat unconventional, since there is no way to determine by examing the program how many pages will be produced or what fonts will be used. In fact, our program is useful preciesely because it is not device independent. In general, different printers will produce different results.

Since the formatting is done entirely in the printer, it is even possible to shutdown the computer once the code has been completely downloaded, and the printer will continue to print (often many pages) until it is finished.

The complexity of PostScript fonts makes it nearly impossible to completely check a font for errors or corruption before using it. Fortunately, the PostScript language offers an alternative — the *stopped* operator. This operator allows us to trap errors and recover from them after they occur. Unfortunately, the uses of *stopped* are rather limited, since most PostScript programs can't simply report an error (like we do) and continue.

Since corrupted fonts can produce errors when we load them using *findfont*, or later when we attempt to make use of them using *show*, we use *stopped* to recover from errors in either situation. Note that we don't make any attempt to determine what caused the error; only that an error did occur. We report that the font in question is defective, and proceed to the next font.

MinionMM 001.000

Weight 345 Width 450 OpticalSize 6	The quick brown fox jumps over the lazy dog.
Weight 620 Width 450 OpticalSize 6	The quick brown fox jumps over the lazy dog.
Weight 345 Width 600 OpticalSize 6	**The quick brown fox jumps over the lazy dog.**
Weight 620 Width 600 OpticalSize 6	The quick brown fox jumps over the lazy dog.
Weight 345 Width 450 OpticalSize 72	**The quick brown fox jumps over the lazy dog.**
Weight 620 Width 450 OpticalSize 72	The quick brown fox jumps over the lazy dog.
Weight 345 Width 600 OpticalSize 72	The quick brown fox jumps over the lazy dog.
Weight 620 Width 600 OpticalSize 72	**The quick brown fox jumps over the lazy dog.**