LaserWriter IIf and IIg: Printing Font Sample with Many Fonts

Article Created: 27 March 1992
Article Last Reviewed:
Article Last Updated:

TOPIC ---

I have a LaserWriter IIg with a 40MB hard drive attached.

I've reformatted the hard drive several times and downloaded fonts with the
LaserWriter Utility (as shipped with the LaserWriter IIg). When I try to
print a Font Sample, the LaserWriter processes it for a while and then
comes back with an error "Unable to Print Because of A PostScript Error."

I've tried it several times with different Fonts and found that when there
are more than 40 or so fonts it will get the error, but when there are 30
fonts or less, it will print.

DISCUSSION ---

This is a known limitation of the code that prints the font samples.
Printing a font sample of many different fonts can exceed the memory of a
LaserWriter (depending on the amount of memory installed). The code could
work around this by breaking the samples across pages as memory grew short,
but it doesn't.

Below is PostScript code that prints a sample of all available fonts,
one per page. (This can take a long time with many fonts.) It was written
by a Sun Microsystems employee, but can be freely distributed. We've
inserted some code to prevent page clipping on a LaserWriter IIg.

Copy the text between the "cut" lines and paste it into a blank text
document. Once saved, you can download it to a LaserWriter with the
LaserWriter Utility included with the LaserWriter IIf and IIg.

```
---- cut here ----
%!PS-Adobe-1.0
%%DocumentFonts: (atend)
%%Title: FontBook
%%Creator: Keith Vincent, SSE, Sun Microsystems
%%CreationDate: Mar 18, 1988
%%Pages: (atend)
```

```postscript
%%EndComments

/currentpagedevice where        %D.W. 3/92 - Disable PhotoGrade
{ pop currentpagedevice /PreRenderingEnhance known
  { << /PreRenderingEnhance false >> setpagedevice } if
} if %currentpagedevice

/inch { 72 mul } def
/cvis { 1 string dup 0 4 -1 roll put } def

/DrawBoundMarks {       % width x y => - (draw the character bounding marks)
 gsave
  0 setlinewidth
  moveto
  0 1 16 div inch neg rmoveto
  0 7 16 div inch rlineto
  1 16 div inch neg 6 16 div inch neg rmoveto
  2 16 div inch add 0 rlineto              % this is where width gets used
  1 16 div inch neg 1 16 div inch neg rmoveto
  0 7 16 div inch rlineto stroke
 grestore
} def

/LabelChar {                    % num wid x y => - (print the character label)
 gsave
  /Times-Roman findfont 6 scalefont setfont          % num wid x y
  moveto                                              % num wid
  1 64 div inch neg 5 32 div inch neg rmoveto         % num wid
  exch                                               % wid num
  16 (    ) cvrs show                                % wid
  ( ) show                                           % wid
  1000 mul { cvi } stopped { pop -1 } if 1000 div (      ) cvs show  % -
 grestore
} def

/LabelName {                    % str x y => -    (print the character name)
 gsave
  moveto
  1 64 div inch neg 7 32 div inch neg rmoveto
  /Times-Roman findfont 6 scalefont setfont
  show
 grestore
} def

/LowAscii
<20212223242526272829202A2B2C2D2E2F3031323334353637383939A3B3C3D3E3F4041424344
45464748494A4B4C4D4E4F50515253545556575859A5B5C5D5E5F606162636465666768696
A6B6C6D6E6F707172737475767778797A7B7C7D7E7F> def
/HiAscii
<A0A1A2A3A4A5A6A7A8A9AAABACADAEAFB0B1B2B3B4B5B6B7B8B9BABBBCBDBEBFC0C1C2C3C4
C5C6C7C8C9CACBCCCDCECFD0D1D2D3D4D5D6D7D8D9DADBDCDDDEDFE0E1E2E3E4E5E6E7E8E9E
AEBECEDEEEFF0F1F2F3F4F5F6F7F8F9FAFBFCFDFEFF> def
%%EndProlog
```

```
%%Page: 1

/str 250 string def
/buf 256 string def

FontDirectory {
 dup type /integertype eq { pop pop }
 {
  pop
  0 1 255 { buf exch 0 put } for
  /GivenFont exch def
  GivenFont findfont 24 scalefont setfont
  currentfont /FontName get cvx buf cvs
  gsave
   /Times-Roman findfont 24 scalefont setfont
   1 inch 10.5 inch moveto
   buf show
   ( \(24 point\)) show
  grestore

  32 16 127 {
   /yval exch def
   /y 9.5 inch 11 16 div yval 16 div 2 sub mul inch sub def
   0 1 15 {
    /xval exch def
    /num yval xval add def
    /x xval .5 mul inch 7 16 div inch add def
    /width num cvis stringwidth pop def
    /name { currentfont /Encoding get } stopped
     { pop pop /.notdef }
     { { num get } stopped
      { /.notdef }
      if
     }
    ifelse def
    width x y DrawBoundMarks
    num width 24 div 0.0005 add x y LabelChar
    name (                        ) cvs x y LabelName
    x y moveto num cvis show
   } for
  } for
  160 16 255 {
   /yval exch def
   /y 10 inch 11 16 div yval 16 div 2 sub mul inch sub def
   0 1 15 {
    /xval exch def
    /num yval xval add def
    /x xval .5 mul inch 7 16 div inch add def
    /width num cvis stringwidth pop def
    /name { currentfont /Encoding get } stopped
     { pop pop /.notdef }
     { { num get } stopped
      { /.notdef }
```

```
        if
       }
     ifelse def
     width x y DrawBoundMarks
     num width 24 div 0.0005 add x y LabelChar
     name (                        ) cvs x y LabelName
     x y moveto num cvis show
    } for
  } for

  GivenFont findfont 10 scalefont setfont
  7 16 div inch 5.5 inch moveto LowAscii show

  7 16 div inch .5 inch moveto HiAscii show
  showpage
  clear
 } ifelse
} forall
%%Trailer
%%DocumentFonts:
%%Pages:
---- cut here ----
```

Copyright 1992 Apple Computer, Inc.