PostScript: Daisywheel emulator for the LaserWriter (1 of 2)

Article Created: 2 May 1986
Article Reviewed/Updated: 8 February 1995

TOPIC ---

This article provides PostScript information for building a daisy wheel emulator for the LaserWriter. This information is provided by Adobe Customer Support. Apple Computer Inc. is not responsible for the contents of this article.

DISCUSSION ---

Caution

If you choose to use the PostScript code provided in this article, you assume all risks involved in making these changes.PostScript code, if not entered correctly, can place the LaserWriter into a condition requiring service.

Below is a generalized PostScript program that provides a framework for building ANY daisy wheel emulator. Anyone with knowledge of PostScript will be able to figure out how to customize this program to meet their special needs. Unfortunately the sensitivity to ^C^D^T^S^Q,CR,LF is not avoidable.

```
%!
%Copyright (c) 1985, Adobe Systems Incorporated, Palo Alto, CA.
%"Used with permission."
%This program may be reproduced, used, and sold, as long as the above copyright
%notice appears in all copies (in any media) and on labels of machine-readable
%media.
%
%Definitions for program "Daisyprint".
%Last revision: MJF 1/29/86.
%
% Generic daisy-wheel emulator.  Current parameters are set up for Vanilla.
% None of the following defs should be changed.  Procs and defs that can be
% safely changed are those in the second half of this program, following the
% double bar line: ========================================================
% Instructions for use at end of this listing.

/DaisyEmDict 30 dict def
/Daisyprint  {DaisyEmDict begin printfile end} def
```

```
DaisyEmDict begin   %-----------------------------------------------------

/printfile   %main job loop
  { initjob
    CmdDict begin
    {currentfile read  %Read one byte at a time.
                       %Be sure not to send ^c,^d,^t,^s,^q

      {printchar}{showpage exit} ifelse
    }loop
    popCmdStack end   %CmdDict
  }def

/initjob
  {  RomanFont setfont
     /linect 1 def
     /charct 0 def
     /CmdDictBase countdictstack 1 add def
     initpage
  }def

/initpage
  { orientpage
    LeftMargin FirstLine moveto
    /linect 1 store
    /charct 0 store
  }def
%Caution - This program stacks dictionaries.  It is therefore wise to use
%"store" rather than "def" to assign new values to old variables.

/printchar    %( n  --  )   This routine actually does the emulation.
  { dup cmd?
      {doCmd}{emit}ifelse
  }def

/cmd?    %( n -- b )    Looks for command only in topmost dictionary
  { currentdict exch known }def

/doCmd   %as per the topmost dictionary, since doCmd is only invoked
         %if cmd? finds the command in the topmost dict.
         %default top dict:  CmdDict

  {load exec} def

/emit    %( n --  )    convert char code to printable char, and show it.
  { charct MaxCharCt  ge
      { (\\) show CR LF }if
    workstring  dup 0 4 -1 roll put show
    /charct charct 1 add store
  }def

/CmdError    %To handle errors during multi-character command sequences,
             %"emit" is redefined, thanks to dictionary stacking and
```

```
            %late-binding.  See /EscDict later on in this listing.

  { popCmdStack emit }def

/popCmdStack    %clean up dict stack after completion or interruption of
                %multi-character command sequences.
                %CmdDictBase was defined in initjob.
                %CmdDict is never popped off the dict stack.

  { countdictstack CmdDictBase sub  {end}  repeat }def


% Routines to assist page handling ---------------------------------------

/orientpage
  { setlandscape?
      { 8.5 72 mul 0 translate 90 rotate }
      {}
    ifelse
  }def

/workstring 1 string def

/CR    % cr only (no linefeed)

  { LeftMargin currentpoint exch pop moveto
    /charct 0 store
  }def

/LF    % linefeed without cr  --  ff if needed.
       % linect was def'ed in initjob.

  { linect MaxLineCt  lt
      { /linect linect 1 add store
        currentpoint LnHt sub moveto }
      { currentpoint pop FirstLine
        FF  moveto }
      ifelse
  }def

/FF    %showpage and formfeed

  { showpage initpage }def


Article Change History:
08 Feb 1995 - Added PostScript caution.
Adobe Customer Support


Keywords:  <None>

================================================================
```

19960215 11:05:19.00

Tech Info Library Article Number: 1713