The *Fancy Font*™ System: Before and After Example

The example shown on the back of this page depicts the input to the *Fancy Font* printing program (Pfont) and the resulting output. This is a somewhat complicated example as we have included a variety of features for demonstration purposes. It is best to understand in a very general way how *Fancy Font* works and then look at this specific example.

General Description

Fancy Font has been designed to work with any text editor or word processing package that produces an ASCII file as output. Thus *Fancy Font* works with WordStar, Perfect Writer, Mince and many others. This saves you time and money – you need not either purchase or learn to use a new editing program. Using your favorite editor, you create a file containing text to be printed by Pfont. If you wish to take advantage of the Pfont formatting and font selection features, you embed a variety of commands in the text. The example is divided into 3 parts: the text file (labeled THE INPUT FILE) is the file you create with your editor; the second part (labeled THE COMMAND LINE) is the command you issue to tell Pfont to start printing and exactly how that printing is to be accomplished; the final part (labeled THE RESULTING OUTPUT) is the end product – high quality, proportionally spaced, multi–font print.

The Text File

All commands in the text are preceded by the "\" character (this can be changed if you so desire). The first command, "\r", indicates that all subsequent text on the line is to be *right aligned*, that is, printed flush with the right margin (notice the placement of *Demonstration Diskette* in the output). The next line introduces 2 new commands: \c for centering and \f for font selection. The \c indicates that the entire line is to be centered between the margins, again notice the corresponding centered output. \f is the most frequently used command. It is used to specify a font to be used for printing. In the example, all characters are selected from font 0 (the default font) until the \f2 appears. Notice that \f2 causes SoftCraft to be printed in a different font (Old English); the subsequent \f1 switches selects the number 1 font (Roman 18 pt. in this case), and the \f3 selects the *Fancy Font* font. You may switch fonts as often as you like using as many as 10 different fonts in any document.

The next paragraph begins with a \j command to indicate justification. When justification is on (i.e. \j), all lines are printed so that they have an even left and right margin. This is accomplished by increasing the width of all spaces in the line by units of one 120^{th} inch. Justification is turned off (i.e. lines have a ragged right edge) by a \k command (see the middle of the second paragraph). Justification can be temporarily turned off by the \b command. This is useful at the end of paragraphs and elsewhere to allow short lines (e.g. see the end of the first paragraph in the demonstration).

The \u command is used in the first paragraph and elsewhere to both turn on and turn off underlining. The first .\u begins underlining, the second ends the underlined region. The second paragraph contains more centering and font selection commands (\c and \f respectively) as well as introducing a new command: vertical spacing (\v). Notice that super and subscripting is achieved by selecting a super or subscript font (i.e. \f5 for superscript and \f6 for subscripting in the example). Thus super and subscripting are as simple as any other font selection. Font 4 in the example contains several special characters that do not appear on any key on most keyboards. In this case, we chose a labeled key to represent a special character in a particular font. In the example, the character "C" in font 4 represents the copyright symbol.

The vertical spacing command indicates the distance between the bottom of the previous line and the top of the current line (i.e. the amount of white space preceding the current line). The vertical space command is measured in units of printer's points (one 72nd inch). Measure your demonstration output and notice that there are exactly 18 points (1/4 inch) between the line containing the copyright symbol and the following line.

The line beginning \a0120 is one of the more unusual and interesting lines. Here we are using absolute horizontal positioning to overprint a background pattern and normal text. The absolute horizontal motion is measured in units of one 120th inch; thus the command \a0120 positions the print head one inch to the right of the left margin. Following the horizontal positioning, notice several words of text, another \a command and a peculiar sequence of b's and G's. The second \a command repositions the print head to 1 inch from the left margin, \f3 selects font 3, and the b's and G's are special characters in font 3 (the 20 point *Fancy Font* in this example). The "b" character represents a vertical bar (an individual bar is shown at the right side of the line), and the "G" represents a 1/2 inch wide background character (an individual "G" appears at the end of the line). By combining several G's and b's we form the background pattern you see in the output.

Printing the File

We have seen how to construct a text file to be used as input to the *Fancy Font* printing program. The next step is to use the printing program (Pfont). Pfont can be used in a variety of ways to control the printing process. Basically, Pfont accepts a variety of parameters to specify different aspects of the printing operation. The most often used parameters are 1) the name of the file to be printed and 2) the fonts to be used during the printing. In the example (following the line: THE COMMAND LINE ...), the Pfont command is issued with "Demo2.ff" as the name of the file to be printed, "+Fo" to indicate the fonts to be used and a list of fonts (e.g. Romn12). The order in which the font names are listed is very important. This determines the correspondence between font numbers (e.g. \f0) in the text and the actual font to be used. For example, \f2 in the text refers to font "Olde20" which is an Old English, 20 point font. Notice that the fonts are numbered from 0 to 6 in this example. This correspondence between font numbers and names saves typing an entire font name in your text each time you want to change fonts and additionally allows you to change the fonts used to print your file without actually modifying the file (i.e., just change the list of fonts following +Fo).

The final parameter used in this example is "+lw 5.5". This indicates that a Line Width of 5 1/2 inches is to be used when printing. There are many additional parameters which can be specified to control top and bottom margins, headers, footers, page length etc. Each of these has a preset value which you only change if you want to.

Pfont can be used in this "command line" manner, or can be used in 2 other ways depending upon your level of expertise and what you are trying to do. The printing parameters can be entered interactively. That is, Pfont prompts you for parameters, always allows you to ask for help and provides general help or help specific to the parameter you are using; allows you to inspect and change the settings of any parameters; responds to and helps you to correct errors and generally makes it as easy as possible to control printing. Finally, Pfont can be used in a "canned" manner. That is, a file can be created containing Pfont parameter settings. Pfont can then be totally or partially controlled by the settings specified in this *parameter input file*. This is extremely useful for developing settings for different types of printing and then selecting the appropriate parameter input file relevant to the type of printing you are doing at the moment. Don't be confused by all the options provided by Pfont. When you are beginning, you use all the built-in settings and just indicate the name of the file to be printed and the fonts to be used.

THE INPUT FILE - CREATED BY ANY STANDARD EDITOR OR WORD PROCESSOR

```
\rDemonstration Diskette

\f2SoftCraft \f1presents The \f3Fancy FontTM \f1System\f0

\jThe \f4Fancy Font\f0 system provides font sets in a large variety of
styles, sizes and faces with sizes from 8 points to 40 points; styles
including Roman, Sans Serif, Script and Old English; bold, italic and
regular faces.  The package also includes the \uHershey character database\u
containing over 1500 characters and graphic symbols that can be scaled
to different sizes and formed into new font sets.\b

\f5Super\f0script and \f6sub\f0script are available:\b
\cx\f52\f0 + y\f52\f0 = z\f52\f0          CH\f63\f0CH\f62\f0OH
Create any special symbols you need, for example:\k
\c\f4)\f0  Copyright: \f4C\f0  Carriage Return: \f4R\f0  Trademark\f4TM  (\f0
\v0018\c\uThe following demonstrates shading, overprinting
\cand absolute horizontal positioning:\u

\a0120  Use Absolute positioning for overprinting\a0120\f3bGGGGGGGb b G

\f2SoftCraft\f0  8726 S. Sepulveda Bl. Suite 1641  Los Angeles, CA  90045
```

THE COMMAND LINE USED TO PRINT THE FILE WITH FANCY FONT

```
B> Pfont Demo2.ff +Fo Romn12 Romn18 Olde20 ff20 ff12 Romn8p romn8b +lw 5.5
```

THE RESULTING OUTPUT

Demonstration Diskette

SoftCraft presents The *Fancy Font*™ System

The *Fancy Font* system provides font sets in a large variety of styles, sizes and faces with sizes from 8 points to 40 points; styles including Roman, Sans Serif, Script and Old English; bold, italic and regular faces. The package also includes the Hershey character database containing over 1500 characters and graphic symbols that can be scaled to different sizes and formed into new font sets.

Superscript and subscript are available:

$x^2 + y^2 = z^2$ CH_3CH_2OH

Create any special symbols you need, for example:

[•] Copyright: © Carriage Return: ♪ Trademark™ [•]

The following demonstrates shading, overprinting and absolute horizontal positioning:

Use Absolute positioning for overprinting

SoftCraft 8726 S. Sepulveda Bl. Suite 1641 Los Angeles, CA 90045