


The *Fancy Font*[®] System
User's Manual
Edition 2.1

SoftCraft, Inc.
222 State Street
Madison, Wisconsin 53703
(608) 257-3300

This manual was printed by the *Fancy Font* System.

Reduction Scale: 
1 inch

Copyright notice

Copyright © 1984 by SoftCraft, Inc. All rights reserved. No part of this publication may be reproduced without the prior written permission of SoftCraft, Inc., 222 State Street, Madison, Wisconsin 53703.

Trademarks

Fancy Font System is a registered trademark of SoftCraft, Inc. CP/M is a registered trademark of Digital Research, Inc. WordStar is a registered trademark of MicroPro International Corporation.

Table of Contents

	<u>Page</u>
Part I -- Introduction	1
I.1 What to Read	1
I.2 Program Interaction	1
I.3 Font Terms -- Size, Style, Face	2
Part P -- Printing with Fonts	3
P.1 The Text File	6
P.1.1 Terminology	8
P.1.2 The Embedded Commands	10
P.2 Parameter Specification	53
P.2.1 Interactive Mode	53
P.2.2 Command Line Mode	55
P.2.3 Parameter File Input	56
P.2.4 Which Disk Drives to Use/Switching Diskettes	56
P.2.5 Special Characters	57
P.3 Parameters	60
P.3.1 Parameter Description Format	61
P.3.2 The Parameters	62
P.4 Commands While Printing	104
P.5 Error Interaction	104
Part E -- Editing Fonts	107
E.1 Commands	112
E.2 Printing Mechanics	129
E.2.1 Interline Spacing Limitation	132
E.2.2 Baseline and Relative Vertical Font Location	133
E.3 Maximum Character Size	134
Part C -- Creating Fonts	137
C.1 Commands	139
C.2 Batch Mode Operations	154

	A1-1
	A2-1
	A3-1
	A4-1
	A4-1
	A4-16
	A4-23
A5	A5-1
A6	A6-1
A7	A7-1
A8	A8-1
A9	A9-1

Descriptions and Samples

Using *FancyFont* with Word Processors

Error Messages

Pfont Messages

Efont Messages

Cfont Messages

A5 Distribution File Descriptions

A6 Hershey Database

A7 Data File Formats

A8 ASCII Conversions

A9 Parameter and Command Summary

Part I -- Introduction

Part I -- Introduction .

The *Fancy Font* system provides the tools necessary to fully exploit the superb resolution offered by dot matrix printers, such as the Epson line of printers. The system is embodied by three separate programs, the main one being the printing program (Pfont). Most users will be mainly interested in this program. The other two programs allow editing of existing fonts (Efont) and creation of new fonts from a master character table (Cfont).

Pfont achieves high printing quality by passing the print head over each character many times -- six times for standard characters, more for tall characters -- moving the paper as little as a 216th of an inch between each pass. Faster, rough draft modes are obtained by using fewer passes.

I.1 WHAT TO READ

The remainder of the manual is composed of three parts, describing Pfont (part P), Efont (part E) and Cfont (part C). Most users of the *Fancy Font* system will probably be concerned only about printing text files and will not care about editing or creating fonts. If that is the case only part P needs to be read. Parts E and C should be read by those desiring to edit or create fonts. Part E also contains interesting information on the detailed printing method and vertical line spacing limitations. This latter information will be critical to anyone trying to modify a word processor for use with *Fancy Font*, but should not be necessary for occasional users of Efont.

There are numerous useful appendices. The glossary defines terms that may be foreign to casual computer users and also defines terms specific to the world of typesetting and character design. The last appendix gives a summary of the commands and parameters accepted by all three programs of *Fancy Font* and can be consulted during reading of the manual to get a better feeling for the whole of the system structure.

I.2 PROGRAM INTERACTION

The three programs in *Fancy Font* perform different functions and, of necessity, have slightly different user interfaces. Pfont asks the user to specify some parameters and then it prints for an extended period time. In computer science terms we would say it has a declarative nature. Efont and Cfont have more of an interactive command interface and thus have a more procedural nature. All of the programs provide online help, but Pfont in particular provides extensive help (on demand) and has very

Introduction

SIZE, STYLE, FACE

are included along with the *Fancy Font* programs and font sets. A font is generally a collection of characters of a particular size. The sizes of the characters are given in printer's points (approximately one 72nd of an inch). For comparison, typewriter size is generally 10 to 12 points (unrelated to 10 and 12 typewriter mechanisms). Two common font styles are Roman and script and appendix 2 presents a variety of other font styles. Several variations such as italic or bold may be contained within one font family. Subscript and superscript fonts are usually smaller point size fonts appearing lower or higher than other fonts. The glossary contains more detailed definitions of these terms but this short introduction should suffice for reading of subsequent parts of the manual.

Part P -- Printing with Fonts

Part P -- Printing with Fonts

As mentioned in the introduction, Pfont is the primary program in the *Fancy Font* trio of programs. The basic function of Pfont is to format and print text on a dot matrix printer. The text can be provided to Pfont either from a file on disk or interactively from the keyboard. The text file can be printed by typing

```
A>Pfont text.ff
```

The underlined characters are typed by the user, 'A>' is the operating system prompt, and '↵' represents the return (enter) key. Note that this and other examples make use of drive 'A'. Pfont may be used with any disk drive, including a hard disk. In the example above, 'text.ff' is the name of the file to be printed. Executing Pfont in this way will cause the contents of 'text.ff' to be printed using the initial settings for unspecified parameters such as line spacing, line width, font size and style, etc. These settings can easily be changed. For example,

```
A>Pfont text.ff +sp .5i
```

will cause the file to be printed as before but with a half inch of space between successive lines of text. In general, many different parameters can be specified on the command line. Alternatively, parameters may be specified interactively to take advantage of the extensive online help within Pfont. To specify parameters interactively, simply execute Pfont without any command line arguments, e.g.

```
A>Pfont
```

Parameters can then be specified in response to the Pfont prompt, which is composed of two angle brackets. For example,

```
A>Pfont  
>> text.ff +sp .5i  
>> ↵
```

is equivalent to the previous example where the interline spacing was set to a half inch. The extra return ends the interaction and tells Pfont to begin printing.

The user can intermix specification of parameters on the command line and interactive specification; help can be obtained about the parameters at any time.

Part P -- Printing with Fonts

This completes the introduction to the Pfont *printing parameters*. The other aspect of Pfont to be explained is the types of special commands that can appear *in the text file*. That is, different words on a line of text can be printed in different fonts, words can be underlined, and lines can be centered or flushed against the right margin. These actions and many others are accomplished by placing commands in the text file at the location where the action is desired. All commands are of the form: backslash (\) followed by a single letter. For example ‘...\uquality software\u...’ appearing in the text file would cause the words ‘quality’ and ‘software’ to be underlined, as in quality software. The ‘\u’ command is called an underlining "toggle"; i.e., it turns underlining on if it was off and off if it was on. The most commonly used command is probably the ‘\f’ (change font) command. The ‘\f’ is followed by a single digit specifying a new font number. Thus, ‘...\f2Fancy Font\f0...’ would cause ‘Fancy’ and ‘Font’ to be printed in font number 2, as in *Fancy Font*, and the remainder of the text to be printed in font zero; the correspondence between font numbers and the actual font is specified on the command line by the Pfont ‘\fo’ parameter and can be changed with each printing of the text file if desired. (In the case above, font 2 corresponds to the font FF12.fon). The ‘\fo’ parameter is described in detail in section P.3 and the complete list of allowable in-text commands is described in the following section.

How to Read About Pfont

This introduction provides you with the basic ideas behind Pfont -- the capability to select and specify printing parameters and the embedded in-text commands to control the appearance of sections of text.

The remaining sections of part P detail the in-text commands, the modes of interaction with Pfont, and the parameters. They also provide additional information on controlling printing and on understanding and recovering from errors.

All commands and parameters are described in a consistent format to aid understanding and to provide convenient reference.

Which Disk Drives To Use

Fancy Font diskettes are not copy protected and therefore you can rearrange the contents of the diskettes in any manner you choose. If you are using a system with a hard disk, you should copy the *Fancy Font* diskettes to your hard disk. For most floppy diskette based systems, the following configuration is recommended.

Part P -- Printing with Fonts

1. Create a diskette, formatted and containing a copy of your operating system (called the *Pfont* diskette). Copy Pfont and as many frequently used fonts as will fit onto the diskette (copy from the *Fancy Font* diskettes using your system's Copy or PIP facility. To use Pfont, you must include at least the following files on the Pfont diskette:

Fancfont.pro	<i>Fancy Font</i> configuration file
Pfont.hle,Pfont.hlp	Help files (may be omitted)
Pfont.exe	(DOS only) Main Pfont program
Pfont.com	(CP/M only) Main Pfont program
Pfonto0.ovl,Pfonto1.ovl,Pfonto2.ovl,Pfonto3.ovl	(auxilliary Pfont overlays)

2. Create a blank, formatted diskette (called the *text* diskette); copy onto this diskette any special or extra fonts not on the *Pfont* diskette, but needed to print your text.
3. Place your word processing diskette (not part of *Fancy Font*) in disk drive A and your *text* disk in drive B.
4. Use your word processor to create a file on the *text* disk (in drive B). The file should contain text to be printed by *Fancy Font* and any *Fancy Font* embedded commands to change fonts and format that you desire.
5. Remove your word processing diskette from drive A and insert the *Pfont* diskette (created in step 1) in drive A.
6. Now use Pfont indicating the name of the file to be printed, desired fonts and other printing directives. Make sure to precede the names of all text files and font files with 'B:' if they are on the disk in drive B. For example,

A> Pfont B:Text.ff +FO Romn12 B:Olde20

will print the contents of the file 'Text.ff' using the Roman 12 pt. font (found on the *Pfont* diskette) and the Old English 20 pt. font (found on the *text* diskette in drive B).

Part P -- Printing with Fonts

1 THE TEXT FILE

The most common use of Pfont is to print a file containing text. The text can be created with any standard text editor or with most word processing software. The use of Pfont was introduced in the previous sections and is described in detail in sections P.2 and P.3. Pfont can actually be used to print the contents of any ASCII (DOS or CP/M) file, regardless of how the file was created.

It is usually desirable to change fonts and specify style and formatting information in the text file. There is a collection of in-text (embedded) commands that indicate appearance and formatting characteristics of portions of the text to be printed. Each of the commands is preceded by a command indicator to distinguish it from normal text. The built-in command indicator is the backslash character (\), however if this is inconvenient for a particular file, you can use the Command Indicator (CI) parameter (see section P.3) to specify a different command indicator for the entire text file. The command indicator is followed by one or a few characters to specify the details of the requested appearance. The characters following the command indicator may be either upper or lower case.

Thus all embedded commands follow the format:

`<Command Indicator><Command Character><Value>`

Note that the notation '<' name '>' is used here and throughout the manual to describe input to *Fancy Font*. You never type the '<', '>' or the words between these symbols. You, instead, type an item described by this notation. An example of the notation and of the commands described above is,

`\f5`

indicating that subsequent text is to be printed using font 5. The '\ is the <Command Indicator>, 'f' is the <Command Character> and '5' is the <Value>.

Some embedded commands require a value that is a measure of distance. These values must therefore specify the unit (e.g. inches, centimeters, etc.) to be used for the measure. The number specifying distance must be immediately followed by a single character. 'i' represents inches, 'd' printer specific dots, 'c' centimeters, and 'u' a user defined unit.

Part P -- Printing with Fonts

or example,

`\h3.5i`

specifies a positioning of the printhead exactly 3 1/2 inches to the right of the left margin.

Finally, many measures can specify relative as well as absolute positioning. This means that you don't always have to measure from the left margin or top of the page, but can also measure relative to the current print position. This is achieved by inserting a '+' or '-' before the value. For example:

`\h+2c` indicates to move the printhead forward 2 centimeters, and

`\h-1.5i` indicates to move the printhead backward 1 1/2 inches.

The complete form of an embedded command with a measured value is:

`<Command Indicator><Command Character>
<+ or - option><Value><Unit>`

`<Command Indicator>` is '\'

`<Command Character>` is the name of a command, e.g. 'h' in '\h+3.5i'

`<+ or - option>` is '+' (measure forward from current print location)

'-' (measure backward from current print location)

or **nothing** (measure from the top of page or left margin depending upon the command)

`<Value>` a number, possibly with a fractional part (e.g. .4, 5.72, 3)

`<Unit>` 'i' inches, 'c' centimeters, 'd' horizontal or vertical printer specific dots, 'u' user defined unit (see UU parameter, section P.3). The MX version of *Fancy Font* is designed to print 120 dots per inch horizontally while the FX version prints 240 dots per inch horizontally. Both versions use a vertical dot unit of 72 dots per inch (although printing is actually 216 dots per vertical inch).

Note: in all cases, you must *never type a space* or any other extra character inside of an embedded command. CP/M users note that there is a 2 digit accuracy limitation in the fractional part of measured values and that conversion of centimeters to dots is not exact.

The embedded commands are described in the following pages. Each

Part P -- Printing with Fonts

command is described in terms of its purpose, its syntax (as in the example above), how to use the command (usage), and at least one example of use. Additionally, the possible values and errors are detailed. Finally, a set of notes and suggestions for use are included. The syntax line contains the word `break` at its right edge if the embedded command causes an implicit break. Several commands cause such a break in order to allow more natural separation between paragraphs, headings, indented sections etc. An *implicit break* means that a `\b` is *automatically* processed (by Pfont) before such a command is processed. Implicit breaks only apply when using word wrap – see `\w` and `\b` embedded commands in section P.1.2.

Three *short* commands have been added for convenience and compatibility with other software. These commands are ‘Control-H’ (see `\-`), ‘Control-I’ (see `\t`) and ‘Control-L’ (see `\p`).

P.1.1 Terminology

You will need to know a few words and symbols used in describing the Pfont commands and parameters. A quick summary follows, a more complete terminology is described in the glossary of this manual.

`< . . . >` Brackets are used to describe a value. You do not actually type any words contained in the brackets nor should you type the brackets. (e.g. `<list of real numbers between 0 and 150 inches>`, indicates a list of numbers – you might type ‘1.1 3 4.675’).

list One or more values; the values are separated by at least 1 space (e.g. `5 10.5 12 36` or `File1 File2 File3`).

String A sequence of characters. Any characters may appear in a string, but if a space character is in a string, the entire string must be surrounded by double quotes to mark its beginning and end (and to distinguish it from a list). For example, `"This is 1 string"`, is a string. If you wish to have the `"` character appear within a string, you must type the quote twice. Example, `"This ""string"" contains quotes"`. The paired quotes are undoubled when the string is used, thus the example yields:

This "string" contains quotes

Integer A whole number, no fractional part (e.g. `1, 3, 12, 0, 5`). There are no negative numbers in Pfont parameters (although negative numbers may be used within embedded commands).

Part P -- Printing with Fonts

Digit A single digit, usually between 0 and 9 inclusive.

Real Number A number that may include a fractional part (e.g. 1.5, .7, 0.2, 6, 12.8).

Horizontal Measure A number that may include a fractional part and must be immediately followed by a unit (e.g. 1.5i, .7c, 0.2i, 6d). A Horizontal Measure indicates a distance across the page, usually between 0 and the current or maximum line width (8 or 13.6 inches depending upon printer). It always specifies a distance either from the left margin or from the current print position.

Vertical Measure A number that may include a fractional part and must be immediately followed by a unit (e.g. 1.5i, .7c, 0.2i, 6d). A vertical measure indicates a distance up and down the page and is usually a number between the current vertical print head position and 425 inches. For printers that can move the paper backwards, the Vertical Measure can result in a position between 0 and 425 inches from the top of a page. It always specifies a distance either from the top of the page or from the current vertical print position.

File Name The name of a file using standard DOS (or CP/M) conventions (e.g. B:Name.Ext). If the file name contains Pfont special characters (see below), the name must be enclosed in quotes. The 'B:' prefix indicates the name of the disk drive containing the file to be used. If files are on other than the currently logged disk drive, such a prefix *must* be used.

Font File Name The name of a file containing the characters of a font. Font file names usually have an extension of '.fon' as in 'Romn12.fon'. The '.fon' is optional when input to Pfont: it will be provided if it is missing, otherwise this is a standard DOS (or CP/M) file name (see File Name). For example, if 'Romn12.fon' is on drive B, you only need to type 'b:romn12' (note that case is not significant, that is, characters can be a mixture of upper and lower case).

Page Number The number of a printed page. This is usually an integer between 1 and 9999.

Pfont Special Characters Several characters have a special meaning when parameters are indicated to Pfont. If you want to use one of these characters, but not its special meaning, the character must appear in a quoted string (e.g. "?" or "A & B"). The special characters are:

Part P -- Printing with Fonts

? & " @ < + - ,

See section P.2.5 for a description of each.

P.1.2 The Embedded Commands

The descriptions of the embedded commands follow in alphabetical order on the ensuing pages. The commands, grouped according to function, are:

Font Selection

font selection
draft mode
underline
continuous underline

b break
c center
j justify
k justify off
r right align
w word wrap on
x word wrap off
> left indent
< right indent

Format Control

Horizontal Motion

h horizontal motion
i short horizontal
m horizontal mark
t tab
- backup 1 character

Substitution

Ascii character
string substitute

v vertical motion
l line spacing
n mark vertical position
p new page
q conditional new page

Vertical Motion

Miscellaneous

: reverse field
! comment

Part P -- Printing with Fonts

`\a` -- Ascii Character

Ascii Character -- `\a`

Specify an ASCII character by its decimal equivalent.

Syntax `\a<Integer between 000 and 127>`

Usage: The `\a` command will be replaced by a single character that is the ASCII equivalent of the indicated decimal number. See appendix A8 for a table of decimal-ASCII equivalents.

Values: `<Integer between 000 and 127>`
The value must be exactly 3 digits long (e.g. 000, 122). The value indicates an ASCII character by its decimal equivalent and must be between 0 and 127, inclusive.

Examples: The A character is: `\a065`.
Produces: The A character is: A.

Errors: If there are no digits following the `\a` command, or if the digits represent a number larger than 127, an error message is displayed. If less than 3 digits are specified, these digits are processed as if there were leading zeroes and a warning message is displayed.

Notes:

- `\a` can be used to avoid interpretation of special characters - for example: if Control-I, Control-L, Control-H are specified by `\a009`, `\a012`, `\a008` respectively, then their special *Fancy Font* meanings will not be used. (See `\t`, `\p`, and `\-`).
- `\a000` must be used if you need to include a 'null' character in your file. You cannot enter a null character in your file without using `\a000` as Pfont uses the true null character to indicate an end of an input line and will therefore ignore all remaining characters.
- `\a026` must be used to include a 'Control-Z' (EOF) character in your file as 'Control-Z' is otherwise treated as an end of file marker.

Suggestions: Use `\a` to include characters (such as control characters) in a text file that are otherwise difficult to enter using your text editor or on the command line.

Part P -- Printing with Fonts

\b -- Break

Break -- \b

Temporarily end justify and word wrap.

Syntax

\b

break

Usage: When using word wrap (\w), the \b command indicates the end of a print line. When \b is encountered, a line will be printed whether or not it is complete. Such a line will not be justified. See \w for more discussion of breaks and word wrap.

When using justification but not word wrap, \b indicates that the line is not to be justified. In this case, the \b may be placed anywhere in an input line.

Values: none

Examples:

- \jThis line is not justified, notice the right margin.\b
Produces: This line is not justified, notice the right margin.
- \wThis will print\b as two lines using word wrap.
Produces: This will print
as two lines using word wrap.

Errors: none

Notes:

- Several commands (\v, \p, \q, \>, ...) are implicitly preceded by a \b command. Likewise, a blank line or a line beginning with a space or tab character will be assumed to be preceded by a \b. These implicit breaks are for convenience and ease of text entry and are exactly analogous to the \b command.
- \b\b\b or \b\b\b... will behave exactly as one \b.

Suggestions: Use \b to override justification and word wrap for special lines within groups of lines that are normally justified or word wrapped. See \k and \x for turning off justification or word wrap for groups of lines.

Part P -- Printing with Fonts

`\c` -- Center

Center -- `\c`

Adjust text so that it is centered on the print line.

Syntax

`\c`

Usage:

The `\c` embedded command specifies text to be centered and the horizontal region of the page in which it is to be centered.

Any text following the `\c` command up to the first tab (`\t`), horizontal position command (`\h`), right alignment command (`\r`) or end of line, will be centered. If the end of line or the right alignment command are encountered, the text will be centered over the entire line (i.e. from the left margin for the width of the line minus any left or right indentation -- see LM and LW parameters, `\>` and `\<` commands). If a tab or horizontal position command (`\t` or `\h`) is encountered, the centering region will start at the position of the `\c` command and end at the horizontal position that the tab specifies.

If `\c` is used in word wrap mode, when an end of line is encountered on any line that contains a centering command, that end of line will force a break. The centered text will be printed and subsequent input will be printed on subsequent output lines.

Centering is performed with respect to the left margin, line width and current left and right indentation settings. Thus, a modification to indentation will modify the centering position.

Values:

none

Examples:

- `\cCentered Title`

Produces: Centered Title

- `\h.125i|\ccentered between tabs\h3i|\ccolumn 2\h6i|`

Produces: | centered between tabs | column 2 |

- `Left\cCentered\rRight`

Produces: Left Centered Right

In the first example the centered text is ended by a `\r` (end of line). Thus it is centered over the entire line width. The second example shows two centered strings, one between .125 and 3 inches, the other between 3 and 6 inches. If the final horizontal position

Part P -- Printing with Fonts

(\hfi) were not specified, 'column 2' would be centered over the entire line (causing overprinting). The final example divides the line into 3 sections: left, centered and right aligned. The centered text is centered over the entire line since it is ended by a right align command (\r).

Errors:

- A center command cannot be specified within another center command (i.e., before a tab or end of line); an error message will be displayed and the second center command will be ignored.
- Centering will not work properly if the centering region specified is too small for the string to be centered, or if the end of the region is a horizontal position less than the beginning of the region.

Notes:

- Centering can result in overprinting when the region is ended with \r or end of line.
- Commands that force breaks (e.g. \b, \v, \p) will serve to mark the end of input to be centered. Because of this, be certain to place \c *after* \v if both need to be used (for example, \vli\c..... not \c\vli.....).
- When using word wrap, be careful to know the contents of the line to be centered. You may want to start the line with a \b (or an implicit break).

Suggestions:

Use in conjunction with HL and FL parameters to create centered header and footer lines. Use a combination of tabs, horizontal positioning and center commands to create labelled tables.

Part P -- Printing with Fonts

\d -- Draft Printing

Draft Printing -- \d

Specify horizontal rough drafting modes.

Syntax

\d<Integer between 0 and 3>

break

Usage:

Place the `\d` command at the beginning of any print line to indicate that line and all subsequent lines are to be printed in a new horizontal draft mode. Horizontal draft modes represent the maximum number of dots per printed horizontal inch and therefore control the quality and speed of printing.

Values:

- **<Integer between 0 and 3>** (FX version)

For `+RD 0` to `+RD 3` (vertical resolution 216 dpi), the modes are:

- 0 240 dpi -- best quality, 2 full passes per line (odd/even dots) -- same as `+RD 0` (which is same as `-RD`)
- 1 240 dpi -- good quality, 1 pass per line, consecutive dots may not be printed -- same as `+RD 1`
- 2 120 dpi -- 25,000 dots/square inch quality -- same as `+RD 2`
- 3 120 dpi -- rough draft quality, consecutive dots may not be printed -- same as `+RD 3`

For `+RD 4` to `+RD 6` (vertical resolution 72 dpi), the modes are:

- 0 120 dpi -- good quality, 1 pass per line -- same as `+RD 4`
- 1 120 dpi -- rough draft quality, consecutive dots may not be printed -- same as `+RD 5`
- 2 60 dpi -- same as `+RD 6`

- **<Integer between 0 and 1>** (MX version)

For `+RD 0`, the modes are: (vertical resolution 216dpi)

- 0 120 dpi -- best quality -- same as `+RD 0` (`-RD`)
- 1 60 dpi -- rough draft quality -- no corresponding `+RD` mode

For `+RD 1` and `+RD 2` (vertical resolution 72dpi), the modes are:

- 0 120 dpi -- good quality -- same as `+RD 1`
- 1 60 dpi -- *sandpaper* quality -- same as `+RD 2`

Part P -- Printing with Fonts

Examples:

(using +RD 0, FX version)

- \d0Draft mode 0, normal printing.

Produces: Draft mode 0, normal printing.

- \d1Draft mode 1.

Produces: Draft mode 1.

- \d2Draft mode 2.

Produces: Draft mode 2.

- \d3Draft mode 3.

Produces: Draft mode 3.

Notice that the word "Produces" is also printed in the draft mode. Draft mode can only be set by print line, it cannot be changed within one line.

Errors:

If an improper draft mode is selected, the mode will not be changed and a warning message will be displayed. If a change of draft mode is attempted anywhere on a line, then the entire line will be printed in the last draft mode specified on the line.

Notes:

- \d is preceded by an implicit break since an entire line must be printed in one draft mode.
- \d can only affect the *horizontal* draft modes since the fonts are loaded into memory using a specific vertical draft mode *before* Pfont starts to print.

Suggestions:

Use the draft mode command to print different lines at different dot densities to increase speed and/or create special appearances. Try printing larger fonts at higher draft modes and thus higher speeds and smaller fonts using the best quality modes.

Part P -- Printing with Fonts

\f -- Font Selection

Font Selection -- \f

Specify the font to be used for all printing until the next \f command.

Syntax

\f<Digit> or \fp or \ff or \fn

Usage:

Place the \f command anywhere in your text file. All subsequent text until the next \f will be printed in the font represented by the symbol following \f.

- \f<digit> indicates the number of the font to use — see FO parameter and the table below. The <digit> must be between 0 and 9 for CP/M users but MSDOS users may specify 'a', 'b', 'c', 'd', 'e' for fonts 10 thru 15 when using Fancy fonts.
- \fp returns to the previous font selection (i.e., the numbered font in effect before the previous \f command). Font 0 is considered the previous font if there have been no previous font changes.
- \ff indicates that the high-quality, fancy fonts, are to be used (this is the normal, initial setting).
- \fn indicates that fonts native to the printer are to be used. (See the list below for details). Fancy fonts and native printer fonts can be combined in the same document, but not on the same print line.

Values:

- <Digit>
For *Fancy Font* printing, the digit between 0 and 9 corresponds to a font specified using the +FO parameter. MSDOS users may specify 'a', 'b', 'c', 'd', 'e' for fonts 10 thru 15. Font 0 is the first font specified by +FO, font 1 the second, etc. That is, if the following command line is used:

```
Pfont Text.ff +FO Romn12 Romn14,
```

then, '\f0' corresponds to 'Romn12' and '\f1' corresponds to 'Romn14'. Pfont initially uses font 0 until changed by a \f command.

For printing using native printer fonts (i.e. \fn), the font numbers select from a different set of fonts. The

Part P -- Printing with Fonts

correspondence of numbers to fonts for the Epson MX, FX, RX and the IBM Graphics printer is: (this correspondence can *not* be changed)

<u>Font Number</u>	<u>Epson MX,FX or IBM Font</u>
0	Normal
1	Compressed
2	Expanded
3	Compressed/Expanded
4	Normal Italics
5	Compressed Italics
6	Expanded Italics
7	Compressed/Expanded Italics
8	Normal Emphasized
9	Expanded Emphasized

- **p**
Select the previous font (font 0 if no previous font).
- **f or n**
Switch between high quality, Fancy Fonts (f) and Native printer fonts (n). The current font number remains in effect when switching between high quality and native fonts.

Examples: (using Pfont parameters: +FO Romn12 Romnb12)

- `\f0This is \fiwonderful weather\fp.`

Produces: This is wonderful weather.

- `\fn\f0This is \fiwonderful weather\fp.`

Produces: This is wonderful weather.

Where \f0 refers to Romn12 and \f1 refers to Romnb12. \fp, in this case, is the same as \f0. Notice how \fn switches to lower quality, native printer fonts. Remember to switch back to Fancy Fonts with \ff at the beginning of the next line. Either \ff or \fn must be used for an entire line.

Errors:

- The digit following `\f` must correspond to a font specified with the +FO parameter (if not using native printer fonts). If there is no correspondence, a message will be displayed and no font change will take place. Most fonts define a similar set of characters (alphabet, numbers, punctuation, etc.).
- If you select a character for printing that is not defined in the current font (see Appendix A2), nothing will be printed. An error message is not displayed in this event,

Part P -- Printing with Fonts

and the resulting printing will have several missing characters.

- `\fn` and `\ff` must be the first commands processed on any new print line. If either of these commands is issued at other than the beginning of the line, the entire line will be printed in the type of fonts specified last on the line, horizontal and vertical formatting will not work properly, and a warning message will be displayed. When using word wrap, it is recommended that you precede `\ff` or `\fn` with `\b` to force a line break.

If, when using `\fn`, a font number is selected that does not correspond to a `\ff` font number available for use, then a subsequent `\ff` command will display a warning message automatically select font 0.

Notes:

- When you change fonts, you change the definition of all characters including spaces. Thus, the width of a space may vary depending on the current font. For example,

```
\f0This is\f1 a test.\f0
```

may be spaced differently than:

```
\f0This is \f1a test.\f0
```

- `f`, `n` and `p` may be upper or lower case characters.
- The amount of vertical space occupied by a printed line depends upon the tallest and deepest fonts used on the line. `Pfont` attempts to maintain uniform spacing between lines, but cannot always do so (see section E.2). When the requested interline spacing is not possible, a message is displayed at the end of all printing. On printers that allow reverse paper motion, *Fancy Font* will remove this extra vertical space by rolling back the paper. (See `+RV` parameter).
- The height of a blank line is determined by the current font. The placement of the `\f` command determines the spacing. For example, headlines in large fonts should be typed as `'\f1Headline\f0'` rather than putting the `\f0` on a subsequent line. See `\v` for finer control of vertical white space.
- Native printer fonts and Fancy Fonts cannot be used

Part P -- Printing with Fonts

within the same print line.

- \: used in conjunction with native printer fonts produces 'double-strike' print.

Suggestions:

- Use \f commands throughout text to print with multiple fonts for emphasis, super and sub-scripting, section headings, special symbols, etc.

- Use \fp in substitution strings when font switching is to be done and then returned to the font in effect before the substitution string was used. For example, to create a string to print "*Fancy Font*":

+SU "\f3Fancy Font\fp"

would produce *Fancy Font*, when \s0 was used, and then switch back to the font in effect before the substitution string.

- Use a combination of \ff and \fn to print titles or letterheads with high quality Fancy Fonts and the remaining sections of your documents with high speed, native printer fonts.

Part P -- Printing with Fonts

\h -- Horizontal Motion

Horizontal Motion -- \h

Move the print head forward or backward on the current print line.

Syntax `\h<+ or - option><Horizontal Measure> or \hm<Digit>`

Usage:

The `\h` command can be placed anywhere within text to position the print head. It can be used to move forward or backward, to cause overprinting or to position outside of an indented printing region. The positioning can be absolute with respect to the Left Margin (LM, not indent), or can be relative to the current placement of the print head. Instead of specifying a Horizontal Measure, `\hm<Digit>` can be used to position the print head at a previously defined mark (see `\m`).

Values:

- **<Horizontal Measure>**

The Horizontal Measure must evaluate to a distance between 0 (left margin) and the current line width (see LW parameter). This allows for positioning on the entire print line, including outside of indented areas of text. All values must include a specification of a unit (i,c,d,u) immediately at the end of the number (e.g. `\h3.4i`). Values may be preceded by either a '-' or '+' to indicate motion backward or forward relative to the current print position. Absolute measures are based on the left margin (independent of indentation). The print head may be positioned within an indentation area (that is, to the left of the normal printing area) to form outdented lists and other special effects. Positioning past a right indented region will cause a warning message to be displayed if not using word wrap, but can be used. Positioning past the right indented area is not recommended when using word wrap - this may result in the generation of a new line.

- **m<Digit>**

The digit must be between 0 and 9 and must have been previously defined by a `\m<Digit>` embedded command. The print head will move to the referenced position on the line.

Examples:

- `Start\h2ileft margin + 2 inches.`
Produces: Start left margin + 2 inches.
- `Start here\h+1.75iand leave a 1 3/4 inch gap.`
Produces: Start here and leave a 1 3/4 inch gap.

Part P -- Printing with Fonts

- This is a \m0marked line, see the line-up?
\hm0X marks the spot!

Produces: This is a marked line, see the line-up?
X marks the spot!

Errors:

- Attempt to specify a horizontal measure outside of the current printing range (determined by +LW and +LM settings) will display an error message and will cause no change of the print head position. If the position is within the right indentation region (see \<), a warning message will be displayed, but the print head will be positioned as requested. Positioning within the left indentation region (see \>) is ok.
- The \h command can be used to cause characters to be overprinted. If this occurs, and is not intended, it is likely that a horizontal position less than the current print head position, was specified.

Notes:

- If horizontal motion is used in conjunction with justification (\j), then justification will begin after the last horizontal motion command on a line. Use \i to not affect justification in this way.
- A horizontal motion command at the beginning of a line does *not* cause a word wrap break, so if word wrap is on and you want to line up some text under a mark point on another line of text, then the line containing the mark point should end with \b.

Suggestions:

- \h can be used in place of a tab and thus allows the simulation of multiple 'tab' stops within a text file.
- Use \h to create outdented lists. That is, you can explicitly place text within the indent region using \h after setting an indent with \> or \<.

Part P -- Printing with Fonts

`\i` -- Relative Horizontal Motion

Relative Horizontal Motion -- `\i`

Specify short horizontal motion, forward or backward from the current print head position.

Syntax

`\i<+ or -><Horizontal Measure>`

Usage:

The `\i` command will cause the print head to move forward or backward the specified number of units. If no '+' or '-' is indicated, a '+' is assumed. `\i` differs from `\h` since it is relative only, limited to a short distance, and most important, it does not affect justification nor does it serve to end a region such as those used for centering and right aligning. (See `\c` and `\r`).

Values:

<Horizontal Measure>

The value specifies the distance to move the print head forward or backward. It can be specified in any available unit, but must result in a number of printer dots between 0 and 255 (1 to 2 inches depending upon the printer).

Examples:

- There is 1 inch between |'s: `|\-\i1i|`.
Produces: There is 1 inch between |'s: | |.
- A 2 and 1/2 inch long blank line: `_\i1i\i1i\i.5i_`.
Produces: A 2 and 1/2 inch long blank line: _____

Errors:

none

Notes:

This works with only approximate accuracy for native font printing.

Suggestions:

- If you wish to squeeze slightly more characters on a line but do not want to adjust the line width, you may replace a space character with a `\i` command to reduce the size of the space character.
- Use `\i` instead of `\h` for short horizontal moves whenever `\h` would cause undesired side effects, such as when used on a justified line or within centered and right aligned strings. See `\h` command.
- `\i` can be used in conjunction with continuous underline (`_`) to create horizontal lines as in the example above. Since `\i` is limited to short movement, you can combine several `\i` commands to achieve the desired length.

Part P -- Printing with Fonts

\j -- Justification On

Justification On -- \j

Turn on justification. Each space character in the printed line will be expanded so that the right edge of the text is even with the right margin.

Syntax

\j

Usage: Place the \j command anywhere in your text file. All subsequent text (including the line in which \j appears) will be justified until turned off by \k, or temporarily overridden by \b, \c or \r.

Values: none

Examples: \jThis line is justified, notice the right margin.
Produces: This line is justified, notice the right margin.

Errors: none

- Notes:**
- Justification is initially off.
 - Only spaces following the last tab or \h command in a line are expanded to provide justification. Therefore the \h command affects justification, but the \i command does not. Also, leading spaces on a print line are *not* justified.
 - \b, \r (right align) and \c (center) temporarily override justification for the line on which they appear. Justification will again be in effect on the next print line.
 - When using word wrap, justification is temporarily overridden by the above commands, as well as by implicit breaks, such as \v, \p, \>, lines beginning with a space or tab character, and blank lines.
 - Lines which require spaces wider than approximately 1 inch each for proper justification will not be completely justified (justification would not look good in these cases).
 - Lines which exceed the maximum line width will not be justified.
 - Pfont does not do letterspacing. That is, it does not add space between the letters of a word, but only between words. If letterspacing is absolutely necessary, use the \i

Part P -- Printing with Fonts

command to explicitly preform the letterspacing.

- Suggestions:**
- A combination of justification (`\j`) and word wrap (`\w`) produces nicely aligned text. If very long words are combined with short lines, justification may leave larger than desired gaps. In this case, you may want to insert your own hyphenation to improve the appearance of your printed document, or choose not to use justification.
 - All implicit breaks, and of course the explicit `\b` command, force a temporary end to justification when word wrap is enabled. This allows you to easily print paragraphs and other sections of documents without needing to explicitly turn on and off justification. When not using word wrap, `\b` only, temporarily ends justification.

Part P -- Printing with Fonts

\k -- Justification Off

Justification Off -- \k

Turn off justification.

Syntax

`\k`

Usage: Place the `\k` command anywhere in your text file. No subsequent text will be justified until a `\j` command is encountered. The line in which the `\k` appears will *not* be justified.

Values: none

Examples: `\kThis line is not justified, notice the right margin.`
Produces: This line is not justified, notice the right margin.

Errors: none

Notes: This command has no effect on word wrap.

Suggestions: Use a combination of `\j`, `\b` and `\k` embedded commands to control justification.

Part P -- Printing with Fonts

\l -- Line Spacing

Line Spacing -- \l

Set the interline spacing.

Syntax

`\l<+ or - option><Vertical Measure>`

break

Usage: The line on which the `\l` command is placed will be printed preceded by the indicated amount of vertical white space.

Values: <Vertical Measure between -425 and 425 inches>
The amount of vertical white space to appear before the current print line and all subsequent lines. The value must be a real number followed by a unit specification. The value may be preceded by a plus or minus to add to, or remove from the current setting of line spacing. A vertical measure without + or - will set the line spacing to exactly the indicated value. The initial line spacing is set to the value of the +SP parameter.

The measure is the distance from the bottom of one line to the top of the next. The bottom of a line is the location of the lowest font on the line independent of the characters actually typed on the line.

Examples: This is the first line.
`\l.5i`This line is preceded by 1/2 inch of white space.

Produces: This is the first line.

This line is preceded by 1/2 inch of white space.

Errors: Line spacing greater than the current page length (see +PL parameter) will force a new page between each line. This would normally be undesirable, and would likely be the result of an error.

Notes:

- The interline spacing begins with the setting specified by the +SP parameter. The `\l` command performs the same function as +SP, however it can be used multiple times within a document to alter the interline spacing.
- The standard interline spacing is .045 inches (3 dots on Epson MX and FX type printers).
- `\l` causes a break when used in word wrap mode.

Part P -- Printing with Fonts

Suggestions: Use \l to set off special sections of text. For example, a quotation may be printed indented and with less than normal interline spacing.

Part P -- Printing with Fonts

`\m` -- Horizontal Mark

Horizontal Mark -- `\m`

Mark the current horizontal position of the print head.

Syntax

`\m<Digit>`

Usage: Place the `\m` command anywhere in your text to remember the horizontal position at which the next character will be printed. Up to 10 horizontal marks can be set and used in a document. Each of the marks can be reset as often as desired.

Values: <Digit between 0 and 9>
A number between 0 and 9 used to reference the mark. Thus, 10 marks are possible. See the `\h` command to use the marks set by the `\m` command.

Examples: `\m0This is a well \m1marked line\m2.`

`\hm1↑Mark 1\hm2↑Mark 2\hm0↑Mark 0.`

Produces: This is a well marked line.

↑Mark 0. ↑Mark 1 ↑Mark 2

Errors: Missing or non-digit values will cause the command to be ignored and a warning message displayed.

Notes:

- The position at which the next character would be printed is marked – this is considered the current print head position.

- See `\h` command to make use of marked positions.

Suggestions:

- *Fancy Font* is a typesetting system as well as a printing and word processing package. Most of the fonts are proportional and characters printed on successive lines normally do not line up with one another. The marking facility is therefore useful in lining up text.

- The `\m` command is useful in lining up columns of text and or numbers. As depicted in the example above, one or more positions in a line may be marked and then used in subsequent lines. Try this command for lining up numbers, for overprinting, etc.

Part P -- Printing with Fonts

`\n` -- Vertical Mark

Vertical Mark -- `\n`

Mark the current vertical position from the top of the page.

Syntax

`\n<Digit>`

Usage:

Place the `\n` command anywhere in your text to remember the vertical distance from the top of the current page to the current print head location. Up to 10 vertical marks can be set and used in a document. Each of the marks can be reset as often as desired.

Values:

<Digit between 0 and 9>

A number between 0 and 9 used to reference the mark. Thus, 10 marks are possible. See the `\v` command to use the marks set by the `\n` command.

Examples:

- `\n3`The vertical position of this line is marked.
The position of the top of this line from the top of the page is marked and can be referenced by `\vm3`.
- `\w\j\<3.25i\n0`This is an example of printing two columns of text for an insert within a page of otherwise one large column of text. It is also possible to print entire pages with two or more columns, however you are responsible for performing explicit pagination, headers and footers.
`\vm0\<0i\>5.25i`This is the second column. See how it lines up with the first column due to the use of a combination of vertical marking (`\n`) and vertical motion (`\v`). Also notice the use of the left and right indent commands (`\>`, `\<`) to determine the width of the columns. `\<0i\>0i`

Produces:

This is an example of printing two columns of text for an insert within a page of otherwise one large column of text. It is also possible to print entire pages with two or more columns, however you are responsible for performing explicit pagination, headers and footers.

This is the second column. See how it lines up with the first column due to the use of a combination of vertical marking (`\n`) and vertical motion (`\v`). Also notice the use of the left and right indent commands (`\>`, `\<`) to determine the width of the columns.

Errors:

none

Part P -- Printing with Fonts

Notes:

- Negative vertical motion (reverse paper feed) is only available for printers that provide this feature (e.g., Epson FX80).
- The vertical position is marked from the top of the page to the top of the inkable area for the current print line.
- \n marks a vertical position from the top of a page. This position will be marked for all pages. The position is not a combination of page and vertical position, it is position only.

Suggestions:

- Useful in printing multiple columns with printers that can reverse paper feed. As shown in the example above, mark the start of the first line of the first column on a page or in a table, then return to that mark to print each subsequent column.
- Can be used to line up text at the exact same vertical location on many different pages; as used in multiple, overlay transparencies.

Part P -- Printing with Fonts

`\p` -- New Page

New Page -- `\p`

Begin printing on a new page.

Syntax	<code>\p</code>	break
---------------	-----------------	--------------

Usage: The `\p` command causes the current page to be ended (footer will be printed) and the next page to be printed (starting with the header). If not using word wrap, regardless of where the `\p` appears in a line, the *entire* line will appear on the new page. Using word wrap, `\p` causes an implicit break and thus ends the current print line.

Values: none

Examples: `\p`This line is on a new page.

This is hard to demonstrate on a single page. Pfont ends the current page, starts a new page and prints: This line is on a new page. on the first line of the new page.

Errors: none

- Notes:**
- The Process Form Feeds (PF) parameter does not affect `\p`.
 - Control-L is a synonym for `\p`, but it is affected by the PF parameter. See PF parameter to ignore Control-L characters. Use `\a` to make use of Control-L as a printable character (or use `-PF`).
 - `\p` is implicitly preceded by a break.

Suggestions: Use `\p` to assist in generating desired pagination; e.g., forcing chapters and figures to begin on a new page, thus avoiding page breaks in the middle of figures. See `\q` for conditional new page.

Part P --- Printing with Fonts

\q --- Conditional New Page

Conditional New Page --- \p

Begin printing on a new page if the indicated vertical space is not available on the current page.

Syntax	<code>\q<Vertical Measure></code>	break
---------------	---	--------------

Usage: Indicate a vertical measure. Pfont will determine whether the amount of space you indicated is still remaining between the current vertical print head position and the top of the bottom margin area of the page. If the space remains, the \q command will have no effect (except causing an implicit break); if the space is not available on the current page, the \q command will cause the footer of the current page to be printed and then will begin subsequent printing on a new page.

Values: <Vertical Measure between 0 and 425 inches>
A real number indicating the amount of space that must be remaining on the current page if the page is not to be ejected. The number must be ended with a unit.

Examples: `\q2.5i`
\p if less than 2 1/2 inches of vertical space left in printing area of page, otherwise nothing.

Errors: none

Notes:

- \q command forces an implicit break in word wrap mode regardless of whether a page is ejected.
- \q with a Vertical Measure larger than the current page length is equivalent to \p (since there will never be the requested amount of space left on the page).

Suggestions: Use \q command to ensure that new sections of text will start on a new page if there is not sufficient room for them on the current page.

Part P -- Printing with Fonts

`\r` -- Right Alignment

Right Alignment -- `\r`

Align a string of text so that it is flush with the right margin or a specified horizontal position.

Syntax

```
\r<any text><new line, break, \h or tab>
```

Usage:

All characters to the right of the `\r` command and preceding the next new line character, tab or horizontal positioning command will be printed so that they are flush right with the right margin (in the case of new line) or with the horizontal position specified by `\h` or `\h`.

Values:

`<any text><new line, break, \h or tab>`

Follow the `\r` by any text. The region is ended by the end of the line (new line character), by any break, implicit or explicit, by a horizontal position command (`\h`) or a tab.

Examples:

- Left of page `\rright` aligned.
Produces: Left of page right aligned.
- `\rAligned at 5 inches.\h5i`
Produces: Aligned at 5 inches.
- `\r42.\h3i72`
`\r5.\h3i2974`
Produces: 42.72
5.2974

Errors:

Centering and right align within a right align statement are prohibited; an error message will be displayed, and they will otherwise be ignored.

Notes:

If using word wrap (`\w`), the text from the `\r` command to the next `\b`, new line, or implicit break command (e.g. `\v`, `\p`, `\q`) is considered to be the text to be right aligned.

Suggestions:

- `\r` is useful in formatting header lines (HL) and footer lines (FL). The command is also useful in lining up columns of variable width numbers as shown in the 3rd example above.
- Consider using a fixed width font instead of `\r` for creating tables of aligned columns.

Part P -- Printing with Fonts

\s -- Substitute

Substitute -- \s

Insert the current page number, file name or a pre-specified string in the text.

Syntax `\s#` or `\sf` or `\s<Digit>`

Usage:

The `\s` command will be replaced in the text by a string of characters determined by the character following `\s`. The substitution takes place before justification or any other processing of embedded commands.

Values:

- `#`
If the character following the `\s` command is `#`, then the entire command (`\s#`) will be replaced by the number of the current page.
- `f`
If the character is `f`, then the command (`\sf`) will be replaced by the full name of the file currently being printed.
- `<Digit>`
If the character is an integer between 0 and 9 and there is a corresponding substitution string specified (see Substitution +SU), then the entire command will be replaced by that substitution string.

Examples:

- `This is page \s#, file \sf.`
Produces: This is page 35, file fmanp2b.man.
- (using Pfont parameters: +SU "String 1" "page is \s#")
Substitution occurs here: \s1 and here: \s0.
Produces: Substitution occurs here: page is 35 and here: String 1

Errors:

- Substitutions can be indicated within substitution strings (see second example above). It is possible that a substitution string attempts to substitute itself (recursive substitution). A line length exceeded error message will be displayed and subsequent program behavior will be unpredictable. Immediately exit the program using Control-Z in this case (CP/M, use Control-C).
- If the number of a nonexistent substitution string is used, an error message is displayed and the substitution request is ignored.

Part P -- Printing with Fonts

Notes:

- When not using word wrap, be careful that lines do not exceed the line width *after* substitution occurs.
- The 'f' in '\sf' may appear as either an upper or lower case character.
- Any embedded commands may appear in a substitution string and will be processed after substitution.

Suggestions:

The embedded command \s# is essential in printing the page number in the header or footer line. See HL and FL for details. Substitution strings can be used to fill in variable fields in form letters (one line at a time), to include frequently typed words or to represent a sequence of other embedded commands. When defining a substitute string (using +SU) that changes fonts, consider using \fp rather than a specific font number to switch back to the font that was in effect before the substitute string was used.

Part P -- Printing with Fonts

`\t` -- Tab

Tab -- `\t`

Horizontal motion to the next tab stop.

Syntax

`\t`

Usage: Whenever the `\t` command (or its Control-I synonym) is recognized in the text, an attempt will be made to find the next available tab stop (see `Tabs` parameter). If found, the print head will be positioned at the tab stop, otherwise the tab will be ignored.

Values: none

Examples: (using `Pfont` parameter: `+TB 4i`)

Tab coming `\t` first stop.

Produces: Tab coming first stop.

The word first appears 4 inches from the left margin.

Errors: If there is no remaining tab to the right of the current print position, an error message is displayed and the tab command is ignored.

- Notes:**
- `\t` and Control-I (the normal "Tab" key on your keyboard) are synonyms and can be used interchangeably (with the exception of parameter strings such as `+HL` and `+SU`, where `\t` must be used).
 - Tabs can only move the print head forward (towards the right). See `\h` or `\i` for backwards horizontal motion.
 - Tab stops cannot be changed within a document; use `\h` or `\i` instead.
 - All tab stops are relative to the left margin, not the left indent location.
 - A tab issued while to the left of an indented left margin will cause the print head to move directly to the indented left margin — regardless of any tab stops between the current position and the indent.
 - Left indent does *not* change tab settings in any way other than described in the preceding note.
 - Tab stops work similarly to those used by mechanical

Part P -- Printing with Fonts

typewriters; the current position of the print head and the tab stop settings will determine the extent of the tab. Different sized characters and different numbers of spaces may produce different results.

- Control-I or `\t` at the start of an input line will force an implicit break when using word wrap. In this use only, they are equivalent to `\b\t` and `\bControl-I`.
- Tabs affect justification. Only spaces following the last tab on a line will be used to adjust the length of a justified line.

Suggestions:

Use a combination of tabs and horizontal motion (`\h`, `\i`) to help line up text in columns and tables.

Part P -- Printing with Fonts

1 -- Underlining

Underlining -- \u

Turn on and off underlining of text.

Syntax

\u

Usage:

Place the \u command anywhere in your text file. All subsequent text will be underlined until another underline command (\u or _). Thus, \u is a toggle. Each time it appears in text underlining will be turned off if it was on, or on if it was off.

The continuous underlining command (_) is similar to \u, however it will underline white space, whereas \u will underline only characters. (See _ embedded command).

Values:

none

Examples:

- Some of this is \uunderlined text\u, some is not.

Produces: Some of this is underlined text, some is not.

- _Fully underlined including spaces.\u

Produces: Fully underlined including spaces.

Errors:

- Make sure to turn *off* underlining before switching between fancy fonts and native printer fonts (see \ff and n); extraneous lines may result.
- When using native printer fonts (see \fn) if underlining remains on between printed lines, the left margin will be underlined.

Notes:

- White space is not underlined for Fancy Font fonts, only characters with ASCII codes other than 32 are underlined (see example above). Continuous underlining, including spaces, can be achieved by using the \u command as shown by the second example above.
- Notice that \u can be turned off by another \u or by _. Therefore, \u and _ cannot be used at the same time.

Suggestions:

Don't forget to turn off underlining (by using another \u or _). If you forget, the remainder of your text will be underlined.

Part P -- Printing with Fonts

`\v` -- Vertical Motion

Vertical Motion -- `\v`

Specify the amount of white space to precede the current line, or movement of the paper to begin printing the specified distance from the top for the page.

Syntax `\v<+ or - option><Vertical Measure>` or `\vm<digit>`

Usage:

To position the print head a specified distance from the top of the page, use `\v` with an absolute number of units. For example, `\v4i` will begin printing the top of the current line 4 inches from the top of the page. The use of a relative positive measure (e.g. `\v+.25i`) will leave exactly the specified amount of white space between the bottom of the previous print line and the top of the current print line. Negative distances (relative to the *current* location of the print head, are only possible with printers that allow backwards paper motion.

`\vm<digit>` will position the top of the current print line to the indicated position, however this is only usable for printers that allow reverse paper motion (such as Epson FX80).

Values:

- `<Vertical Measure between 0 and 425 inches>`
The value can be absolute or relative, but must be less than 425 inches when evaluated.
- `m<digit>`
Move directly to the position marked by the indicated vertical mark (`\n<digit>`).

Examples:

There is 1/2 inch of white space between this line
`\v+.5i` and this line.

Produces: There is 1/2 inch of white space between this line

and this line.

Errors:

An attempt to move the paper outside of the possible range of motion will cause the `\v` command to be ignored and an error message to be displayed.

Notes:

- If `\v` appears on the first line of a printed page it will be ignored unless the new page was explicitly requested by a `\p` command, a Control-L character, or if it is the first

Part P -- Printing with Fonts

page of a document.

- `\v` can cause a new page to be printed. In this case any unused portion of the distance specified with `\v` will be discarded and `\v` is identical to `\p`.
- Remember that interline spacing refers to the distance between the bottom of one line (including descenders) and the top of the next. Whether or not the actual printed line contains descenders or ascenders doesn't matter; it's the overall height of the *font* that matters.
- Vertical motion to a mark set on a previous page will position the print head the previously marked amount down from the top of the *current* page.

- Suggestions:**
- Use the `\v` command to temporarily override the value of the Interline Spacing parameter (SP). The command is useful for specifying the exact amount of white space to be left before paragraphs, sections, examples and in tables. You may leave a blank line in your text to achieve vertical white space. `\v` provides this same capability, but with greater control over the amount of white space.
 - Remember to specify `\v+...` when leaving extra vertical space between lines; without the '+' an absolute position results.

Part P -- Printing with Fonts

\w -- Word Wrap On

Word Wrap On -- \w

Enable moving of words between lines to automatically fit line length.

Syntax

\w

Usage:

Place the `\w` command anywhere in your text. The line on which `\w` occurs and all subsequent lines will be word wrapped. That is, the end of line in your text input will be largely ignored. Pfont will read words from your input text and fit as many words as possible on each output line.

There are a variety of ways to stop word wrap or to temporarily 'override' this feature. `\x` will turn word wrap off (the built-in setting). `\b`, any implicit break (e.g., `\v`, `\p`, `\q`), an input line beginning with a space or a tab, or a blank line will cause the *previous line* not to be word wrapped. Thus, paragraphs can be separated by a blank lines and/or the first line of a paragraph can be indented to automatically create the desired effect between successive paragraphs. This provides a natural means of typing text to be formatted and printed.

Values:

none

Examples:

`\w`These lines are to be combined
notice how they print in a different manner than they appear
in the input text.

Produces: These lines are to be combined notice how they print in
a different manner than they appear in the input text.

Errors:

Improper 'long line splits' may occur if very long words (longer than approximately 20 characters) are frequently used without using new lines periodically within paragraphs (every 200 or so characters). This is extremely rare, since words are usually shorter than 20 characters.

Notes:

- Pfont behaves rather differently when word wrap is enabled. The implicit breaks become operational for both word wrap and for justification. This is a significant difference and allows text to be entered without the use of any special commands; specifically `\b` is not often required. There are no implicit breaks when not using word wrap.

Part P -- Printing with Fonts

- Word wrap will adjust lines to be the length of the current margins and indentation settings. When indentation is changed, word wrap automatically reflects the new effective line length.
 - New line characters are normally turned into single spaces; however, if immediately preceded by '.', '?' or '!', new line characters are turned into double spaces to separate sentences.
 - Pfont removes all trailing spaces from lines input from files. In this way, no groups of extra spaces should mistakenly appear in output. However, if you place more than one space between words within a line, the extra spaces will not be removed by Pfont.
 - You can type entire paragraphs, sections or an entire document without using any new line characters to mark the end of input lines.
 - Lines will be broken for wrapping immediately before a space (or new line) character.
- suggestions:
- It is recommended that `\w` be issued at the start of a document or the start of a sequence of normal paragraphs.
 - `\w` may be used with or without justification (`\j`). Justification is independently enabled and disabled by the `\j` and `\k` commands respectively.

Part P -- Printing with Fonts

\x -- Word Wrap Off

Word Wrap Off -- \x

Do not move words between lines (the built-in setting)

Syntax

\x

Usage:

The line on which `\x` is encountered and all subsequent output lines will be printed as they appear in the input text. That is, line endings will appear exactly as they are in the input text. Lines that are too long to fit within the margins will be chopped off and only the portion fitting on the output line will be printed.

Additionally, implicit breaks will no longer serve to temporarily end justification – the explicit `\b` break must be used.

Values:

none

Examples:

`\x`This line is not word wrapped.

The previous line is very short as is this one.

Produces: This line is not word wrapped.

The previous line is very short as is this one.

Errors:

'Line length exceeded' messages will be displayed for all lines that are longer than the current line width and indentation. These lines will be truncated if using *Fancy Font's* (`\ff`) or 'wrapped' as to overlap themselves if using native printer fonts (`\fn`).

Notes:

This is identical to the formatting mode available in *Fancy Font* versions prior to version 2.0.

Suggestions:

It is useful to turn off word wrap when printing tables, letter salutations or other groups of text where 'as is' appearance is desired.

Part P -- Printing with Fonts

-- Reverse Field Printing

Reverse Field Printing -- \:

Toggle between normal printing (ink forming characters) and reverse printing in which a black background is printed with non-black areas appearing as printed characters.

Syntax `\:`

Usage: Place `\:` anywhere in your text. All subsequent printing will appear in the reverse field until another `\:` command is encountered in the text. Normally, printing will be black (or ink color) on white (or paper color). For the Epson MX,RX and FX printers, if a native printer font is in use, reverse field will enable and disable double strike printing.

Values: none

Examples:

- (for *Fancy Font* fonts)
`\f4This \:is reverse\:` and now normal.

Produces: This **IS REVERSE** and now normal.

- (for native printer fonts)
`\fnThis \:is reverse\:` and now normal.

Produces: This is reverse and now normal.

Errors:

- This doesn't work particularly well for *Fancy Font* fonts. The reverse field characters tend to be overwhelmed by the reverse field itself. It works best for large, thick or bold characters.

- Different height characters produce different height backgrounds.

Notes: Try using this with rough draft mode for slightly better appearance.

Suggestions: This can be used to produce thick, dark lines or perhaps with special characters as a visual separator. You are pretty much on your own with this one, however; let us know if you find a good application.

Part P -- Printing with Fonts

\- -- Backup Character

Backup Character -- \-

Cause the print head to move backwards by the width of the last printed character.

Syntax

Usage:

Place \- anywhere in your text to cause two successive characters to be overprinted. When \- is encountered, the print head will be moved backwards by exactly the width of the most recently printed character.

Values:

none

Examples:

One way to print `resum\-'` or `Helm^\-ut`.

Produces: One way to print `resumè` or `Helmût`.

Errors:

none

Notes:

- The 'Control-H' character embedded in text can be used as a synonym for the \- command. Depending upon the text editing program you are using, it may be difficult to actually insert a 'Control-H' character in your text, however.
- (Notice that \- is the 'minus' sign).

Suggestions:

- Use this for overprinting single characters for accents as shown above and for simultaneous super- and sub-scripts. (Either order of characters is possible). See also the \i, \h, \m commands and the use of negative character margins in Efont.
- Use \i to finely adjust overprinting of characters having different widths.

Part P -- Printing with Fonts

`_` -- Continuous Underlining

Continuous Underlining -- `_`

Turn on and off continuous underlining of text.

Syntax

`_`

Usage:

Place the `_` command anywhere in your text file. All subsequent text will be underlined until another underline command (`_` or `\u`). Thus, `_` is a toggle. Each time it appears in text underlining will be turned off if it was on, or on if it was off.

The `\u` underlining command is similar to `_`, however it will underline only characters, whereas `_` will underline spaces. (See `\u` embedded command).

Values:

none

Examples:

- `_Fully underlined including spaces._`
Produces: Fully underlined including spaces.
- `Some of this is \uunderlined text\u, some is not.`
Produces: Some of this is underlined text, some is not.

Errors:

- Make sure to turn *off* underlining before switching between fancy fonts and native printer fonts (see `\ff` and `n`); extraneous lines may result.
- When using native printer fonts (see `\fn`) if underlining remains on between printed lines, the left margin will be underlined.

Notes:

- `_` will underline spaces and relative horizontal motion forward (`\i+..`). Any white space created by tabs or the `\h` command will *not* be underlined.
- Notice that `_` can be turned off by another `_` or by `\u`. Therefore, `_` and `\u` cannot be used at the same time.
- Notice that `_` is the 'underscore' key.

Suggestions:

- Use continuous underlining in conjunction with the `\i` command to create horizontal lines of any desired length. Since `\i` is limited in the amount of space it can leave, use multiple `\i` commands to create as long a horizontal line as is desired.

Part P -- Printing with Fonts

- Don't forget to turn off underlining (by using another `\` or `\u`). If you forget, the remainder of your text will be underlined.

Part P -- Printing with Fonts

> -- Left Indent

Left Indent -- \>

indent the specified number of units from the left margin; effectively setting new left margin for all subsequent printing.

Syntax \><+ or - option><Horizontal Measure> **break**

Usage: \> causes all normally printed lines to be indented by the specified amount. Notice that the direction of the 'arrow' points in the initial direction in which the indentation occurs.

Values: <Horizontal Measure>
A number preceded by '+' will add the indicated amount to the indentation, '-' will subtract and no '+' or '-' causes the indentation to be set exactly the specified amount from the left margin. The evaluated Horizontal Measure (that is, the final position on the line) must be between 0 inches and the current line width minus the current right indent, (i.e. LW minus \<). See LW parameter and \< embedded command.

Examples: \>1iNotice the indentation for this line and all subsequent lines until the left indentation\>0iis again changed.

Produces:

Notice the indentation for this line and all subsequent lines until the left indentation is again changed.

Errors: Indent past the line width (LW parameter) is not allowed; an error message will be displayed and the command ignored.

Notes:

- A tab (\t or Control-I) processed with the print head positioned to the left of the indented area will cause the print head to move directly to the indented area, regardless of any the absence or presence of tab stops. This is useful for outdented lists (for example, \>2i\h0i3.\t...\>0i).
- Indentation causes an implicit break.

Suggestions: Use \> to alter the left margin within your text. Useful for indenting quotes and other special sections of text.

\< -- Right Indent

Right Indent -- \>

Indent the specified number of units from the right margin; effectively setting a new right margin for all subsequent printing.

Syntax

\<<+ or - option><Horizontal Measure>

Usage:

\< causes all normally printed lines to be indented by the specified amount. Notice that the direction of the 'arrow' points in the initial direction in which the indentation occurs.

Values:

<Horizontal Measure>

A number preceded by '+' will add the indicated amount to the indentation, '-' will subtract and no '+' or '-' causes the indentation to be set exactly the specified amount from the right margin. The evaluated Horizontal Measure (that is, the final position on the line) must be between 0 inches and the current line width minus the current left indent, i.e. LW minus \>. In other words, the Horizontal measure must specify a position to the right of the left indent location.

Examples:

\<3iNotice the indentation for this line and all subsequent lines until the right indentation \<0iis again changed. See the long line after the change of indentation!

Produces:

Notice the indentation for this line and all subsequent lines until the right indentation is again changed. See the long line after the change of indentation!

Errors:

Right indentation past the current left indentation (\>) or left margin (LM parameter) is not allowed. A warning message will be displayed and the command ignored.

Notes:

- An indentation command causes an implicit break.
- With some effort, \< and \> commands can be combined with the backwards vertical motion (when provided by printer) to print multiple columns.

Part P -- Printing with Fonts

Suggestions:

Use \< to alter the right margin within your text.
Useful for indenting quotes and other special text.

Part P -- Printing with Fonts

\! -- Comment

Comment -- \!

Insert a notation (comment) in your text.

Syntax **\!<any characters><new line>**

Usage: Place \! anywhere in your text. All subsequent characters on the input line will be totally ignored (that is, not printed). The new line at the end of the input line stops the commenting feature.

Values: <any characters><new line>
Any and all characters may follow the \! command. All characters up to, but not including the new line, will be ignored.

Examples: This is printed.\!This is neither printed nor \f4processed.
Produces: This is printed.

Errors: none

Notes: The new line at the end of the comment is treated as a new line character if word wrap is off (\x), it is treated as a space character if word wrap is on (\w).

Suggestions: Use the comment command to include notes to yourself or others, in your text file. For example, it may be useful to put a comment at the beginning of a text file to indicate the date of creation, the intended fonts, the name of the person creating the file, etc.

Part P -- Printing with Fonts

P.2 PARAMETER SPECIFICATION

The Pfont program is designed to be adaptable to many different applications. This adaptability is controlled by a number of 'parameters'. Each parameter may be used to control some aspect of printing such as line width, page length, fonts, etc. In general, parameters are designed to have normal initial settings so you don't need to specify each parameter each time you use Pfont.

You will need to provide the name of the file you want to print and probably will also want to indicate the fonts to be used. We call these 'parameters' to Pfont.

Different users have different needs at different times and many users have different approaches to using computers. Therefore, we have designed Pfont to provide 3 methods of interaction. You may select to use any one or a combination of these methods each time you use Pfont. The methods are: 1) **Interactive**— parameters are specified one or more at a time, help may be requested at any time; 2) **Command Line**— all parameters are specified on the DOS (or CP/M) command line; and 3) **Parameter Input File**— input is obtained from a prepared file containing parameter specifications (it is intended that this file be reused for multiple "runs" of Pfont).

New users should be most interested in the interactive method, frequent and experienced users should learn how to use the parameter input files and command line methods.

Regardless of the method of interaction, you can specify any number of parameters using upper or lower case or any combination thereof. If you specify the same parameter multiple times, only the last specification of the parameter will be used. If the parameter specifies a list (such as +FO Romn12 Romnb12 ...), each time the parameter is respecified the old list is discarded. Thus, if you try to specify +LW (line width) twice, only the second specification will be used. Likewise, using +FO (fonts) a second time will completely negate the first +FO.

When learning or uncertain about the use of Pfont, it is best to experiment. Make frequent use of the help facilities (?, &) and feel secure in that Pfont performs no destructive operations.

P.2.1 Interactive Mode

Pfont is designed to assist in its use; this is particularly true in the interactive mode. To use Pfont interactively, start by typing the following:

Part P -- Printing with Fonts

\! -- Comment

Comment -- \!

Insert a notation (comment) in your text.

Syntax **\!<any characters><new line>**

Usage: Place \! anywhere in your text. All subsequent characters on the input line will be totally ignored (that is, not printed). The new line at the end of the input line stops the commenting feature.

Values: <any characters><new line>
Any and all characters may follow the \! command. All characters up to, but not including the new line, will be ignored.

Examples: This is printed.\!This is neither printed nor \f4processed.
Produces: This is printed.

Errors: none

Notes: The new line at the end of the comment is treated as a new line character if word wrap is off (\x), it is treated as a space character if word wrap is on (\w).

Suggestions: Use the comment command to include notes to yourself or others, in your text file. For example, it may be useful to put a comment at the beginning of a text file to indicate the date of creation, the intended fonts, the name of the person creating the file, etc.

Part P -- Printing with Fonts

P.2 PARAMETER SPECIFICATION

The Pfont program is designed to be adaptable to many different applications. This adaptability is controlled by a number of 'parameters'. Each parameter may be used to control some aspect of printing such as line width, page length, fonts, etc. In general, parameters are designed to have normal initial settings so you don't need to specify each parameter each time you use Pfont.

You will need to provide the name of the file you want to print and probably will also want to indicate the fonts to be used. We call these 'parameters' to Pfont.

Different users have different needs at different times and many users have different approaches to using computers. Therefore, we have designed Pfont to provide 3 methods of interaction. You may select to use any one or a combination of these methods each time you use Pfont. The methods are: 1) **Interactive**— parameters are specified one or more at a time, help may be requested at any time; 2) **Command Line**— all parameters are specified on the DOS (or CP/M) command line; and 3) **Parameter Input File**— input is obtained from a prepared file containing parameter specifications (it is intended that this file be reused for multiple "runs" of Pfont).

New users should be most interested in the interactive method, frequent and experienced users should learn how to use the parameter input files and command line methods.

Regardless of the method of interaction, you can specify any number of parameters using upper or lower case or any combination thereof. If you specify the same parameter multiple times, only the last specification of the parameter will be used. If the parameter specifies a list (such as +FO Romn12 Romnb12 ...), each time the parameter is respecified the old list is discarded. Thus, if you try to specify +LW (line width) twice, only the second specification will be used. Likewise, using +FO (fonts) a second time will completely negate the first +FO.

When learning or uncertain about the use of Pfont, it is best to experiment. Make frequent use of the help facilities (?, &) and feel secure in that Pfont performs no destructive operations.

P.2.1 Interactive Mode

Pfont is designed to assist in its use; this is particularly true in the interactive mode. To use Pfont interactively, start by typing the following:

Part P -- Printing with Fonts

B> Pfont

Pfont will respond with its usual greeting message as well as reminding you to indicate parameters such as font names and a file name and that '?' may be typed *at any time* for assistance. Pfont prompts with ">>" and awaits input. You may enter as many or as few parameters as you like and may also make use of the special characters '?', '&', '@' (or '<') and 'Control-V' (see section P.2.5).

In the following transcript of Pfont use, user input is underlined, ↵ is 'return' and commentary is *in small italics*.

B> Pfont

Start Pfont.

Pfont: Print with Fonts, ...

Pfont identification and version. This message

Copyright (C) 1984 SoftCraft, Inc.

overwrites a temporary "loading" message.

Enter printing parameters, for example:

+FO Romn12 RomnB12 +FI File.ff

Then press RETURN (possibly more than once) to start printing.

Type '?' for assistance at any time.

>> ?

Request for assistance.

Expecting a Parameter Specification - a '+' or '-' followed by

One of:

Pfont is expecting any of the following

FI (Filenames)

FO (Fonts)

...

Pfont lists many other options

Pfont will indicate "-- more --" after the number of lines that fit on your screen are displayed, press any key to continue.

>> +rd?

Select the 'rd' parameter, then ask for help.

Expecting value for RD (Rough Draft) - a Integer Between 0 and 6

Current Value: <Off>

Pfont indicates the current setting for Rough Draft (Off) and possible settings.

>> RD (Rough Draft)> 1

User selects rough draft mode 1.

>> +fi Sample.FF

User indicates that file Sample.FF is to be printed (+FI is optional here).

FI (Filenames)> ↵

↵ indicates no more file names.

>> +fo Romn12 RomnB12 Romni12

User selects 3 fonts.

FO (Fonts)> ↵

↵ indicates no more fonts.

>> &

Request summary of current settings.

...

Pfont lists the settings of all parameters.

>> ↵

A final ↵ ends interaction and starts printing.

Loading Romn12.Fon as font number 0 . . . in memory

Loading RomnB12.Fon as font number 1 . . . in memory

Loading Romni12.Fon as font number 2 . . . in memory

The 3 fonts are now ready for use.

Make sure the printer is 'On Line'

Reminder to get the printer ready.

Printing Sample.FF

Pfont will now start printing.

Page 1, 42 Lines

Pfont has finished printing the first page.

B>

All done.

Part P -- Printing with Fonts

Notice that return (\backslash) is used to end a parameter specification and *also* to end interaction and begin prompting. The *ending* function of \backslash only applies when it is the only input on a line. When you type \backslash at the end of a sequence of characters, it allows you to continue specifying the same parameter. If this seems confusing or if you ever feel *stuck* while using Pfont, simply type a couple of \backslash 's and Pfont will either start printing or display a message indicating additional parameters that must be specified.

P.2.2 Command Line Mode

Pfont can be used without any interaction by specifying all of its parameters on the DOS (or CP/M) command line. For example,

```
B> Pfont +fi Sample.FF +fo Romn12 RomnB12 RomnI12
```

will print the contents of the file Sample.FF using 3 Roman fonts. The maximum length of the parameter specification list is 128 characters. For longer parameter specification lists, use interactive mode or a parameter input file. File and font names can be typed in any combination of upper and lower case. CP/M users note: CP/M converts all command line input to upper case, so parameters that have case sensitive values should be entered either interactively or using a parameter input file. Pfont modifies the CP/M convention somewhat; it converts the first character of each string to upper case and converts all others to lower case. For example:

```
input:      Pfont test +hl "CHAPTER TWO" +fl "end of page"  
result:     Pfont Test +Hl "Chapter two" +Fl "End of page"
```

To avoid this capitalization, use the parameter file input method described in section P.2.3.

Any of the special characters '?', ',', '&', '<', '@' etc. can be used on the command line, '@' and ',' are of particular use (see section P.2.5).

If you make an error during command line interaction you do not have to retype the line. Pfont retains the portion of the command line following the error, begins interactive mode, and displays an error message and several options of correcting the error and continuing. Often you need only retype the word causing the error; Pfont will continue with the rest of your command after the error has been repaired. See section P.5 for details.

Part P -- Printing with Fonts

P.2.3 Parameter File Input

As you become an experienced Pfont user, you will find that you often wish to print a letter or document multiple times or to print similar letters or documents using the same settings for Pfont parameters. Pfont's parameter file input capability makes this possible. A parameter file is simply a file containing parameter specifications exactly as you would type them on the command line or interactively. To request parameter specifications that have been stored in a parameter file, you need only type:

@file name

Where *file name* is the name of the parameter input file (names with the extension '.ffi' are used by convention, e.g. Sample.ffi). The *@file name* can be included anywhere on the command line or be used at any time in the interactive mode. (CP/M users may type '<' in place of '@' if desired). To create your own parameter files, use your editor to create a file containing parameter specifications; the parameter file may contain one or more lines. One or more parameter input files can be used for any "run" of Pfont. Parameter input files can contain additional requests for parameter input (i.e. *@file name*). However, any parameters specified in the original parameter input file *after* the '@' will be ignored. For most uses, if a parameter input request is used within a parameter input file, the request should be the last item in the file. *Note that parameter input files should never contain blank lines.* Such a line is treated as a command to start printing ('Return') and thus will prevent subsequent lines from being processed.

It is recommended that you create a separate parameter input file for each type of printing you normally do (e.g. letters, mailing labels, disk labels, etc.). A parameter input file for labels might contain the following:

```
+FO Romn12 Olds18
+LW 2.5i +PL 1i +TM .125i +BM .125i
-LM
```

P.2.4 Which Disk Drives To Use/Switching Diskettes

Pfont requires that each of the files Pfont.exe, FancFont.pro, Pfont.hlp, Pfont.hle (CP/M Pfont.com, PfontO0.ovl, PfontO1.ovl, PfontO2.ovl, PfontO3.ovl, Pfont.hlp, Pfont.hle and FancFont.pro) be resident on one of the system disk drives. It is simplest to put all of these files on one disk and use Pfont while "logged" to the drive in which that disk is mounted. However, Pfont will find its overlays (PfontO0, PfontO1, PfontO2, PfontO3, FancFont.pro) if they are either on the logged disk or are on a disk mounted in disk drive 'A'. Pfont.hle and Pfont.hlp are not absolutely

Part P -- Printing with Fonts

necessary — they are 'help' files and may be deleted if necessary. If not available on the working disk, then no help will be available when recovering from Pfont interaction errors and the Control-V general help command will provide no help. Otherwise, the contents of the *Fancy Font* system diskettes may be arranged in any way you desire, including copying to a 'hard' disk.

For DOS systems with limited disk capacity (one diskette drive), the Pfont program may be invoked (for example, by typing Pfont) and then the Pfont disk removed and a disk containing all fonts and files to be printed inserted in the diskette drive. Type the appropriate parameters and begin printing as usual.

For CP/M systems with a small disk storage capacity, such as the Osborne 1, it may be necessary to switch diskettes in order to print long documents with multiple fonts. The following procedure is recommended:

1. Put all fonts to be used on one diskette (the **font** diskette).
2. Place the **font** diskette in drive B.
3. Place the file or files containing text on one diskette (the **text** diskette).
4. Insert the **Pfont** diskette in drive A.
5. Invoke Pfont using interactive mode.
6. Remove **Pfont** diskette and insert **text** diskette in drive A.
7. Interactively (or with parameter input file) specify Pfont printing parameters.
8. Remove **text** diskette and insert **Pfont** diskette in drive A.
9. Type final ↵ (carriage return) to start printing.
10. Pfont will prompt for disk change; following prompt, remove Pfont diskette and insert **text** diskette in drive A.
11. Printing should start, no further disk changes are required.

P.2.5 Special Characters

These special characters are recognized and acted upon by Pfont in any interaction mode, they are primarily used in the interactive mode, however.

'?' *Always* provides assistance. '?' may be used at any time during command line or interactive modes of Pfont use. The response to a '?' assistance request is information pertinent to the current usage of Pfont. In interactive mode, pressing '?' will produce a list of all available parameters if issued following the '>>' prompt. If pressed following a parameter name (e.g. +Rd ?), a

Part P -- Printing with Fonts

description of acceptable parameter values as well as the current value for the parameter will be displayed. For example:

```
>> +Rd ?
Expecting value for RD (Rough Draft) - ...
Current Value: <Off>
RD (Rough Draft)>

>> +Fo Romn12 RomnI12?
Expecting FO (Fonts) list element - a Font File Name
Current Value:
  0: Romn12.Fon
  1: RomnI12.Fon
FO (Fonts)>
```

'?' may appear anywhere on the command line following Pfont. It will cause the interactive mode to be entered and Pfont will provide assistance. The most common command line use, however, is:

iPfont ?

This provides general assistance on using Pfont and displays a list of all parameters and their default settings.

'&' Display the current settings for all parameters. This is a good way to review any parameters you have specified and to determine the initial settings for all other parameters. & may be issued interactively or on the command line. For example:

```
>> &
FI (FileNames) - List of up to 15 File Names
Current Value:
  0:Sample.ff
FO (Fonts) - List of up to 10 Font File Names
...
```

';' Explicitly begins the interactive mode. Use this at the end of the command line if you have additional features you wish to specify interactively. For example:

```
B>Pfont @Test.ffi ,
Pfont: Print with Fonts, version x.x (Epson FX)
Copyright (C) 1984 SoftCraft, Inc.
>>
```

'@' Obtain parameter specifications from the parameter input file. Continue processing any remaining command or interactive input

Part P -- Printing with Fonts

line parameters after input from the parameter input file. '@' may be used multiple times on the command line and/or interactively. See section P.2.3 for more information. CP/M and DOS 1.1 users may substitute '<' for '@', all users may use '<' or '@' everywhere other than on the command line. For example:

```
B>Pfont @Control.ffi ,↵
```

will read the file Control.ffi as though its contents were typed on the command line and then Pfont will enter interactive mode (because of the comma) to allow further specification of parameters or override parameters specified in Control.ffi.

'Return' (↵) Ends input — it can be used to exit the interactive parameter specification, to end a list or to end a parameter if you decide it is not the one you wanted (useful after using ?, for example: '+rd ?' followed by ↵ leaves 'rd' as it was). 'Return' is ignored if it is not the first key typed on an input line and thus can be used to help in typing long lines.

'Control-V' Display on the terminal a description of the special characters (this can only be used interactively). The file 'Pfont.hlp' must be on the logged disk.

'Control-Z' Exit Pfont, ignore all parameter settings and do not print. Pfont will ask for a confirmation of a Control-Z request. Control-Z may be pressed whenever you are using Pfont in the interactive mode. (CP/M users must use 'Control-C' in place of 'Control-Z'.)

'Back Space' (normally Control-H) Backs up one character at a time to correct typographical errors. The back space character and its displayed action can be set in the profile file (FancFont.Pro — see appendix A5).

'Control-U' Delete an entire input line. This line delete character is analogous to the DOS (or CP/M) line delete function and is operational in interactive mode.

Part P -- Printing with Fonts

2.3 PARAMETERS

Font offers a variety of parameters (features) to control the appearance, formatting and printing of a document. Most of the parameters have preset (default) values and hence need not be specified during each use of Font. However, you will want to use these parameters at different times to fully exploit the capabilities of the *Fancy Font* system. Each parameter controls some aspect of the formatting and printing of text. A parameter may be turned *off* or *on* or left at its normal (default) setting.

This section describes each of the parameters in detail, discussing the common aspects of using the parameters and then the method by which the parameters are described in the manual. Each parameter specification has the format:

<On/Off switch> <Parameter Name> <Parameter Value>

The On/Off switch is used to indicate whether the parameter is to be used or not. '+' indicates On and '-' indicates Off. The name indicates which parameter is being specified and the values detail the use of the parameter (some parameters do not require values). Many parameter values require a unit (i.e. inches, dots, centimeters) indicated by a single character (i,d,u) immediately at the end of the value. For example,

+TM 1.5i

Specifies that the top margin parameter be used and that the top margin be 1.5 inches.

-TM

Specifies no top margin (this is exactly the same as +TM 0i). Some parameters have no values, they are either on or off. For example:

+PP

Specifies that single pieces of paper are being used (the page pause parameter is on), and

-PP

Specifies continuous form paper is in use (do not use the page pause parameter). Finally, some parameters can have a value that is a list of fonts:

+FO SanS10 SanS12 Romn12

Specifies on the *Fancy Font* parameter (+FO) and indicates that each of the 3 fonts listed is to be used. As before,

-FO

Specifies that no *Fancy Fonts* are to be used (the parameter is *off*).

Parameter names may be specified in either upper or lower case or some

Part P -- Printing with Fonts

ombination of both. There must always be at least one space between the end of the parameter name and the parameter value.

3.1 Parameter Description Format

The rest of section P.3 describes each of the parameters, each parameter on a new page. The top line on the page indicates the descriptive name of the parameter. The second line gives a very general description of the meaning of the parameter. Then the *syntax* of the parameter is presented. The *syntax* indicates how to type the parameter, either turning it on or off, whether it requires any values and what those values are. At the right edge of the *syntax* line, the default (normal) use of the parameter is indicated. If you do not specify the parameter, the default for that parameter will be used.

The *usage* indicates how to use the parameter; first how to turn the parameter on, then how to turn the parameter off and the meaning of each of these. The *values* section provides details about the parameter's values, describing acceptable and unacceptable values and the meaning of each. The final 3 sections provide *examples*, *notes* and *suggestions*. Each of these sections will help clarify the use and meaning of the parameter. Notes provide the most detailed information and suggestions provide ideas as to how to best use the parameter.

Section P.1.1 describes some particular terminology used in describing Pfont printing parameters.

Part P -- Printing with Fonts

P.3.1 The Parameters

The parameter descriptions follow in alphabetical order on the ensuing pages. The parameters, grouped according to function, are:

Font Selection

FO font selection
RD draft mode
SD screen display

Page Setup

HL header line
FL footer line
PN page number

Horizontal Format

TB tab stops
LW line width
LM left margin

Print Control

NF first page special
PP page pause
PF process form feeds
FP select first page
LP select last page
NC number copies

Vertical Format

HM header margin
FM footer margin
TM top margin
BM bottom margin
PL page length
PG pagination enable
RV reverse paper feed
SP interline spacing

Miscellaneous

SU string substitution
FI file names
CF concatenate files
IS initialization string
UU user defined units
CI command indicator

Part P -- Printing with Fonts

BM -- Bottom Margin

Bottom Margin -- BM

Specify the amount of white space to appear after the bottom of the main body of text on a page.

Syntax `+BM <Vertical Measure> or -BM` [default: +BM .75i]

Usage:

- `+BM <Vertical Measure>`
The amount of white space to place at the bottom of each page. This is the distance from the actual bottom of the page to the bottom of the last line of text on the page (other than the footer line).
- `-BM`
Do not leave any white space at the bottom of each page, no bottom margin (same as `+BM 0i`).

Values:

`<Vertical Measure between 0 and 425 inches>`

A number between 0 and 425 inches. This is the amount of white space to be left at the bottom of each page.

Examples:

- `+BM .75i`
Leave a .75 inch bottom margin on each page.
- `-BM`
No bottom margin.

Notes:

- The bottom margin is actually measured in printer specific vertical units (1/72nd inches for Epson MX, RX and FX printers). The value is converted to vertical units; any fractional units are ignored.
- If there is no bottom margin (`-BM`), then there can be no Footer Line (FL).
- If FL is specified, the bottom margin must be large enough to accommodate the height of the footer line plus the footer margin (FM).

Suggestions:

If you change the Page Length (PL) you will probably want to also change the top and bottom margins (TM and BM).

Part P -- Printing with Fonts

CF -- Concatenate Files

Concatenate Files -- CF

Indicate whether files are to be individually printed or concatenated during printing.

Syntax	+CF or -CF	[default: +CF]
---------------	-------------------	-----------------------

Usage:

- **+CF**
The text from all files specified with the Filenames (FI) parameter is to be printed as if it came from one file. No page breaks are automatically included between files. However, each file must end with a carriage return or one will be provided (if not using word wrap). If using word wrap, lines will be appended to each other as words fit, even between concatenated files.
- **-CF**
The text from all files specified with the Filenames (FI) parameter is not to be concatenated. A page break is automatically inserted before printing the contents of the next file and all embedded printing controls are reset (e.g. font, underlining, word wrap, indentation). Page numbering will restart at 1 or the value specified with the Initial Page Number (PN) parameter for each file.

Values:

none

Examples:

- **+CF**
Concatenate the contents of all files in the Filenames (FI) list without page breaks.
- **-CF**
Force a page break between printing the contents of each file in the Filenames (FI) list.

Notes:

- See Initialization String (IS) parameter for related information.
- See Control-F - next file interactive command, section P.4.

Suggestions:

Use **+CF** for printing a document that has several sections where each section is contained in a different file. Use **-CF** for printing several unrelated documents in the same "run" of Pfont.

Part P -- Printing with Fonts

-- Command Indicator

Command Indicator -- CI

Specify a character to identify commands embedded in text files.

Syntax `+CI <1 character> or -CI` *[default: +CI \]*

Usage:

- `+CI <1 character>`
A single character to be used as a flag to indicate embedded commands within the text files. Normally this character is '\'.
• `-CI`
Use the default command indicator ('\'). Same as `+CI \`.

Values:

`<1 character>`

The command indicator must be 1 character and should not be a character used for an embedded command. The upper and lower case characters and special symbols #, !, :, <, >, _ are specifically not recommended. It is best to select an infrequently used character such as “ ”, ‘/’ or ‘%’.

Examples:

- `+CI %`
Change the command indicator from '\’ to ‘%’.
- `-CI`
Use the default command indicator.

Notes:

- Command indicators are case sensitive; that is, they only work in the case (upper or lower) that you specify. Warning, CP/M users: if the command indicator parameter is specified on the CP/M command line, the CI will automatically be converted to upper case.
- If you need to print the command indicator, type it twice in the text file (e.g. ‘\’ will print ‘\’).

Suggestions:

- Change the command indicator if the default command indicator ‘\’ is frequently used in a document or is difficult to generate with your keyboard. Be careful to change the command indicator each time you reprint a file containing other than the default command indicator (using an FFI file, section P.2.3, will make this easy).
- Apple users – specify ‘+CI /’ (for example) to allow the command indicator to be a character easily producible from your keyboard.

Part P -- Printing with Fonts

FI -- Filenames

Filenames -- FI

Specify the files containing the text to be printed by Pfont.

Syntax: `+FI <list of 1 to 15 DOS (or CP/M) file names> [default: none]`

Usage: `+FI <list of 1 to 15 DOS (or CP/M) file names>`
Print the contents of the listed files. When more than 1 file is specified the files can either be concatenated or printed with intervening form feeds (\p) as specified by the Concatenate Files (CF) parameter. The default is to concatenate files. Note that `-FI` is meaningless.

Values: `<list of 1 to 15 DOS (or CP/M) file names>`
Each file named must be a valid DOS (or CP/M) file. File names containing special Pfont characters ('+', '-', '?', '&', ...) must be enclosed in double quotes (""). Two special names may be used in the file list. A file name of 'tty:' (or 'con' for DOS only) indicates that input is to be obtained from the user's terminal as if it were input from a file. With this parameter, Pfont allows your computer to be used as an electronic typewriter. ('Control-Z') indicates the end of terminal input. 'tty:' may appear anywhere in the file name list and may appear more than once. The file name 'loop:' indicates that the list of file names is to be used repeatedly until the user explicitly causes printing to stop. 'loop:' should appear as the last name in the list and should not be the only name in the list.

Examples:

- `+fi manual.txt B:appendix`
Print the contents of files "manual.txt" and "appendix" ("appendix" from disk 'B').
- `+FI envelope.ff tty: letter.ff loop:`
Print contents of "envelope.ff", then enter text to be printed using the keyboard, then print contents of "letter.ff". Continue this sequence indefinitely until user enters Control-Z (CP/M - user Control-C) to force it to stop.

Notes:

- At most 15 items may appear in the list of file names.
- A sequence of file names may be entered without the '+FI' control for convenience, but the file names may not immediately follow any other list. This allows a command line such as:

`Pfont manual.txt appendix`

Part P -- Printing with Fonts

FI is special in this way, it is the *default parameter*.

- The files are checked to determine that they can be opened for use by the Pfont program.
- See the Concatenate Files (CF) parameter for controlling the printing of multiple files.

ggestions:

Use **'tty:'** (or **'con'**) for using your computer as an electronic typewriter or for filling variable sections of letters and forms. Use **'loop:'** for printing multiple copies of the same file or sequence of files. When using **loop:**, it is usually desirable to specify **-CF** so that each copy begins on a new page. See also **+NC** for making multiple copies.

Part P -- Printing with Fonts

+FL -- Footer Line

Footer Line -- FL

Control printing of a footer line in the bottom margin of every page or separately for odd and even pages.

Syntax +FL <list of 1 or 2 strings> or -FL [default: -FL]

Usage:

- +FL <string> or +FL <string> <string>
Specify 1 line to appear at the bottom of every page at a distance from the bottom of the page indicated by Footer Margin (FM). If 2 strings are specified, the first is to appear on odd numbered pages only, the second on even numbered pages only. If only one string, then this will appear on all pages.
- -FL
No footer line will appear in the bottom margin.

Values:

<list of 1 or 2 strings>

The string is a line to appear in the bottom margin of every page. The string can contain any embedded commands to control fonts, underlining etc. The length of the string must not exceed the Line Width (LW). If the string contains any spaces or special Pfont characters ('&', '?', etc.), surround the string with quotes (""). If one string is used, this will be the footer line for all pages. If two strings, the first will be used for all odd numbered pages, the second for all even numbered pages.

Examples:

- +FL "A Day at the Lake -- Section 3"
The string A Day at the Lake -- Section 3 will appear in the bottom margin of every page. It will start at the left margin.
- +FL "\f2The Biker\f1\rpage \s#"
The bottom margin of each page will contain a line with The Biker left justified and in font 2, and the word page followed by the page number right justified and in font 1.
- +FL "Page \s#\rChapter 2" "Chapter 2\rPage \s#"
The chapter and page number information will be printed at the bottom of every page, however it will be printed in reverse order on odd and even pages. This would be used to ensure the page number appears on the outside edge of each page in a bound document printed on both sides of the paper.

Notes:

- The footer line will always be printed using font 0, no underlining etc. unless otherwise specified.
- Each footer must be restricted to a single print line.

Part P -- Printing with Fonts

- See the Not On First Page (NF) parameter to disable the printing of the footer line on the first page of a document.
- See FM and BM for determining the exact location of the footer line on a page.
- The current font, underlining, etc. for normal text printing will not be affected by any embedded commands in the footer line.
- To print a footer line on even or odd pages only, you must use a 'filler' for the other type of page. For example:

```
+FL "" "Even footer"
```

will print 'Even footer' on even pages with no footer printed on odd pages. Likewise,

```
+FL "Odd footer" ""
```

will print 'Odd footer' on odd pages.
- CP/M users: CP/M converts all characters on the command line to upper case. Pfont will adjust so that the first word of a string is capitalized. For best results, avoid this problem by using interactive mode or a parameter input file to input the footer line.
- Most embedded commands can be used within footer lines to control the formatting of these lines. For example, \f, \r, \c, \<, \> will work properly. Vertical formatting commands (e.g. \v, \p) will be ignored when used within footer lines.
- Word wrap (\w) could be used inside of footer lines. In this case, the footer line will be filled with the proper number of words to fit within the current margins and indentation. This means that extra words to place in the footer may be taken from the text file. This is not recommended as the results are very difficult to predict.

Suggestions:

Use FL with the \s# embedded command to print the page number at the bottom of each page. Use FL with the \c and \r embedded commands to produce a left, centered and right justified footer line.

Part P -- Printing with Fonts

FM -- Footer Margin

Footer Margin -- FM

Specify the amount of white space between the bottom of the footer line and the bottom of a page.

Syntax: `+FM <Vertical Measure> or -FM` [default: `+FM .25`]

- Usage:**
- `+FM <Vertical Measure>`
Leave the specified white space between the bottom of the footer line and the bottom of the page.
 - `-FM`
Place the footer line at bottom of page, no white space following the footer line.

Values: `<Vertical Measure between 0 and 425 inches>`
A number between 0 and 425 inches. This is the amount of white space between the bottom of the footer line and the bottom of the page.

- Examples:**
- `+FM .5i`
Leave .5 inches of white space between the bottom of the footer line and bottom of page.
 - `-FM`
No footer margin.

Notes: The footer margin is actually measured in vertical printer dots (1/72nd inches for Epson printers). The number of inches is converted to dots; any fractional dots are ignored. The footer margin plus the height of the footer line must be less than the bottom margin. If no footer line (FL) is specified, the footer margin is not used.

Suggestions: Consider changing the header margin and footer margin (HM and FM) when changing PL, TM or BM.

Part P -- Printing with Fonts

-- Fonts

Fonts -- FO

icate the fonts to be used for printing.

+FO <list of 1 to 15 font names> or -FO [default: -FO]

age:

- **+FO <list of 1 to 15 font file names>**
Print using *Fancy Fonts*. The names and order specified in the file name list specifies the fonts to be used. See the \f embedded command. CP/M users: a maximum of 10 font file names is allowed.
- **-FO**
Do not use *Fancy Fonts*. Use native printer fonts (see \f command). If +FO is not provided with one or more font names, then only native printer fonts can be used in the document.

lues:

<list of 1 to 15 *Fancy Font* file names>
Each name should be that of a font file containing the definitions of characters. The conventional '.fon' extension need not be given; if it is left off, Pfont will append it. The first font in the list will be known as font 0, the second as font 1, etc. CP/M users: a maximum of 10 font file names is allowed.

amples:

- **+FO Romn12 Romnb12 Sans10**
Use font file 'Romn12.fon' for font 0, 'Romnb12.fon' for font 1 and 'Sans10.fon' for font 2.
- **+FO b:Sans8.fon Sans10.fon**
Use font file 'Sans8.fon' (from disk B) for font 0 and 'Sans10.fon' for font 1.

tes:

- The files are tested before printing begins to see if they are valid font files.
- The order of the files is important. It specifies the mapping to the embedded \f commands, but also controls printing speed by partially determining which fonts fit in memory. For fastest printing, put your most frequently used fonts first in the list. If you have an electronic disk emulator, printing speed will be improved by moving 'on disk' fonts onto the electronic disk.
- Before printing, Pfont will indicate which fonts fit in memory and which will be used directly from the disk.

Part P -- Printing with Fonts

- In some CP/M systems there may not be enough memory to accommodate 10 fonts. In this case, an error message will be displayed. You will need to reduce the number of fonts in your document and/or reduce the length of header lines (HL), footer lines (FL) and substitution strings (SU).
- See the appendix for details on fonts (height, width, characters etc.).

uggestions:

When creating a document and determining which fonts to use, write down the names and numbers of each font and reorder the list so that frequently used fonts appear first.

IP

Sele

Usa

Val

Exa

No

Su

Part P -- Printing with Fonts

First Page

First Page -- FP

Print a portion of the document to be printed.

+FP <Page Number> or -FP [default: -FP]

Usage:

- **+FP <Page Number>**

Suppress all printing until the specified page number is to be printed. Resume normal printing from the specified page to end of document or to +LP specification.

- **-FP**

Normal printing, begin printing with the first page of document.

Options:

<Page Number between 1 and 9999>

The page number that is the first page to be printed. Any integer between 1 and 9999 may be specified, however if the number is greater than the page number of the last page, no pages will be printed.

Examples:

- **+FP 4**

Page number 4 will be the first page printed.

- **-FP**

Normal printing, begin printing with first page.

Notes:

- FP works with the same number that would be printed on a page; i.e., the one specified with the Initial Page Number (PN) parameter.

- Pages which are not printed are still processed by Pfont and thus all embedded commands are checked and processed and all warning and error messages are displayed.

Suggestions:

- Use this parameter in conjunction with the Last Page (LP) parameter to select one, or a sequence of pages to be printed. This is particularly useful if you need to reprint a portion of a document, but not the entire document.

- Since error and warning messages will be displayed for all pages of a document (whether printed or not), the +FP parameter may be used to check a document for errors (such as improper commands) without any printing or screen display (see Screen Display, SD). This is accomplished by using the FP parameter and specifying a

Part P -- Printing with Fonts

large page number (e.g. +FP 9999). To speed font loading, always use +SD in conjunction with this technique.

HL

Cor

Us

Va

Ex

N

Part P -- Printing with Fonts

-- Header Line

Header Line -- HL

Control printing of a header line in the top margin of every page.

+HL <list of 1 or 2 strings> or -HL [default: -HL]

age:

- **+HL <string> or +HL <string> <string>**
Specify 1 line to appear at the top of every page a distance from the top of the page indicated by Header Margin (HM). If two strings are specified, the first string will appear on all odd numbered pages and the second string will appear on all even numbered pages. If only one string is specified, it will appear on all pages.
- **-HL**
No header line will appear in the top margin.

ues:

<list of 1 or 2 strings>

The string is a line to appear in the top margin of every page. The string can contain any embedded commands to control fonts, underlining etc. The length of the string must not exceed the Line Width (LW). If the string contains any spaces or special Pfont characters ('&', '?', etc.), surround the string with double quotes ("").

amples:

- **+HL "Chapter 4 -- Going to School"**
The string Chapter 4 -- Going to School will appear in the top margin of every page. It will start at the left margin.
- **+HL "\f2Canoeing\cThe Wilderness\f1\rpage \s#"**
The top margin of each page will contain a line with Canoeing left justified and in font 2, the words The Wilderness centered (also in font 2) and the word page followed by the page number right justified and in font 1.
- **+HL "Page \s#\rChapter 2" "Chapter 2\rPage \s#"**
The chapter and page number information will be printed at the top of every page, however it will be printed in reverse order on odd and even numbered pages. This would be used to ensure the page number appears on the outside edge of each page in a bound document.

tes:

- The header line will always be printed using font 0, no underlining etc. unless otherwise specified.
- See the Not On First Page (NF) parameter to disable the printing of the header line on the first page of a document.

Part P -- Printing with Fonts

- See HM and TM for determining the exact location of the header line on a page.
- The current font, underlining, etc. for normal text printing will not be affected by any embedded commands in the header line.
- To print a header line on even or odd numbered pages only, you must use a 'filler' for the other type of page. For example:

```
+HL "" "Even header"
```

will print 'Even header' on even numbered pages with no footer printed on odd numbered pages. Likewise,

```
+HL "Odd header" ""
```

will print 'Odd header' on odd numbered pages.
- CP/M users: CP/M converts all characters on the command line to upper case. Pfont will adjust so that the first word of a string is capitalized. For best results, avoid this problem by using interactive mode or a parameter input file to input the header line.
- Most embedded commands can be used within header lines to control the formatting of these lines. For example, \f, \r, \c, \<, \> will work properly. Vertical formatting commands (e.g. \v, \p) will be ignored when used within header lines.
- Word wrap (\w) could be used inside of header lines. In this case, the header line will be filled with the proper number of words to fit within the current margins and indentation. This means that extra words to place in the header may be taken from the text file. This is not recommended as the results are very difficult to predict.

Suggestions:

Use HL with the \s# embedded command to print the page number at the top of each page. Use HL with the \c and \r embedded commands to produce a left, centered and right justified header line.

Part P -- Printing with Fonts

-- Header Margin

Header Margin -- HM

Specify the amount of white space to precede the Header Line (HL).

Fontaz `+HM <Vertical Measure> or -HM` **default: +HM .25i**

Page:

- `+HM <Vertical Measure>`
The amount of white space between the top of a page and the top of the Header Line (HL).
- `-HM`
Do not leave any white space between the top of the page and the top of the header line.

Values:

`<Vertical Measure between 0 and 425 inches>`
A number between 0 and 425 inches. This is the amount of white space between the top of the page and the top of the header line.

Examples:

- `+HM .5i`
Leave .5 inches of white space before the top of the header line.
- `-HM`
No header margin.

Notes:

The header margin is actually measured in vertical printer dots (1/72nd inch for Epson MX, RX and FX printers). The number of inches is converted to dots; any fractional dots are truncated. The header margin plus the height of the header line must be less than the top margin. If no header line (HL) is specified, the header margin is ignored.

Suggestions:

Consider changing the header and footer margins (HM and FM) when changing PL, TM or BM.

Part P -- Printing with Fonts

IS -- Initialization String

Initialization String -- IS

Specify a string to be printed at the beginning of each file.

Syntax **+IS <string> or -IS** **[default: -IS]**

- Usage:**
- **+IS <string>**
Insert a string of characters, possibly containing embedded commands at the beginning of the file to be printed. If files are not concatenated (see CF parameter) this string will be inserted before *each* file.
 - **-IS**
No special initialization, use default formatting controls.

- Values:**
- <string>**
A string of characters, usually embedded commands, to specify the starting font, underlining mode and other printing characteristics. If the string contains any blanks or special Pfont characters, surround the entire string with quotes ("").

- Examples:**
- **+IS "\f3\j"**
Begin printing with font 3 and justified text.
 - **-IS**
No initialization.

- Notes:**
- The initialization string will be applied before printing each file that is not concatenated. Thus, it will always apply to the first file, and will apply to all others if **-CF** (Concatenate Files) is specified.
 - The built-in initialization is: font 0, no justify, no underline (see section P.1 for additional information).

- Suggestions:** This parameter is convenient for printing files without a need to modify them (e.g. listings, data files).

Part P -- Printing with Fonts

-- Left Margin

Left Margin -- LM

Control the Left Margin of the text to be printed.

+LM <Horizontal Measure> or -LM [default: +LM 1i]

Usage:

- **+LM <Horizontal Measure>**
Specify a left margin for all text to be printed.
- **-LM**
Turn off the left margin parameter; there will be no left margin. (Same as +LM 0).

Values:

<Horizontal Measure between 0 and 8 inches>

The number of inches to be used as a left margin for all text to be printed. The left margin must be between 0 and 8 inches (13.6 inches for wider printers). A left margin of 0 inches indicates no left margin and is equivalent to -LM.

Examples:

- **+LM 1.5i**
Set a left margin of 1.5 inches.
- **-LM**
No left margin - text will be printed flush with left edge of page.

Notes:

- The default left margin for Epson FX80 printers only, is .75 inch. The different default is due to the inability to print on the true left edge of form feed paper using this printer.
- All horizontal formatting parameters and commands (Tab, Line Width, \hld, \>, etc.) are specified relative to the left margin. Thus, if the left margin is changed, all the other parameters will reflect that change. If you specify a Line Width (LW) of 6.5 inches and a left margin of 3 inches then you have actually requested a right margin of 9.5 inches which is of course too large for many printers. The left margin cannot be directly changed within a text file; however, the left indent command (\>) can provide the same effect.
- **+LM 0** may cause slanted characters at the beginning of a print line to "disappear" since such characters slightly overhang the margins. We recommend, therefore, using a very small left margin (e.g. +LM .05i).

Part P -- Printing with Fonts

Suggestions: Use the left margin to horizontally reposition text on a page. Use `\>` embedded command to change the indentation from the left margin and effectively change the left margin from within the text file.

LE
Se
U
V
E
N
S

Part P -- Printing with Fonts

-- Last Page

Last Page -- LP

Select a portion of the document to be printed.

Syntax +LP <Page Number> or -LP [default: -LP]

Usage:

- +LP <Page Number>
Stop printing after the indicated numbered page is printed. The Pfont program will terminate following this page.
- -LP
Normal printing, do not stop printing until the end of all specified files.

Values:

<Page Number between 1 and 9999>

The page number that is the last page to be printed. Any integer between 1 and 9999 may be specified.

Examples:

- +LP 4
Page number 4 will be the last page printed.
- -LP
Normal printing, end printing after the last page.

Notes:

LP works with the same number that would be printed on a page; i.e., the one specified with the Initial Page Number (PN) parameter.

Suggestions:

Use this parameter in conjunction with the First Page (FP) parameter to select one, or a sequence of pages to be printed. This is particularly useful if you need to reprint a portion of a document, but not the entire document.

Part P -- Printing with Fonts

LW -- Line Width

Line Width -- LW

Specify the maximum width of a printed line.

Syntax `+LW <Horizontal Measure>` **default: +LW 6.8i**

Usage:

`+LW <Horizontal Measure>`

Indicate the width of printed lines of text. The width is measured from the left margin. Thus a left margin of 1 inch (`+LM 1i`) and a line width of 7 inches (`+LW 7i`) yields a line end of 8 inches. The line width is of particular importance for `\j`, `\c` and `\r` embedded commands. Turning off the line width is not allowed; neither `-LW` or `+LW 0i` would allow any characters to be printed.

Values:

`<Horizontal Measure between 0 and 8 inches>`

The width of the line in inches. The sum of the left margin (LM) and the line width (LW) should not exceed 8 inches for narrow or 13.6 inches for wide printers. If it does, an error message will be displayed and the margin set to the minimum.

Examples:

- `+LW 5i +LM 1.5i`
Line width of 5 inches, left margin 1.5 inches, implied line end at 6.5 inches from left edge of paper.
- `+LW 8i -LM`
Line width of 8 inches, no left margin (0 inches), implied line end at 8 inches from left edge of paper.

Notes:

- If a printed line exceeds the line width, a message will be displayed on the user's terminal and the line will be truncated. No justification will occur for such a line (see `\j` embedded command). This will not happen in word wrap mode.
- *Fancy Font* allows the definition of characters with negative margins. That is, the print head moves backward before or after printing the character. Justification and word wrap are based upon the print head position, thus characters will extend slightly to the right of the printed line. This is generally desirable, occurring mainly for slanted characters. However, you cannot use the complete line width of the printer, you must leave enough room for the overlap (usually 1 or 2 dots in the distributed fonts). On 8 inch printers, the effective right

Part P -- Printing with Fonts

margin, Line Width (LW) + Left Margin (LM) should be set to a maximum of 7.95 inches. A warning message will be displayed if this may be a problem. If the right margin is too large, slanted and overlapping characters may not be printed if they are to appear at the end of a line.

Suggestions:

The left (`\>`) and right (`\<`) indentation commands can be used to effectively change the line width from within the text file being printed.

Part P --- Printing with Fonts

NC --- Number Copies

Number Copies --- NC

Indicate the number of copies to be printed.

Syntax +NC <Integer between 0 and 9999> [default: +NC 1]

- Usage:**
- +NC <Integer between 0 and 9999>
Print the indicated number of copies. Normally only one copy is printed (this means the document is printed exactly once). If +NC is used, *exactly* the same document is printed the indicated number of times. Page numbering, formatting etc. will be identical.
 - -NC or +NC 0 will indicate that no printing is to occur. This is not a good idea, and an error message will be displayed if you ask for 0 copies. (see +SD parameter if no printing is desired).

Values: <Integer between 0 and 9999>
The number of times the document is to be printed, 0 is not recommended!

Examples: +NC 5
Print 5 identical copies of the document to be printed.

- Notes:**
- The special file name 'loop:' can be used with the +FI parameter to print multiple non-identical copies of a document (e.g. consecutive page numbers).
 - +NC can be used with any other parameters, including multiple file names. If +NC is used with the 'con' or 'tty:' file name, you will be prompted for input for each copy. Previous input cannot be reused.

Suggestions: Useful for printing a 'carbon copy' of a letter, multiple copies of return address labels, etc.

Part P --- Printing with Fonts

--- Not On First Page

Not On First Page --- NF

control the printing of Header lines and Footer lines on the first page.

+NF or -NF

[default: -NF]

Page:

- **+NF**
Disable printing of the Header Line (HL) and Footer Line (FL) on the first page of text only. Any header and footer specified by HL and FL parameters will be printed on the second and subsequent pages.
- **-NF**
Retain the normal operation of the Header Line (HL) and Footer Line (FL) parameters.

Values:

none

Examples:

- **+NF**
Disable printing of header and footer lines on first page only.
- **-NF**
Normal printing of header and footer lines on all pages.

Notes:

- TM and BM are not affected by NF.
- Notice that turning this parameter on actually turns off headers and footers on the first page. The built-in setting is off, thus normally any headers and footers will appear on the first page.

Suggestions:

Use the +NF parameter to print letters, and documents with a title page where the first page should not have a header or footer (e.g. page number), but where subsequent pages should have this information.

Part P --- Printing with Fonts

PF --- Process FormFeeds

Process FormFeeds --- PF

Enable/Disable recognition of Control-L characters as page feeds.

Syntax `+PF` or `-PF` **default: +PF**

Usage:

- `+PF`
Any Control-L characters (ASCII decimal 12) in the text will be treated as if they are `\p` commands. That is, they cause the current page to be ejected and a new page to be printed.
- `-PF`
Any Control-L characters (decimal 12) in the text are ignored. They do not force a new page and are not printed.

Values:

none

Examples:

- `+PF`
Control-L in text will force a new page.
- `-PF`
Control-L in text will be ignored.

Notes:

The PF parameter does *not* impact the `\p` embedded command.

Suggestions:

`-PF` is useful for printing assembler listings and other text that may include embedded form feed characters (Control-L).

Part P -- Printing with Fonts

--- Paginate

Paginate --- PG

orthand for enabling or disabling all top and bottom margin parameters.

+PG or -PG

[default: +PG]

age:

- **+PG**
Enable normal function of the Top Margin (TM), Bottom Margin (BM), Header Margin (HM), Header Line (HL), Footer Margin (FM), Footer Line (FL) and Page Length (PL) parameters.
- **-PG**
Turn off (disable) all the top and bottom margin parameters (-TM, -BM, -HM, -HL, -FM, -FL, -PL). This allows Pfont to follow pagination and margin directives from other word processing software. Normal text files will print as one continuous stream of text with no top or bottom margins.

values:

none

xamples:

- **+PG**
Normal pagination.
- **-PG**
Text will be printed continuously with no top or bottom margins.

otes:

none

uggestions:

Normally use the default, +PG. Use -PG if your text already includes page breaks, top and bottom margins, headers and footers or if you do not want pagination of any sort.

Part P -- Printing with Fonts

PL -- Page Length

Page Length -- PL

Specify the length of a printed page including top and bottom margins.

Syntax: +PL <Vertical Measure> or -PL [default: +PL 11]

Usage:

- +PL <Vertical Measure>
Indicate the total length of a page from the top of the paper to the bottom of the paper. A new page will be printed after the specified number of vertical inches has been printed (including top and bottom margins). The page length must exceed the sum of the top and bottom margins.
- -PL
Perform no pagination (same as +PL 0i). There will be no page breaks and no bottom margin. The top margin on the first page will be printed however.

Values:

<Vertical Measure between 0 and 425 inches>
The height of a page in inches. +PL 0i is equivalent to -PL.

Examples:

- +PL 4i
Print using 4 inch pages.
- -PL
No pagination.

Notes:

Maximum page length is 425 inches, however -PL (+PL 0i) specifies no paging and thus an infinitely long page length.

Suggestions:

- Use short pages for printing mailing or diskette labels. For example, address labels are often 1 inch high (including a small gap between successive labels on continuous forms), so '+PL 1i' should be used.
- -PL (+PL 0i) is useful for continuous printing without any page breaks.

Part P -- Printing with Fonts

PN -- Initial Page Number

Initial Page Number -- **PN**

Specify the number to be associated with the first page.

+PN <Page Number> or -PN [default: +PN 1]

- Usage:**
- **+PN <Page Number>**

Normally, if pages are numbered, the numbering sequence will begin with page 1 (-PN or +PN 1). If you require a different starting page, specify that page number by using +PN.

- **-PN**

Use the default initial page number (1). Same as +PN 1.

- Values:**
- **<Page Number between 0 and 9999>**

An integer which is the page number to be associated with the first page. The page number must not exceed 4 digits and must be a positive number.

Examples:

- **+PN 5**

Begin numbering the first page as page number 5.

- **+PN 50**

Begin numbering the first page as number 50.

Notes:

The page number is only printed if the embedded command `\s#` is used to indicate where the page number is to be printed on each page. Page numbers are generally printed in either the header line or footer line, but can be printed anywhere in the text. First Page (FP), Last Page (LP) etc. will refer to the numbering of pages specified by the PN parameter. The page number determines whether odd or even header and footer lines will be used for a given page.

Suggestions:

- The initial page number parameter is convenient for printing different sections of a document using different "runs" of Pfont. For example, if the first section covers pages 1 to 12 then specify +PN 13 when printing the next section. Page numbers must be integers, however a compound page number such as P-89 can be printed by a string such as `P-\s#`.

- To achieve numbered pages (in this case the number centered at the bottom of a page), use:

`+FL "\c\s#" or`

Part P -- Printing with Fonts

```
+FL "\cPage \s#"
```

to print the word 'Page' followed by the actual page number.

PP
Co
U
V
E
N

Part P -- Printing with Fonts

-- Page Pause

Page Pause -- PP

Control automatic and manual loading of paper.

+PP or -PP

default: -PP

Page:

- +PP
Manual paper feed. Printing will stop at the end of each page and the printer will *beep* to indicate end of page. When any key on the keyboard is depressed, printing will resume.
- -PP
Normal, continuous form paper feed. The printer will continuously feed paper and print each page until the end of the document.

Values:

none

Examples:

- +PP
Printer will pause after each printed page.
- -PP
Normal, continuous form printing.

Notes:

- +PP disables the printer's paper out detection circuitry. (Epson MX series printers don't allow this in their graphics mode, so if you wish to supply paper manually then you must manually disable the paper out detection. This is done by either setting the internal printer switch which disables paper out detection or by placing a piece of paper over the paper out detector on the left side of the printer platen).
- The printer may be turned off to allow you to more easily change paper following a page pause, make sure to turn it back on.
- You should probably use -TM and +BM 2i when using +PP because it is difficult to position single sheets of paper in the printer to properly print at the top of the paper.
- The Control-E interactive command (see section P.4) can be used to turn this parameter off during printing.

Part P -- Printing with Fonts

Suggestions: Use this parameter when feeding one sheet of paper at a time. For example when printing on bond paper, letterhead, or preprinted forms.

RD

Spe

Us

Va

Part P -- Printing with Fonts

-- Rough Draft

Rough Draft -- RD

Specify the quality and speed of printing.

Syntax `+RD <Integer between 0 and 6> or -RD` [default: -RD]

Usage:

- `+RD <Integer between 0 and 6>`
or `<integer between 0 and 2>`
Select a rough draft printing mode (between 0 and 6 for FX version, 0 and 2 for MX version).
- `-RD`
Do not use any rough draft mode, use the normal, high quality printing (same as `+RD 0`).

Values:

- `<integer between 0 and 6>`
Epson FX version:
 - 0 Normal printing (240dpi horizontal, 216dpi vertical, even/odd passes, same as `-RD`)
 - 1 Normal printing (240dpi horizontal, 216dpi vertical, consecutive dots may not be printed)
 - 2 MX printing (120dpi horizontal, 216dpi vertical)
 - 3 Rough (120dpi horizontal, 216dpi vertical, consecutive dots may not be printed)
 - 4 Rough (120dpi horizontal, 72dpi vertical)
 - 5 Rough (120dpi horizontal, 72dpi vertical, consecutive dots may not be printed)
 - 6 *Sandpaper* (60dpi horizontal, 72dpi vertical)
- `<integer between 0 and 2>`
Epson MX printers: (substitute 144dpi vertical for 216dpi and 144dpi horizontal for 120dpi for Gemini, C.Itoh printers)
 - 0 Normal printing (120dpi horizontal, 216dpi vertical) same as `-RD`.
 - 1 Draft printing (120dpi horizontal, 72dpi vertical)
 - 2 *Sandpaper* printing (60dpi horizontal, 72dpi vertical)

Examples:

- `+RD 2`
Cause all printing to be at rough draft level 2.
- `-RD`
Highest quality, final copy printing.

Notes:

More fonts fit into memory during rough draft printing (when vertical resolution is reduced), further increasing printing speed.

Part P -- Printing with Fonts

- Suggestions:**
- Use the higher numbered rough draft modes for fastest printing in order to preview documents for pagination, page layout etc. Print your final copy, using the top quality. For some applications, rough draft modes will suffice. (See Screen Display (SD) for other drafting possibilities).
 - See the `\d` embedded command and section P.4 for changing rough draft modes for portions of a document.

Part P -- Printing with Fonts

Reverse Vertical Enable **Reverse Vertical Enable -- RV**
Enable/Disable reverse vertical paper motion.

Syntax **+RV or -RV** **[default: -RV (except FX80)]**

- Usage:**
- **+RV**
Indicate to Pfont that vertical motion commands causing backwards motion should be allowed. This will only work on printers that allow such motion, if enabled for other printers, errors will occur.
 - **-RV**
Do not allow reverse vertical paper motion.
- Values:** none
- Examples:**
- **+RV**
Pfont will attempt to process commands such as |v.5i.
 - **-RV**
Commands such as |v-1.25i will be ignored and an error message will be displayed.
- Notes:**
- This feature can be enabled or disabled regardless of the type of printer being used. However, it should *never be enabled unless you are using a printer that allows backward paper motion* (such as Epson FX80). The default setting for this parameter depends on the printer type specified in the *FancyFont* profile (see appendix 5). If an FX80 printer has been specified, then +RV is the initial (default) setting; in other cases, -RV is the default setting. This capability allows exact interline spacing regardless of fonts and can be used in conjunction with vertical marking and positioning commands to produce more complicated printing.
 - An Epson FX80 printer with auxiliary tractor feed *cannot* perform backward paper motion (apparently the paper will jam).
 - Use this feature to eliminate the interline spacing limitations discussed in section E.2.
 - The use of +RV on a printer that does not support reverse paper motion is likely to result in severe printing and formatting errors.

Part P -- Printing with Fonts

- Suggestions:
- If distortion occurs due to reverse paper motion, then use `-RV`.
 - Using a combination of `\n` to mark vertical position and `\v` to move the print head vertically, you can produce more complicated equations and, with difficulty, multiple columns.

Part P -- Printing with Fonts

SD -- Screen Display

Screen Display -- SD

Preview, on the terminal, the text to be printed.

Syntax `+SD or -SD` **default: -SD**

Usage:

- **+SD**
Display output on the user's terminal in a manner similar to that in which it would be printed. Do not actually print.

- **-SD**
Normal printing -- do not preview the document.

Values: none

Examples:

- **+SD**
Preview document on terminal, no printing.
- **-SD**
Normal printing, no preview.

Notes:

- When using **+SD**, output is displayed at the user's terminal rather than being printed. The output is displayed with the arguments for embedded commands visible. For example, if `\f3` appears in the text file, the `3` will appear in the processed text during screen display.
- If **+SD** is indicated, no fonts will be loaded in memory in order to increase the speed of previewing (the information normally loaded in memory is not used in this mode).
- The Screen Display parameter will preview line width problems and page breaks, but does not display on your console an exact view of printed output (i.e. no visual font changes or proportional spacing).
- **+SD** causes display output paging (the `--more--` message) to remain in effect. Paging is normally turned off during printing.

Suggestions: **+SD** is a quick way to determine whether any command or formatting errors have been made and to preview a document. See the First Page (FP) parameter for an even quicker way to check for errors.

Part P -- Printing with Fonts

SP -- Interline Space

Interline Space -- SP

Control the white space between printed lines.

Syntax `+SP <Vertical Measure> or -SP` [default: +SP .045]

Usage:

- `+SP <Vertical Measure>`
Indicate the amount of white space to appear between printed lines. The distance being measured is from the bottom of one line to the top of the next.
- `-SP`
Indicate that no white space is to appear between printed lines (same as `+SP 0`).

Values:

`<Vertical Measure between 0 and 425 inches>`

The amount of white space to appear between printed lines, measured in inches. A value of 0 (or `-SP`) will cause lines to be printed as close to each other as possible (see section E.2 for details).

Examples:

- `+SP .5i`
Leave .5 inches of white space between printed lines.
- `-SP`
Leave no white space between printed lines.

Notes:

- The `\l` embedded command allows changing of the interline spacing from within the text file. `\l` uses the value of `+SP` as its initial value. `+SP` can not have a negative value, however `\l` may be used with a negative value.
- The actual white space is measured in vertical printer dots (1/72nd inch for Epson MX,RX,FX printers). The specified inches are converted to dots by ignoring fractional dots.
- Depending upon the heights and depths of fonts on 2 successive print lines, more white space than requested may appear. A message will be displayed at the end of all printing to indicate this has occurred. See section E.2 for a description of interline spacing and `+RV` for finer interline spacing control.

Suggestions:

Use the `\l` command to modify interline spacing within your text to be printed or use the `\v+` command to override interline spacing for individual lines.

Part P -- Printing with Fonts

U -- Substitute

Substitute -- SU

Specify substitution strings for \s commands embedded in text files.

+SU <list of 1 to 10 strings> or -SU [default: -SU]

Usage:

- **+SU <list of 1 to 10 strings>**
Specify the contents and order of strings to be substituted for \s commands. The first string corresponds to \s0 in the text, the second to \s1, etc.
- **-SU**
Specify no substitution strings, any \s strings in the text will be flagged as errors.

Values:

<list of 1 to 10 strings>
1 to 10 strings may be listed. Each string may include any embedded command including the \s command. If a string includes spaces or special Pfont characters ('&', '?', etc.) surround the string with double quotes (""). Each string must be no longer 135 characters. The strings are numbered 0 to 9 to allow easy specification of a maximum of 10 strings.

Examples:

- **+SU "The \f2Fancy FontTM\f0 System" "October 1, 1901"**
Substitute the string The Fancy Font System for each occurrence of \s0 in the text file. Notice the change of font. Also substitute the date October 1, 1901 for each occurrence of \s1 in the text file.
- **+SU Xyzzx Xylophone "Zebra?"**
Substitute Xyzzx for \s0, Xylophone for \s1 and Zebra? for \s2.

Notes:

- If an error occurs in a substitution string, the error message may be displayed on the screen prior to the actual printing of the line on which the error occurred.
- CP/M users: if the +SU parameter is specified on the CP/M command line, the first word of each substitution string will be capitalized; all other words in the string will be all lower case. To retain your desired capitalization use interactive mode or parameter input file.

Suggestions:

+SU and \s commands can be combined to associate *styles* with paragraphs or other sections of text. Define a string containing all the embedded commands (fonts,

Part P -- Printing with Fonts

underlining etc.) which define up to 10 *styles* to be used in your documents. Then use the appropriate strings at the start of each paragraph, section, etc. Other uses include specification of frequently used phrases and information that you may wish to change each time a document is printed (e.g. date, addressee).

Part P -- Printing with Fonts

TB -- Tabs

Tabs -- TB

To clear or set evenly spaced or explicit tab stops.

Syntax +TB <list of 1 to 15 Horizontal Measures> [default: +TB .8i]

Usage:

- +TB <list of 1 to 15 Horizontal Measures>
Set up to 15 tab stops.
- -TB
Do not process any tabs; if a tab is detected in the text, an error message will be displayed on the terminal.

Values:

<list of 1 to 15 Horizontal Measures>
(between 0 and 8 inches)

The Horizontal Measures specify tab stops. If only 1 number is given, then up to 15 evenly spaced tab stops will be set (the spacing will be the indicated number of units). If more than one number is indicated, then a tab stop will be set at each of the indicated locations. The numbers must be in increasing order and must be less than the line width. All tab stops are set relative to the left margin. The maximum setting is 8 inches (13.6 inches for wide printers).

Examples:

- +TB .5i 4.25c 701d
Set 3 tab stops, the first 1/2 inch from the left margin, the second 4.25 centimeters from left margin, etc.
- +TB .8i
Set 15 tab stops, .8 inches apart (i.e. .8, 1.6, 2.4 ... inches from the left margin).

Notes:

- Remember that the tabs are measured *from the left margin* and that tab settings do *not* change when indentation is changed ($\backslash>$).
- Tab stops set to the left of the current left indentation setting are ignored. A tab in this area will cause the print head to move directly to the indent location.

Suggestions:

See $\backslash t$, $\backslash h$ and $\backslash i$ commands for use of tabs and for tab alternatives.

Part P -- Printing with Fonts

TM -- Top Margin

Top Margin -- TM

Specify the amount of white space to appear before the top of the main body of text on a page.

Syntax: `+TM <Vertical Measure> or -TM` [default: `+TM .75i`]

Usage:

- `+TM <Vertical Measure>`
The amount of white space to place at the top of each page. This is the distance from the actual top of the page to the top of the first line of text on the page (other than a header line).
- `-TM`
Do not leave any white space at the top of each page, no top margin (same as `+TM 0`).

Values:

`<Vertical Measure between 0 and 425 inches>`
A number between 0 and 425 inches. This is the amount of white space to be left at the top of each page.

Examples:

- `+TM 3.5i`
Leave a 3.5 inch top margin on each page.
- `-TM`
No top margin.

Notes:

- The top margin is actually measured in vertical printer dots (1/72nd inches for Epson MX,RX,FX). The number of inches is converted to dots; any fractional dots are ignored.
- If there is no top margin (`-TM`), then there can be no Header Line (HL).
- If HL is specified, the top margin must be large enough to accommodate the height of the header line plus the header margin (HM).

Suggestions:

If you change the Page Length (PL) you will probably want to also change the top and bottom margins (TM and BM).

Part P -- Printing with Fonts

UU -- User Defined Units

User Defined Units -- UU

Define the 'u' unit of measure.

Syntax +UU <list of 1 or 2 Real Numbers> [default: -UU]

Usage:

- +UU <list of 1 or 2 Real Numbers>
Specify the horizontal and vertical printer dots for the 'u' unit of measure.
- -UU
Do not define the 'u' unit; in this case 'u' will not be processed as a unit.

Values:

<list of 1 or 2 Real Numbers between 0 and 9999>
If one number, then the 'u' unit is defined to be this number of printer dots both vertical and horizontal. If two numbers, the first is the definition of the 'u' unit for horizontal measures (number of horizontal dots per unit), the second is the definition of the 'u' unit for vertical measures (number of vertical dots per unit). CP/M users: you *must* use integers in place of real numbers with this parameter - thus the 'u' unit may be only approximate.

Examples:

- +UU 4.724 2.835
This defines 'u' to be millimeters for the MX80 printer (4.724 horizontal dots per unit is approximately equal to 120 dots per inch). In this case |h+3u would advance the print head 3 millimeters.
- +UU 120 72
Defines inches for the MX version of Fancy Font (that is, 'u' is now identical to 'i').
- +UU 12 72
Defines 'u' as pica characters for horizontal measures and inches for vertical measures.

Notes:

MX printers have a positioning resolution of 120dpi horizontal, 72dpi vertical (smallest unit of Pfont measurement, although actual print resolution is 216dpi). FX,RX printers have a positioning resolution of 240dpi horizontal and 72dpi vertical.

Suggestions:

Use this feature of Pfont to define your own units of measure based on the resolution of the version of *Fancy Font* you are using. Notice how this was done in the example above for millimeters.

Part P -- Printing with Fonts

4.4 COMMANDS WHILE PRINTING

These commands are only available when printing is in progress. It takes a little while to get used to these since there is not immediate response when one of these commands is entered. Type any of these commands *once printing has started*, however the command *will not be acted upon until the start of printing of a new print line*. Furthermore, the print line must be in the main body of text, i.e. not the header or footer line. These commands require a bit of patience but nevertheless can be quite useful.

Control-Z' (CP/M, use 'Control-C') Stop all printing and exit Pfont. Use this if you want to immediately quit printing. Control-Z can be used to temporarily stop printing to allow paper to be repositioned. Make sure to answer 'n' when asked if you really want to quit. Press one additional key to continue printing.

Control-N' (CP/M, use 'Control-P') Stop printing current page and skip to the next page. Use this command to advance to the next page when the current page has an error or is otherwise not worth printing. If you know in advance that you want to use Control-N, try using the FP and/or LP parameters for better control.

Control-D' Cause subsequent lines to be printed using a different horizontal draft mode. You will be prompted for a single digit with which to indicate the desired horizontal draft mode. See \d embedded command for details of horizontal draft modes and their selection, section P.1.2.

Control-E' Pause after the current page has finished printing. If +PP is in effect, this will pause one last time and then will disable subsequent pausing. Paper out will not be detected for a paused page.

Control-F' Stop printing the current file and skip to the next file. Use this if you notice errors in the current file and wish to continue printing the contents of any remaining files.

Control-C' *Immediately* quit printing (CP/M, see 'Control-Z').

ERROR INTERACTION

Pfont is quite flexible in allowing input to be specified in a variety of ways, but it is difficult for Pfont to predict the cause of parameter errors.

Part P -- Printing with Fonts

Therefore, Pfont provides clear indication of where an error occurred, the problem detected and a choice of options as to how to proceed. The options deal with any input that follows the error and generally allow insertion of new input, selective deletion of existing input, continuing as if there were no error, or restarting from the beginning. Dealing with errors is simple if one parameter and one value is typed on each line. It becomes more complicated when the command line interaction, parameter input files and multiple parameters on a line are used.

When an error is encountered the unexpected input, as indicated in the error message, is ignored. Any remaining input can still be processed if you so desire. You can insert, delete, continue with the remainder of input or backup to leave a parameter (such as a font list). **<E>** is added to the normal prompt to indicate that you are currently recovering from an error and therefore have some special options.

The error recovery options are:

Normal Typing: Insert and continue normal processing (type the input to be inserted).

Control-B: Leave the current parameter and continue with other error correction options.

Control-D: Delete the next input word and continue with other error correction options. A final Control-D will stop input from the current parameter input file (if any).

Return (↵): Process remaining input and return to normal interaction.

Control-X: Discard all remaining input on the line containing an error (also stop input from parameter input file) and return to normal interaction (a combination of multiple Control-D's and a Carriage Return).

In the following transcript of Pfont error interaction, user input is underlined, ↵ is 'carriage return' and commentary is *in small italics*.

Part P -- Printing with Fonts

3> Pfont Sample.ff +fo French12 RomnB12 +rd X Y +pl 6i

Invoke Pfont to print the contents of file 'Sample.ff' using 2 fonts, rough draft printing and a page length of 6 inches.

Pfont: Print with Fonts ...

Copyright (C) 1984 SoftCraft, Inc.

Pfont introductory messages.

Sample.ff +fo French12 Romnb12 +Rd X Y +Pl 6i

French12 is an error (font not found). The input line is echoed with a pointer to the error.

Font file 'French12' not found.

Font not on disk.

'French12' has been discarded.

Erroneous input discarded.

Remaining input:

The unprocessed input is displayed and some help is provided indicating error recovery options.

Romnb12 +Rd X Y +Pl 6i

Insert input or press 'Return' to continue

Control-D to delete next input word, Control-B to backup

Control-X to ignore all remaining input and continue

0 Fonts<E> ↵

Prompt indicates error recovery in progress, font file names expected. User types ↵ to continue processing input.

Romnb12 +Rd X Y +Pl 6i

Another error, X is wrong.

Integer was expected.

An integer is required for RD parameter.

'X' has been discarded.

Remaining input:

Y +Pl 6i

Insert input or press 'Return' to continue

Control-D to delete next input word, Control-B to backup

Control-X to ignore all remaining input and continue

Rough Draft<E> Control-D

The user decides to give up with rough draft and to clean up by deleting the next input, 'Y'.

Remaining input:

Now only Pl parameter left.

+Pl 6i

Insert input or press 'Return' to continue

Control-D to delete next input word, Control-B to backup

Control-X to ignore all remaining input and continue

Rough Draft<E> Control-B

Since user decided not to use rough draft, Control-B 'backs out' of the rough draft parameter.

> ↵

User presses ↵ to continue by processing remaining input (i.e. the Page Length parameter.)

Part E -- Editing Fonts

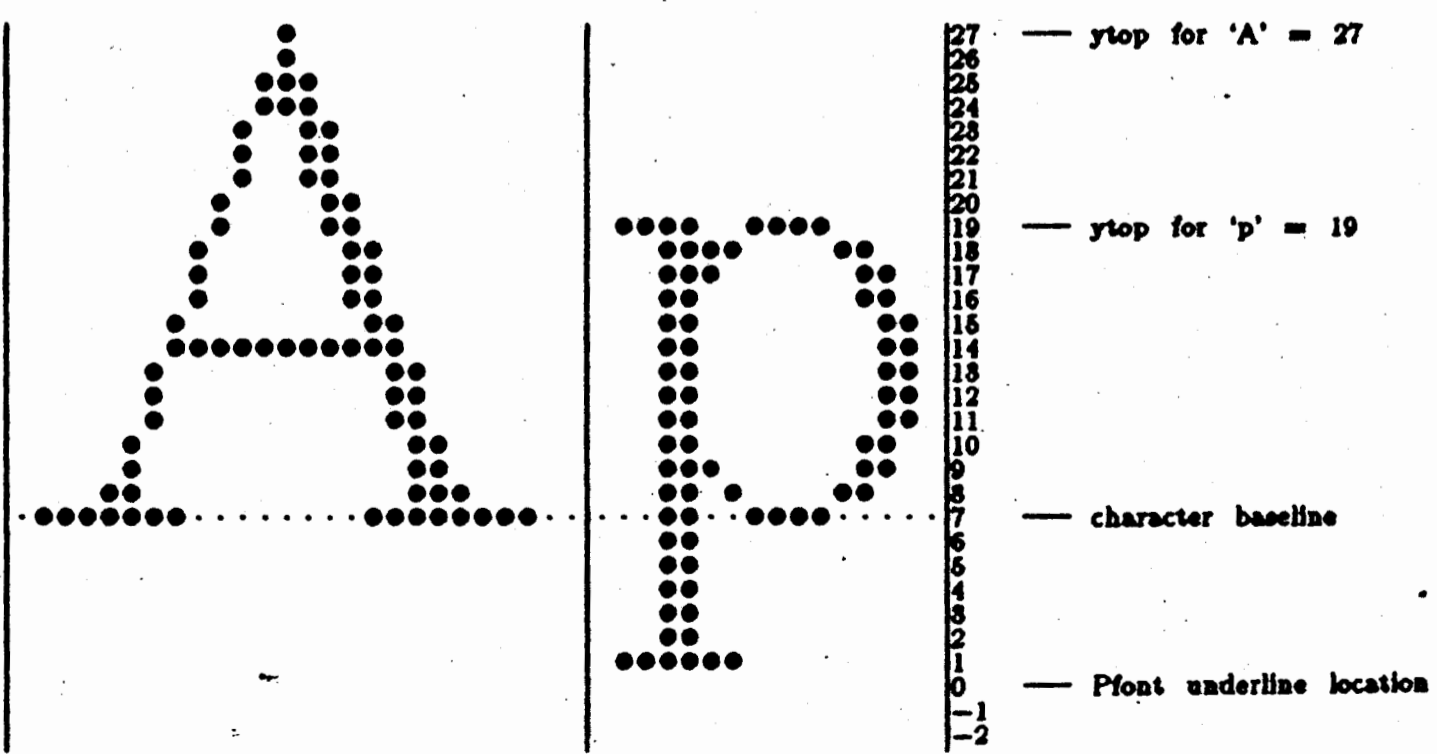
Part E -- Editing Fonts

Efont can be used to create new characters or edit characters in existing fonts. For example, you can use Efont to create a logo or other special graphics symbol; the gray shading in this manual's command description pages was created in this way. Also, you can create new characters such as foreign characters or accents; users have created Sanskrit and linguistic alphabets. Furthermore, you can edit characters in existing fonts in order to achieve special effects or to suit your taste.

Any character in any font set can be edited using any standard text editor. Characters or logos can also be created using a text editor and then installed in a new or existing font set using Efont. The basic operating sequence for editing a character is 1) create a text file for a character using the Efont *edit* command, 2) get out of Efont and edit the character using your favorite text editor, and finally 3) return to Efont and replace the original character with the modified character using the Efont *replace* command.

A character is defined by a pattern of dots (sometimes referred to as pixels). Different printers print dots at different densities. For example, the Epson MX allows printing of as many as 120 dots per inch horizontally and 216 dots per inch vertically while the Epson FX and RX can print 240 dots/inch horizontally and 216 dots/inch vertically.

Consider the following two characters: capital 'A' and lowercase 'p' taken from the Roman 10 point font in the Epson FX version of *Fancy Font*. They are shown here in magnification to illustrate the dots that make up the characters.



The vertical bars on the sides of the characters represent an imaginary

Part E -- Editing Fonts

'box' in which the characters are positioned. When Pfont prints a sequence of characters it places a character so that the left edge of its box just touches the right edge of the box of the preceding character. For example, the left edge of the 'p' box above touches the right edge of the 'A' box (in this picture the two boxes share a vertical line). The spaces around the character within the box are called the character *margins* (sometimes called sidebearings). In this example, the 'A' has a left margin of one dot and a right margin of two dots (the faint dots appearing on the baseline have been included to help count empty dot positions). The 'p' has a left margin of one and a right margin of one.

It is possible for a character to have negative margins, either left or right. Negative character margins are useful in a variety of situations. A character with a negative left margin will 'back up' the print head before printing the ink for the character. Similarly, a negative right margin will cause the print head to back up after printing the character. Slanted characters fit best together with negative margins because of the 'overhang' of slanted characters. Even a vertical character such as 'f' has right overhang that calls for a negative right margin for best fit. Although negative margins are allowed, the overall box width of a character cannot be less than zero. Zero-width characters are allowed, however, and are useful for creating accent characters that are to be overprinted over other characters. Here you might design an accent character that has a large negative left margin so it reaches back over the previously typed character and that has a right margin large enough to bring the print head back to where it was before the accent was printed.

The preceding discussion addressed the horizontal placement of characters on a line; we also need to discuss vertical placement. The numbers printed to the right of the vertical bar next to the 'p' illustrate the vertical dot numbering method used in *Fancy Font* fonts. By convention, almost all of the fonts in the Epson MX and FX versions of *Fancy Font* have their baselines at vertical location seven (section E.2.1 gives some justification for this choice). Counting up from seven we see that the topmost dot in the 'A' is at location 27. We call this the *ytop* value for 'A'. Similarly, the *ytop* value for the 'p' shown above is 19. These three numbers -- left margin, right margin, and *ytop* -- and the dots that make up the character completely define a character. Efont creates and reads files containing this information in editing old characters and creating new characters. We call files of this type '.edf' files because Efont usually saves files with file name extensions of '.edf', as in 'a65.edf' ('edf' is short for 'edit file'). The '.edf' file for the 'A' character above is as follows.

Part E -- Editing Fonts

2 1 27

```

      *
      *
     ***
     ***
    *  **
    *  **
    *  **
   *   **
   *   **
  *    **
  *    **
 *     **
 *     **
*      **
*****
*          **
*          **
*          **
*          **
*          **
**         ***
*****      *****

```

The first number in the file, 2, is the left margin, the second number, 1, is the right margin and the final number, 27, is the *ytop* value. The asterisks represent the dots in the character. If you ask Efont to *edit* the 'A' character in the Roman 10 point font it will create a text file that looks like this. You can then edit that file, changing the margins or *ytop* value to modify the horizontal or vertical placement. You can also move, add or remove asterisks to change the printed form of the character. After that the Efont *replace* command can be used to install the modified '.edf' file as a character in the old font or in a new font.

Note that vertical placement really has no absolute significance; it is only when a font at one vertical location is printed on the same line as a font at another vertical location that the difference in vertical location is noticeable (except for underlining). For example, a superscript 10 point font is exactly the same as the normal 10 point font except that all of the characters have their baselines at a higher location (approximately 18 for the Epson MX/FX version instead of 7 for normal fonts). The way we make a superscript font from a normal font is to increase the *ytop* value of each individual character in the font (the *updown* command conveniently does this for an entire font). A subscript font, of course, has its baseline at a location lower than normal fonts. Section E.2.2 has a more complete discussion about the effects of relative vertical location of characters and fonts.

One common application of Efont is to create special foreign characters. The following transcript demonstrates the procedure for adding an accent to an 'n' in the 12 point Roman font. The small amount of user input is

Part E -- Editing Fonts

underlined and '␣' corresponds to a press of the RETURN (or ENTER) key on the keyboard.

A>efont

Efont: Edit Fonts, version n.nn (Epson FX)

Copyright (C) 1984 SoftCraft, Inc.

Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>l

load a font

Font set file name>romn12.fon

type the font file name

ROMAN 12PT. 4/13/84. COPYRIGHT (C) 1984 SOFTCRAFT, INC.

Top Row=1, Bottom row=0

Blank Points (top)=4, blank points (bottom)=0

Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>e

edit a character

Starting character (0 to 127)>110

'n' is decimal ASCII 110

Ending character (default 110)>␣

accept the default to edit only the 'n'

File name for edit output (default a110.edf)>␣

accept the default file name

a110.edf

Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>q

exit Efont

A>

This will produce a DOS file named 'a110.edf' on the current drive containing the following digits, asterisks and spaces.

1 2 22

```
***** *****
*** *   ***
****   **
****   ***
***    ***
***    ***
***    ***
***    ***
***    ***
***    ***
***    ***
***    ***
***    ***
***    ***
***    ***
***    ***
***    ***
***    ***
***    ***
***    ***
***** *****
```

Use your word processor or text editor to modify the file so that it looks as follows. Asterisks have been added over the 'n' and the ytop value has been changed to account for the increased height of the character.

Part E -- Editing Fonts

1 2 31

```

***          **
*****      **
***  ****   ***
**      *****
**          ***

```

```

*****  *****
*** *    ***
****    **
****    ***
***     ***
***     ***
***     ***
***     ***
***     ***
***     ***
***     ***
***     ***
***     ***
***     ***
***     ***
***     ***
***     ***
***     ***
***     ***
***     ***
*****  *****

```

Now reenter Efont and add the modified character to the romn12 font. We will map the new character to the '^N' (control-N) key to differentiate it from the normal 'n' character

A>efont

Efont: Edit Fonts, version n.nn (Epson FX)

Copyright (C) 1984 SoftCraft, Inc.

Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>l

Font set file name>romn12.fon

load the font again

ROMAN 2PT. 4/13/84. COPYRIGHT (C) 1984 SOFTCRAFT, INC.

Top Row=1, Bottom row=0

Blank Points (top)=4, blank points (bottom)=0

Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>r

replace a character

Starting character (0 to 127)>14

control-N is ASCII 14

Ending character (default 14)>2

replace only one

File name for replacement spec. (default a14.edf)>a110.edf

type the file name rather than accept 'a14.edf'

a110.edf

Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>s

save the font

Output file name for font set>romn12a.fon

give it a new name

Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>q

exit Efont

A>

A control-N (or '\a014') appearing in a text document printed with this new font, romn12a, will print as 'ñ'.

Part E -- Editing Fonts

E.1 COMMANDS

All Efont commands are single letters and do not need to be followed by a carriage return. Immediately upon typing of the command letter the program will prompt the terminal for any required parameters. The list of available commands can be obtained by typing a question mark in response to the 'Command' prompt. Note, however, that question mark generally does not provide help when parameters are being requested by the program. Unlike the Pfont program, Efont uses the standard DOS terminal interface, so Control-C will exit immediately (when in type-in mode) and the normal DOS line editing characters prevail.

The following list summarizes the Efont commands. They appear here ordered according to function while the complete descriptions appear on the following pages in alphabetical order.

l	Load	Load a font set for editing or inspection.
s	Save	Save a modified font set.
e	Edit	Edit a set of characters by creating a text file for each.
r	Replace	Replace some characters with their edit file specifications.
p	Print	Print a set of expanded characters to the printer.
d	Display	Display an expanded character on the terminal screen.
t	Test	Test the font by printing a line of text on the printer.
z	Zap	Remove some characters from the currently loaded font set.
m	Margins	Modify the margins of a range of characters.
u	Updown	Specify a value to move characters up or down by.
a	Auxinfo	Display and/or modify auxiliary font information.
q	Quit	Exit Efont.

The complete descriptions follow.

Part E -- Editing Fonts

-- Auxinfo

Auxinfo -- a

Display and/or modify the auxiliary font information.

Command	a
---------	---

Description: Auxiliary information includes font descriptive commentary and the top and bottom blank points values (section E.2 describes these parameters). This command prints the current values and then asks if they should be changed. Entering a carriage return will always leave the values unchanged. Use this command also to get a quick listing of defined and undefined characters in a font.

Example:

```
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>a
Current values will be listed to the terminal.
Type <return> to retain those values or type a new value.
The file commentary is the following:
Special characters for use with 14pt. 5/5/84.
New commentary (128 chars max.)>Math and special characters for
use with 14pt. 5/8/84.                new commentary
Top row=1, bottom row=0
'A' and 'p' are among the tallest and lowest characters
The computed blank points, top (A) and bottom (p) are:
top=2, bottom=0                        calculated values
The blank top and bottom values recorded in the
file are as follows:
Minimum number of blank points in the top row is 6
New value (default 2) >                here we leave the values unchanged
Minimum number of blank points in the bottom row is 0
New value (default 0) >
The computed font height is 14 points.
There are 98 characters defined in this font set,
and 2 of them are space characters.
You can ask to see the defined or the undefined characters.
Do you want to know which are defined (y or n)>n
Do you want to know which are undefined (y or n)>n
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>
```

Notes:

- When typing new commentary do not type RETURN until the end of the commentary. The input will automatically wrap around on your screen if necessary, as it did in this example.
- The top and bottom blank points are calculated for the characters 'A' and 'p' (ASCII 65 and 112). If these characters are not present or if 'A' is not one of the highest characters in the font or if 'p' is not one of the lowest characters in the font then the blank points values will not be computed but

Part E -- Editing Fonts

you will still be allowed to set them to any value desired.

- The blank points are calculated for 'A' and 'p' and then you prompted for new values. The default values provided by the prompts are the values currently in the font and these are not necessarily the same as the calculated values. If you want to change the values to the calculated values you must enter the calculated values at the appropriate prompt.
- If you accidentally enter this command simply type RETURN ('↵') in response to all of the value-requesting prompts and 'n' in response to the yes/no prompts.

Part E -- Editing Fonts

containing the 'baseline dots' as being the baseline.

- This is similar to the *print* command, but output is to the terminal and only a single character is displayed rather than a range of characters.
- Use the control-S key on your keyboard to control scrolling through the display; hit it once to freeze the display and then again to continue.

Part E -- Editing Fonts

-- Edit

Edit -- e

lit a set of characters by creating a text file for each.

Command

e

Description: One or more characters can be expanded into their asterisk representation and saved into a text file(s). The file can subsequently be edited with any text editor and then the modified character can be reinstalled in the same font or a different font with the *replace* command. The terminal will be prompted for a range of characters to edit (type their decimal ASCII codes). If more than one character is specified then text files with names of 'A<nnn>.edf', where '<nnn>' is the decimal ASCII code for the edited character, will be created, one file for each character, e.g., 'a65.edf'. If only one character is to be edited the terminal will be prompted for a file name for the edit output.

Example:

```
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>e
```

```
Starting character (0 to 127)>65
```

```
Ending character (default 65)>66
```

```
a65.edf a66.edf
```

these files are constructed on the current disk

```
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>
```

Notes:

- Use this command in conjunction with the *replace* command to edit characters, move characters from one font to another, or move characters up and down (in conjunction with the *updown* command).
- If you accidentally enter this command specify a character range of 1 through 0 and no characters will be edited.

Part E -- Editing Fonts

-- Load

Load --

Load a font set for editing or inspection.

Command

l

Description: The terminal will be prompted for a font file name. The complete font name, including the '.fon' extension must be typed. Upon successful loading, font parameters such as top and bottom row, top and bottom blank points, and commentary will be displayed on the terminal (section E.2 describes these parameters). Any font previously loaded will be erased from memory. If the previously loaded font had been modified and not saved, then the terminal will be prompted for confirmation.

Example:

```
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>l
Font set file name>romn12.fon
ROMAN 2PT. 4/13/84. COPYRIGHT (C) 1984 SOFTCRAFT, INC.
Top Row=1, Bottom row=0
Blank Points (top)=4, blank points (bottom)=0
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>>
```

Notes:

- There is slightly more space available for loading a font at the first *load* command than for subsequent loads; if there is not enough room for a font try starting Efont over and load again.
- If there is not enough memory for the font then nothing will be loaded, but the previous font will still be erased.
- This command is different than the load command in Cfont. The Cfont command loads mapping files while this command loads fonts.
- If you accidentally enter this command respond to the prompt with the file name 'fancfont.pro' (or any other non-font file). Efont will notice that the file is not a valid font file and do nothing.

Part E -- Editing Fonts

-- Margins

Margins -- m

Modify the margins of a range of characters.

Command

m

Description: The left and right margins can be set to different values for a range of characters. The terminal will be prompted for a range of characters and for new left and right margin values. (The 'box width' of a character is the horizontal distance taken up by the character on a line. It is the 'ink width' of the character plus the left and right margins. There is currently no way to set the overall box width of a set of characters to a uniform number other than by individually changing the margins of characters according to the ink width of the character.) The margins must be numbers in the range -120 to 120 (for the Epson MX version or -240 to 240 for the Epson FX version). The overall box width of a character must be zero or greater.

Example:

```
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>m
Specify a range of characters to set margins for
Starting character (0 to 127)>65
Ending character (default 65)>66                change 'A' and 'B'
The current margins for character 65 are 0 (left) and 2 (right)
New left margin (-240 to 240)>2                increase the margins
New right margin (-240 to 240)>3
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>
```

- Notes:**
- Negative character margins are useful in a variety of situations. A character with a negative left margin will 'back up' the print head before printing the ink for the character. Similarly, a negative right margin will cause the print head to back up after printing the character. Slanted characters fit best together with negative margins because of the 'overhang' of slanted characters. Even a vertical character such as 'f' has right overhang that calls for a negative right margin for best fit. Use negative margins also for accent characters that are to be overprinted over other letters. In the latter situation you should probably have negative margins that result in a character with a box width of zero.
 - This command has nothing to do with the margins on the printed page (e.g., as set by the Pfont '+lm' and '+lw' parameters); it only affects individual characters.

Part E -- Editing Fonts

- A space character is just a character with a left and/or right margin but no ink. Different sized space characters can be made by simply changing the margins of character codes that were previously undefined in the font.
- If you accidentally enter this command specify a character range from 1 though 0 to leave the margins unchanged.

P --
Prin
Desc
Exa
Not

Part E -- Editing Fonts

- Print

Print -- p

t a set of expanded characters to the printer.

Command

p

Description: A sequence of one or more characters can be expanded into their asterisk representation and printed on the printer. Along with each expansion a *Fancy Font* version of the character will also be printed. The terminal will be prompted for a range of characters to print. Type the decimal ASCII codes of the range of characters to be printed.

Example: Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>p

Starting character (0 to 127)>65

Ending character (default 65)>

a display similar to that of the 'd' command will be printed on the printer in addition to a normal printing of the character

Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>

Note: Also, see the *display* command.

Part E -- Editing Fonts

q -- Quit

Quit -- q

Quit from Efont.

Command	q
---------	---

Description: If any changes have not been made permanent by saving the font, then Efont will prompt for confirmation.

Example: Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>q
Do you want to forget the changes just made (y or n)>n
the font hadn't been saved yet
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>s
Output file name for font set>myfont.fon
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>q
A>

Part E -- Editing Fonts

- Replace

Replace -- r

place a set of characters with their edit file specifications.

Command

r

Description: This is generally used in conjunction with the *edit* command, but can also be used to read in characters or logos created from scratch with a text editor. The terminal will be prompted for a range of characters to be replaced and the replacement file names follow the same convention as do the *edit* command file names. The text file specification of the character to be replaced should begin with three integers, left margin, right margin, and *ytop* values, respectively (see section E.2 for a detailed discussion of the role of *ytop*). The rest of the file should be asterisks and spaces indicating the dot pattern of the new character. The file names of the '.edf' files found will be printed on the screen as the character is replaced. Regardless of the character range given, only those characters with corresponding '.edf' files on the currently logged disk will be replaced. Therefore, if you want all '.edf' files on a disk to be used as replacement files you can simply give the range 0 to 127 instead of giving all of the individual characters.

Example:

```
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>r
```

```
Starting character (0 to 127)>60
```

```
Ending character (default 60)>66
```

```
a65.edf a66.edf
```

'these were the only files on the disk

```
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>r
```

```
Starting character (0 to 127)>44
```

```
Ending character (default 44)>
```

```
File name for replacement spec. (default a44.edf)>mychar.edf
```

```
mychar.edf
```

type a new name to replace with a different file

```
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>
```

Notes:

- Look at a file created by the *edit* command for an example of the required file format.
- See section E.3 for a discussion of the maximum character size.
- If WordStar is used to edit these files be sure to use non-document mode to ensure creation of a DOS compatible file. In general, you should make as plain a file as possible for use with this command.

Part E -- Editing Fonts

- The error message, 'Invalid character ... it should be an asterisk or a space', will be printed on the screen if a character other than an asterisk or a space is found in the replacement file. Some word processors substitute tab characters (ASCII code 9) for sequences of spaces; make sure to turn this feature off in your word processor when creating these replacement files.
- This command may cause the height of the font to change (i.e., if a large character is added to a small font). If the height does change you should use the *auxinfo* command to reset the blank points values.
- If you accidentally enter this command specify a character range of 1 through 0 and no characters will be replaced.

Save
Des
Exa
Not

Part E -- Editing Fonts

Save

Save -- s

the modified font set.

Command

s

Description: Any replacements or other modifications made to a loaded font can be made permanent by saving the modified font on disk. The terminal will be prompted for a font file name; type the entire name including the '.fon' extension.

Example: Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>s
Output file name for font set>myfont.fon
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>

Note: If you try to overwrite an existing file the terminal will be prompted for confirmation. You must give a file name so use the name 'delete.me' if you do not really want to save the font.

Part E -- Editing Fonts

t -- Test

Test --

Test the font by printing a string on the printer.

Command _____ t _____

Description: After replacements or other changes have been made it is useful to test the new font. This command prompts for a string to be printed. A carriage return ends the string input. The Pfont embedded commands '\a', '\u' and '\\ are recognized.

Example: Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>t
String to print>This is a test string
the input string will be printed using the currently loaded font
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>

Note: This command always uses the highest quality mode (i.e., -RD) to print.

Part E -- Editing Fonts

Updown

Updown -- u

Specify a value by which to move characters up or down when editing.

Command

u

Description: The terminal will be prompted for a positive or negative integer. A positive integer indicates that characters should be moved up (by changing the *ytop* value) when they are edited with the *edit* command. A negative integer means move them down. To move characters up or down and make the changes permanent, use the Efont command sequence: 1) *updown*, 2) *edit*, 3) *replace*, 4) *save*.

Example:

```
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>u
```

You can move a character up or down relative to its current vertical position when characters are edited to a file.

```
Move up (positive) or down (neg.) how many dots (default 0)>-5
```

```
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>
```

Notes:

- This command has no immediate effect on the font, but is used in conjunction with the *edit* command. Its effect is also incorporated in the *display* and *print* commands. The *edit*, *display*, and *print* commands are the only commands affected by *updown*.
- The underline location of the `\u` Pfont command is not affected by the *updown* command. That is, if you move a font down by half the height of a character (e.g., to create a subscript font) and then try to underline characters from that font the 'underline' will probably cut through the middle of the character. There is no way to change the underline location of the `\u` command.
- See section E.3 for a discussion of the maximum and minimum allowable vertical locations of characters.
- A character is moved up or down simply by changing the *ytop* value as the '.edf' file is created. The same effect can be achieved by changing the *ytop* number in an existing '.edf' file with your word processor.

Part E -- Editing Fonts

z -- Zap

Zap -- z

Zap (remove) a set of characters from the currently loaded font set.

Command	z
---------	---

Description: This prompts for a range of characters to be removed from the loaded fonts set. The terminal will be prompted for confirmation.

Example: Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>z
Specify a range of characters to delete
Starting character (0 to 127)>0
Ending character (default 0)>31
Are you sure you want to delete these (y or n)>y
Command (one of: l,s,p,e,r,d,m,t,z,a,u,q,?)>

- Notes:**
- This command may cause the height of a font to change, for example, if all of the characters with descenders are removed from a font. If the height of the font does change you should use the *auxinfo* command to reset the *top* and *bottom blank points* values.
 - If you accidentally enter this command give a character range from 0 to 1 and then answer no (type 'n') in response to the confirmation prompt.

Part E -- Editing Fonts

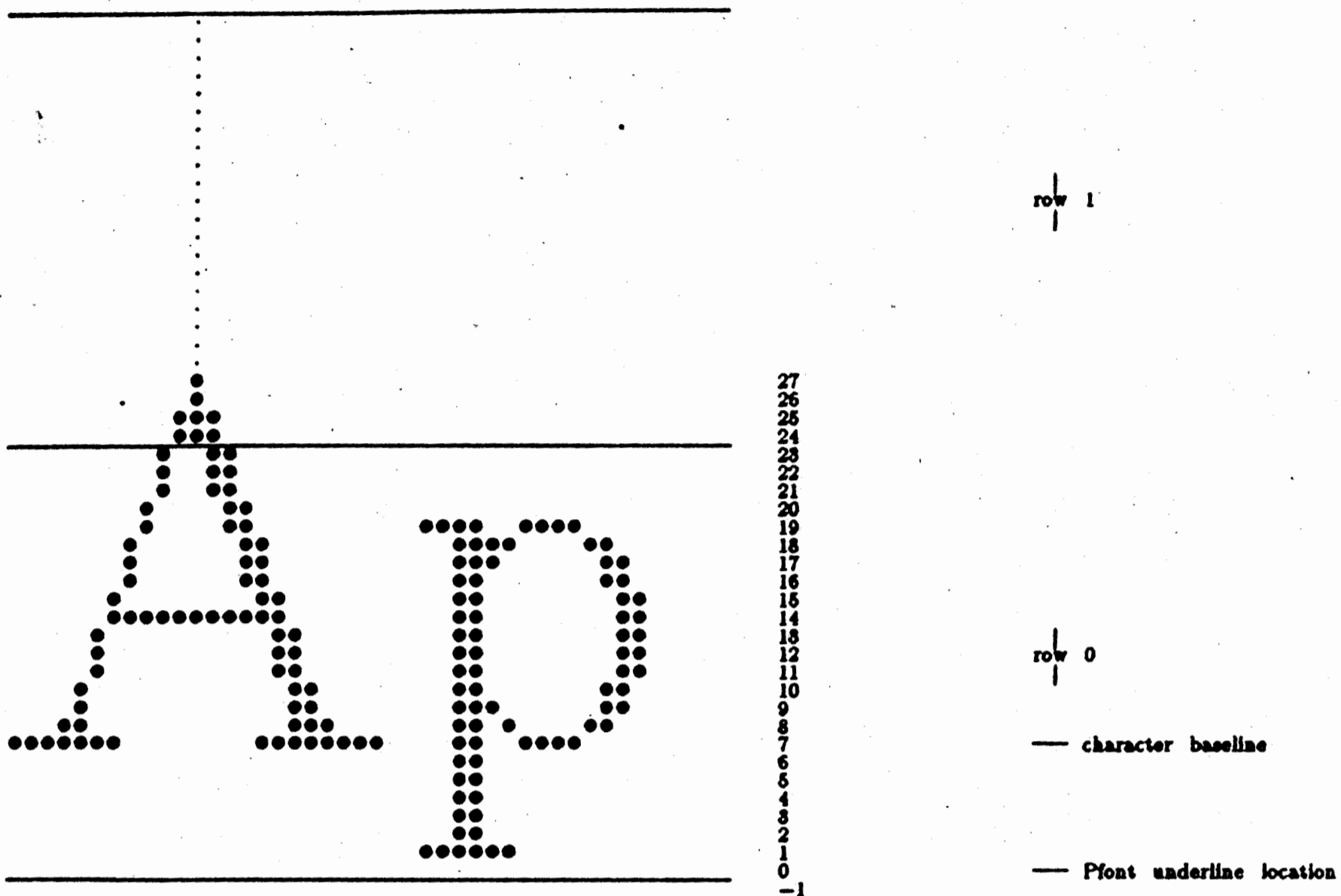
E.2 PRINTING MECHANICS

The different printers supported by *Fancy Font* have different print mechanisms, each resulting in slightly different limitations. These differences will be invisible for the most part but will be of interest to users defining new fonts using Efont and to a lesser extent Cfont. The discussion below addresses the mechanics of the Epson MX and FX versions (remember the FX version works for the RX printer as well) and the resulting interline spacing limitations. Other printers can usually be described with the same words, but with different numbers filled in in various sentences. For example, the Epson MX version prints a column of dots 8 points high (one point is approximately 1/72nd of an inch) in one pass of the print head and requires three passes to fill in such a column while the Toshiba P1350 and the Epson LQ-1500 print a column 9.6 points high and require only one pass.

Because of the parameters of the Epson print head and the minimum allowable paper movement, we find it convenient to define a new term called a 'print head row' or just 'row'. A row contains up to 24 vertical dot locations and is 8 points in height (yielding the 3 dots per point, or 216 dots per inch, resolution of the Epson). Therefore, an 8 point font usually fits into one row.

This term is used because the height of a 'print head row' is usually equal to the height of the printer's print head. However, due to limitations imposed by different printers in their graphics printing modes it may actually require more than one *pass* of the print head to print all of the dots in the row. For example, the Epson FX version requires six print head passes for each row for best quality while the Epson LQ-1500 version requires only one. The following diagrams show how dots are placed on paper in successive rows. The dots between the horizontal lines are all printed in one row with very little paper movement and then the dots between the next set of horizontal lines are printed after a paper movement equal to the height of the row. The letters 'A' and 'p' are from the Roman 10 point font of the Epson FX version and are the same as those used at the beginning of the Efont part of the manual.

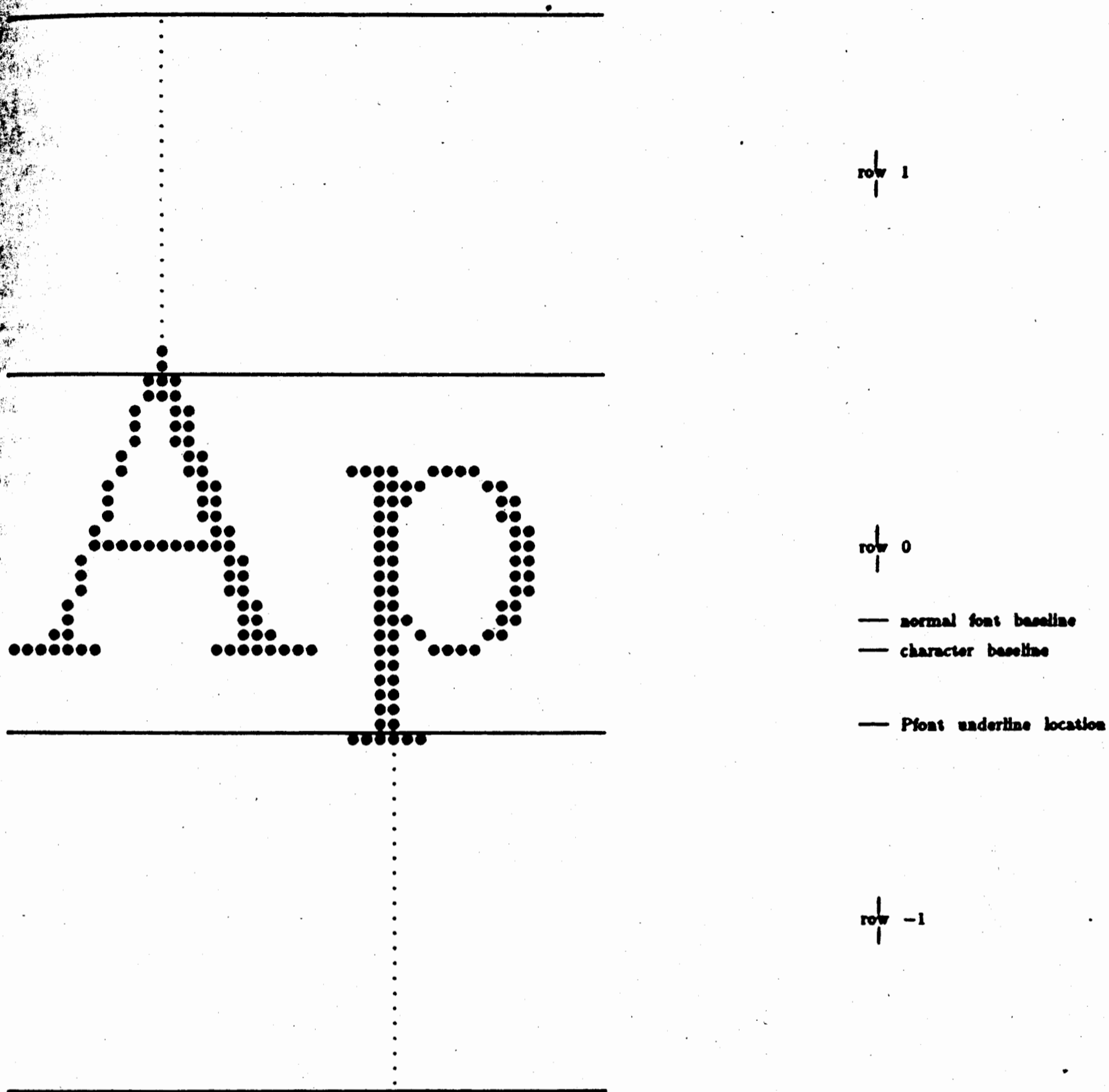
Part E -- Editing Fonts



The rows are numbered for use with Pfont. Each row in the Epson MX and FX version contains 24 vertical dot 'positions' where a position is simply a vertical location at which a dot could be printed. The row number for a row is the height of the lowest position in the row divided by 24. For example, if we count down from the baseline of the 'p' in the figure above (remembering that the baseline for this character is at location 0) we see that the lowest dot position is at location 0. (Remember we are counting positions not dots printed; the lowest dot in the 'p' is at one position above the lowest position in the row where a dot could be printed is at location 24.) Therefore, row zero (dot position '0' divided by 24) contains the center of the 'p' and most of the 'A' while row one contains the top dots of the 'A'. Pfont uses row numbers on a font-wide basis instead of on a character by character basis and so if we assume the 'A' and the 'p' are the highest and lowest characters, respectively, for the Roman 10 font then this font, as a whole, is completely contained in rows zero through row one.

Now consider a font that would be slightly subscript to this font. In this font the baselines of the characters are just two dot locations lower than they were in the normal font. The following diagram shows how the 'A' and 'p' now fit into rows.

Part E -- Editing Fonts



Notice that the 'p' character now descends into row number minus one and therefore the font as a whole uses dot positions in row one, row zero and row minus one. Printing of this font then requires printing of three 'print head rows' instead of the two required on the normal font (18 passes instead of 12 on an Epson FX).

In order for Pfont to achieve precise spacing between two successive lines of print it needs to know two other items of information about a font: two values to indicate the relative position of a font within the top and bottom rows. These are expressed as the number of 'units' of blank space at the top of the top row and at the bottom of the bottom row. (The 'unit' is the vertical spacing unit used in Pfont, i.e., the 'd' unit as in '\v+1d'. For the Epson MX/FX versions the unit is equal to 1/72nd of an inch or three dot positions. For the Epson LQ-1500 version the unit is 1/180th of an

Part E -- Editing Fonts

ch and for the Toshiba version the unit is 1/48th of an inch.) These values are called *top blank points* and *bottom blank points*, respectively.

In the normal (non-subscript) 'Ap' example above the font has a *top blank points* value of 6. The faint dots displayed above the 'A' and below the 'p' are there to facilitate counting the empty positions in the row. For the 'A' there are 20 positions above the topmost dot in the 'A' and because there are three dots per vertical spacing unit in this font we arrive at a *top blank points* value of 6 (truncating '20 divided by 3'). The 'p' has one empty position below its bottommost dot so we get a *bottom blank points* value of zero. These are values for the font as a whole so we have assumed that the 'A' is the highest character in the font and the 'p' is the lowest. Similarly, the subscript font has *top blank points* value of 7 and a *bottom blank points* value of 7 (22/3 and 23/3).

2.1 INTERLINE SPACING LIMITATION

The *top blank points* and *bottom blank points* values are combined in the font program to determine the interline gap or the amount of white space between two lines of text. Consider two consecutive lines of text. Just for discussion call the *bottom blank points* value of the upper line 'upperbp' or upper-line-bottom-blank-points). Call the *top blank points* value of the lower line 'lowertp'. The minimum interline gap between these two lines is 'upperbp+lowertp-7'. For example, consider the subscript 'Ap' font from the previous section. Two consecutive lines of this font would yield an 'upperbp' of 7 and a 'lowertp' of 7, so the minimum interline gap would be '7-7' or 7 points. That is, it would be impossible to have the bottom of a 'p' on the upper line just touch the top of an 'A' on the lower line; there would be at least 7 points of space (21 dot positions) regardless of the value of the '+sp' parameter. (This limitation does not apply to the Epson FX80 printer and other printers that can back the paper up. With these printers Pfont will back up the printer if the requested '+sp' spacing is less than the minimum interline gap. Note that the formula 'upperbp+lowertp-7' may be different for versions other than the Epson (FX.)

Alternatively, look at the normal 'Ap' font. Here 'upperbp' is 6 and 'lowertp' is 0 yielding a minimum spacing of -1 points. That is, the two lines could theoretically overlap by 1 point. (Although the '+sp' parameter does not allow for negative spacing the '\v' embedded command can specify negative spacing and would allow a value of -1, as in '\v-1d', for this case.)

These two examples point up the practical reason for choosing particular values (or baseline value in Cfont). A wider range of '+sp' values is

Part E -- Editing Fonts

ossible when a font fits into the minimum number of rows, thus minimizing the sum *bottom blank points + top blank points*. In Cfont, the standard font baseline of 7 was chosen (for Epson MX/FX versions) so that the descenders of the lower case letters in the 12 point font just fit into row zero. As an additional benefit, minimizing the number of rows for a font minimizes the disk space required to store the font and also maximizes printing speed by reducing the number of passes required to print the font.

2.2 BASELINE AND RELATIVE VERTICAL FONT LOCATION

The manual has stated before that the vertical location of a font has relative significance only, and no absolute significance. This section explains this point further. First, there are three exceptions to the statement. That is, there are three situations where the location does have absolute significance: 1) the location for the Pfont underline (`'\u'`) command is always at zero in the Epson MX/FX versions, 2) characters have to fit within a global coordinate system as described in section E.3, and 3) there are some printer-specific limitations as described above in section E.2.1. Although these are important exceptions the remainder of this section ignores them to simplify the discussion.

This is the key point: Consider a page of text with a line in the middle of it printed in the normal 'Ap' font described in the last section and then consider the same page of text with that line printed in the 'two-dot subscript' 'Ap' font. The page will print exactly the same in both cases right down to the spacing between the lines measured in 1/216th's of an inch. This says that the vertical location of a font is insignificant unless there is another font printed on the same line. If the line of text under consideration were printed half in one font and half in another then, of course, the subscript font would appear slightly lower than the normal font.

When Pfont determines the amount of space to leave between two lines it looks at the bottommost dot of all of the fonts on one line and the topmost dot of all of the fonts on the succeeding line and leaves the '+sp' amount of space between these two dots. (It does not actually look at the characters printed on a line, but instead looks at the top and bottom row values and top and bottom blank points values for a font, as a whole, to determine the topmost and bottommost dots for the fonts on the line.) Therefore with either the normal 'Ap' font or the subscript version there is still the same amount of space between the previous line and the top dot of the line containing the 'Ap' font. In the case where both versions of the font are on the same line there will still be the same amount of space between the previous line and the top dot in the 'A' of the normal font but in determining the interline space for the succeeding line Pfont will look at the bottommost dot in the subscript font because that font is lower

Part E -- Editing Fonts

Overall than the normal font. So in this latter case the overall height of the text line containing the two 'Ap' fonts will be greater than either of the cases where there was only one font on used on the line, but there will be the same amount of whitespace between lines.

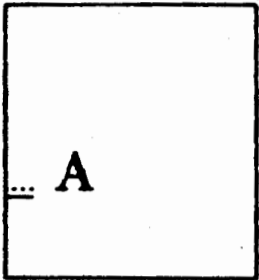
It is sometimes useful to fool Pfont by providing false *blank points* values for a font. For example, a subscript 8 point font that is to be used in conjunction with a normal 12 point font works best if it has a *bottom blank points* value that causes Pfont to think the bottommost dot in the subscript font is at the same location as the bottommost dot in the 12 point font. In this way the spacing between two successive lines of 12 point text looks the same whether there are subscripts on the line or not. If you look closely at a line containing subscripts however, you will see that the bottoms of the subscript characters come closer to the tops of the succeeding line than would the descenders of the 12 point font. That is, the amount of whitespace between the ink of the lines is not uniform but the overall spacing between baselines of successive lines is uniform. The font *auxinfo* command allows you to set the blank points values to any positive integer. In the 8 point subscript case we specify a *bottom blank points* value of 8 to make Pfont think the bottommost dot of the subscript font is at the same location as the bottommost dot of the 12 point font; the 12 point font has a *bottom blank points* value of zero and a bottom row of zero while the subscript 8 point font has a bottom row of minus one, in essence, we tell Pfont that the subscript font should be treated as if it is not extending into row minus one at all (the 8 *bottom blank points* is exactly equal to the height of a row). Note that this method works best on printers that can back the paper up in the printer, such as the Epson MX80. Otherwise, large *blank points* values exacerbate the interline spacing variation described in section E.2.1.

3.3 MAXIMUM CHARACTER SIZE

The maximum character size may vary for different printers, but is always at least one inch by one inch. The maximum character size for the Epson MX/FX versions is one inch wide by a little over one inch high. The figure below consists of two characters, a large box and a 14 point 'A' printed beside the box (remember the reduction factor used in preparation of the manual). The box is the largest character that can be handled by Efont.

Part E -- Editing Fonts

119,167(MX) 239,167(FX)



0,-72(MX & FX)

The 'A' printed inside the box indicates the relative vertical position of the baseline of the distributed fonts. The horizontal location of a character is determined by the left and right margin values (not demonstrated in this example). The solid tick mark at the left of the box is vertical location zero and the dotted mark is location seven, the default baseline. For Epson MX/FX versions the maximum height (i.e., the maximum *ytop* value) is 167, the lowest dot is at height -72. For the Epson MX version the range of horizontal values is 0 to 119 while for the FX version allows a horizontal range of 0 to 239. Other printers have different values depending upon their graphics mode resolution. The *replace* command will ignore any dots lying outside the box. The *generate* command in Cfont also clips characters to fit in the box. See the *baseline* command in Cfont for more information regarding placement of characters inside the vertical range and for an example of the largest character that can be produced from the Hershey character database.

Part E -- Editing Fonts

Pa

The

Her

Pul

Cfo

dat

lon

As

cha

cha

A

He

Sup

fro

380

con

sec

sho

the

coo

He

Cf

ret

ge

th

on

ch

Cf

fo

th

'P

'T

se

pr

Part C -- Creating Fonts

Part C -- Creating Fonts

The Hershey character database is a set of characters created by Alan V. Hershey for the National Bureau of Standards (Wolcott, NBS Special Publication No. 424). The entire database is displayed in Appendix 6. The Cfont program allows the user to choose characters from the Hershey database and map them to ASCII character codes thereby constructing a font set for use with the rest of the *Fancy Font* programs (Pfont and Efont). As indicated in previous parts of the manual, a font set contains up to 128 characters, and usually will contain at least the 95 printing ASCII characters.

A separate tutorial is available for an in-depth introduction to Cfont. Here we provide one simple scenario to illustrate the use of Cfont. Suppose you want to create a font containing only one character chosen from the Hershey database, say a bell character. Hershey character number 380 is a nice bell. Start up Cfont and use the *interactive mapping* command by typing 'm'. Cfont asks for the 'Starting number of font set sequence'. This requests the ASCII code of the character to which the bell should be mapped. In this example, we will map it to 'B' for bell and therefore type '66' as the ASCII code for 'B' (appendix 8 gives the ASCII codes for all characters). Next Cfont asks for the 'Start of matching Hershey sequence'. Here we type '380' to identify the bell character. Next Cfont asks for the 'End of sequence (Hershey number)'. Here we type 380 (↵) to accept the default (380) because only one character is to be generated for the font set. (We could type a number greater than 380 and then Hershey character 381 would be mapped to 'C', 382 to 'D' and so on.) Executing the *generate* command by typing 'g' will cause the bell character to be constructed and associated with character 'B' and then Cfont will ask for the name of an output font file, call it 'bell.fon'. The font file thus created can be used with Pfont to print the bell character in the middle of any other type of text. That is, if the Pfont command line `Pfont text.ff +fo romn12 bell` is executed and the file 'text.ff' contains the line `\f1B\f0 rang`, it will print as 'The \triangle rang'. A transcript of such a session is shown below. User input is underlined and '↵' corresponds to a press of the RETURN (or ENTER) key on the keyboard.

Part C -- Creating Fonts

cfont

Cfont: Create Fonts, Version n.nn (Epson FX)

Copyright (C) 1984 SoftCraft, Inc.

Command (one of: m,l,g,z,d,p,f,b,s,q,?)>m

use the map command

Starting number of font set sequence (default 65)>66

map to the 'B' key

Start of matching Hershey sequence (0 to 1593)>380

choose the bell symbol

End of sequence (Hershey number) (default 380)>1

map only one character

Command (one of: m,l,g,z,d,p,f,b,s,q,?)>g

generate the font

There are 1 characters specified in the font set.

Font set (output) file name>bell.fon

name the font 'bell.fon'

Now sorting the font file.

bell.fon written

top row=1, bot row=0, top blank=2, bot blank=0

The calculated font size is 14 points

Command (one of: m,l,g,z,d,p,f,b,s,q,?)>q

exit Cfont

A>pfont text.ff +fo romn12 bell

I suggest that you print out a copy of the Hershey database on your printer. The copy appearing in the Appendix has been photographically reduced and may have been printed on a printer other than the one you are using. The following transcript shows how to print out a version that will aid you in constructing fonts from the database. Line up the paper in your printer with the perforation about half an inch above the print head before printing.

Command (one of: m,l,g,z,d,p,f,b,s,q,?)>p

Not enough memory to print at that scale factor

Some characters may be clipped

Starting character (default 0) >0

Ending character (default 0) >1593

Command (one of: m,l,g,z,d,p,f,b,s,q,?)>

The Cfont commands are the *scale factor* command, which allows the characters to be reduced or enlarged, the *save mapping* command, which allows interactively specified mappings to be saved on a file and subsequently loaded with the *load mapping* command, and the *print* command, which prints Hershey characters on the printer at the current scale factor for a preview of what the characters will look like when the font set is generated. The *print* command was used to print appendix 6. Some of the characters are truncated in Appendix 6 so that they fit on the page but they can be created in full when generating a font with the *save* command. All of the commands are described in detail below, and a summary of the commands appears in the final appendix.

O.1
The
con
car.
pro
ava
to
doe
anc
con
A
Th
alp
m
s
l
z
g
f
b
p
d

Part C -- Creating Fonts

COMMANDS

The Cfont command interface is exactly the same as that of Efont. All command names are single letters and do not need to be followed by a carriage return. Immediately upon typing of the command letter the program will prompt the terminal for any required parameters. The list of available commands can be obtained by typing a question mark in response to the 'Command' prompt. Note, however, that question mark generally does not provide help when parameters are being requested by the program if the standard CP/M line editing characters are in effect. The command descriptions follow.

A short summary of the commands appears below ordered by function. The following pages contain detailed descriptions of the commands in alphabetical order.

Map	Enter mappings from keyboard keys to Hershey characters.
Save	Save the current mappings to a file for later reuse.
Load	Load a previously saved set of mappings from a file.
Zap	Remove some of the current mappings.
Generate	Generate a font with the current mappings.
Factors	Set the scale factors for enlargement/reduction.
Baseline	Set the baseline of the characters to be generated.
Print	Print a range of Hershey characters on the printer.
Display	Display the current mappings, scale factors and baseline.

Part C -- Creating Fonts

b -- Baseline

Baseline -- b

Set the baseline of the characters to be generated.

Command

b

Description: Any two fonts printed on the same line must have been created with the same baseline if the baselines of the characters are supposed to line up. The default baseline is normally 7, but can be modified to more efficiently store a particular font, to create a subscript or superscript font or to generate a specialized, very large font. For example, you could create a very large font (about 120 points) if it consisted of only upper case letters. After reading section E.3 you would see that a baseline of -72 might allow creation of such a font on the Epson FX printer. A 120 point 'G' (Hershey character 95) appears below in the maximum character box (see E.3). This was created with baseline -72, x1=12, x2=1, y1=34, y2=3 (remember the photographic reduction used in preparation of this manual).



The 'G' is actually 80 points high, but the size of a font measures from the bottom of the descenders to the top of upper case characters, so the 'G' would be considered to be a member of a 120 point font.

Example:

Command (one of: m,l,g,z,d,p,f,b,s,q,?)>b

baseline (default 7)>-20

lower the baseline by 27 dots

Command (one of: m,l,g,z,d,p,f,b,s,q,?)

Notes:

- If you accidentally enter the *baseline* command just type return in response to the prompt to leave the baseline value unchanged.
- There are only four known reasons for changing the baseline of a font to be generated:
 - 1) to create a superscript or subscript font based at a location higher or lower than other fonts (change the baseline by a little more than the number of dots in an 'x'),
 - 2) to create very large characters such that the baseline has to

Part C -- Creating Fonts

be adjusted so the resulting characters fit into the maximum character size box (section E.3),

3) to optimize character placement within 'print head rows' (section E.2.1), and

4) since the location of the Pfont underline command ("`\u`") is fixed you can get the effect of changing the underline location by raising or lowering a font instead.

- Section E.2.2 gives more detail on the effect of different baseline.

Part C -- Creating Fonts

d -- Display

Display -- d

Display the current mappings, scale factors and baseline.

Command

d

Description: The current mappings showing ASCII code, keyboard key and Hershey number will be displayed on the terminal. The terminal will 'freeze' when the screen is full; type any key to display the next screenfull.

Example:

Command (one of: m,l,g,z,d,p,f,b,s,q,?)>d

fontset:Hershey

65(A):0

the 'A' key is mapped to Hershey character 0

66(B):1

the 'B' key is mapped to Hershey character 1

There are 2 characters currently specified

x1=1, x2=1, y1=1, y2=1, baseline=7

the current scale factors

Command (one of: m,l,g,z,d,p,f,b,s,q,?)>

Part C -- Creating Fonts

Factors

Factors -- f

the scale factors.

Command

f

Description: The characters in Appendix 6 can be scaled up or down (reduced or enlarged) by setting the scale factors. As characters are constructed by the *generate* command and the *print* command the horizontal and vertical components of the character are multiplied by a scale factor. Because floating point numbers (e.g., 1.5) are not available in Cfont we use a fraction (e.g., 3/2) as the scale factor. For the horizontal direction we use the fraction $x1/x2$ where $x1$ and $x2$ are integers that can be set using the *factors* command. Similarly, the vertical components are multiplied by a fraction $y1/y2$. This command allows the user to set each of the values $x1$, $x2$, $y1$, and $y2$.

Example:

Command (one of: n.l.g.z.d.p.f.b.s.q.?)>f

The characters will be scaled by the fractions $x1/x2$ and $y1/y2$ in the x and y directions. Normally the y scaling should be about the same as the x scaling.

$x1$ (default 2) >3

change the x scale factor to 3/1

$x2$ (default 1) >1

$y1$ (default 2) >3

same for the y scale factor

$y2$ (default 1) >1

Command (one of: n.l.g.z.d.p.f.b.s.q.?)>

Notes:

- In the Epson MX version the vertical scale factor should normally be about twice as large as the horizontal scale factor.
- The initial scale factors vary for different printers. The Epson MX has about twice the resolution (dots/inch) in the vertical direction as the horizontal direction and therefore the initial Y scale factor ($y1/y2$) is 2 while the X factor ($x1/x2$) is 1. The FX version has initial factors of 2 in both directions.
- The maximum size of a character is one inch wide by a little over one inch high and the character should fall in the box described in section E.3 or else the character will be clipped. See the example in the *baseline* command description.
- Characters can be mirrored by setting $x1$ and/or $y1$ to negative values.

Part C -- Creating Fonts

- Not all of the distributed fonts were derived from the Hershey database and therefore we cannot provide scale factors that would allow you to create matching fonts from the database.
- If you accidentally enter this command simply type return (␣) at each prompt to leave the values unchanged.

Part C -- Creating Fonts

Generate

Generate -- g

Generate the font.

Command

g

Description: After the mappings between ASCII codes and Hershey numbers have been specified the font can be generated. Font generation can take several minutes for standard-sized fonts. Cfont will look for the file 'Hershey.chr' on the logged drive and, failing that, on drive A. If it isn't found then the terminal will be prompted for the proper 'Hershey.chr' file name. A temporary font file is constructed on the logged disk so there must be enough disk space on the logged disk to accommodate two font-sized files (approximately 10K for a 12 point, 95 character Epson MX font or 20K for the same Epson FX font). A period is displayed on the terminal for each character as it is constructed during font generation. After all of the characters are constructed from the Hershey database the terminal will be prompted for the name of the (final) font file. Do not remove the disk in the logged drive at this point as it still contains the temporary font file.

Example:

```
Command (one of: m,l,g,z,d,p,f,b,s,q,?)>g
There are 12 characters specified in the font set.
.....                               one dot for each character
Font set (output) file name>myfont.fon
Now sorting the font file.
.....                               one dot for each non-space character
myfont.fon written
top row=1, bot row=0, top blank=2, bot blank=0
The calculated font size is 14 points
Command (one of: m,l,g,z,d,p,f,b,s,q,?)>
```

- Notes:**
- You can request that the final font be constructed on a different drive by including a disk specifier in the font name (e.g., b:myfont.fon).
 - Osborne note: The 'Hershey.chr' database is larger than the capacity of a single disk, so the file has been split into two parts, 'Hershey.chr' and 'Hershey1.chr'. When 'Hershey1.chr' is on drive A or the logged drive then the continuation should happen automatically. If 'Hershey1.chr' cannot be found Cfont will prompt for a file name. At this point a new disk containing 'Hershey1.chr' can be inserted in the nonlogged drive (the logged disk still contains the temporary font file so



Part C -- Creating Fonts

do not remove that disk).

Part C -- Creating Fonts

Load

Load -- 1

Load a set of mappings from a file.

Command

l

Description: A mapping specified with the *map* command can be saved to a file with the *save* command and subsequently loaded with the *load* command.

Example:

```
Command (one of: m,l,g,z,d,p,f,b,s,q,?)>l  
File for mapping specification>b:myfont.map  
Command (one of: m,l,g,z,d,p,f,b,s,q,?)>
```

Notes:

- If necessary, the *zap* and *map* commands can be used after the *load* command to override some of the loaded mappings. Multiple *load* commands are also permitted to merge mappings from a variety of *saved* mapping files.
- This command loads mapping files, not font files. You should use Efont, not Cfont, to load and modify an already created font file.
- A map file for an Old English font is displayed in section C.2.

Part C -- Creating Fonts

m -- Map

Map -- m

Map keyboard keys to Hershey characters.

Command

m

Description: Enter mappings from the terminal to specify the correspondence between the ASCII codes in the font to be generated and the characters in the Hershey database. Each font generated by Cfont can contain up to 128 characters corresponding to the ASCII codes 0 through 127. The *map* command allows a correspondence to be made between these ASCII codes and the characters in the Hershey database.

The introduction to Part C gives a short example using the mapping command. After the 'm' is typed the program will prompt for the 'Starting number of font set sequence', asking for an ASCII number in the font set to be generated. Next Cfont asks for the 'Start of the matching Hershey sequence.' This specifies a character number in the Hershey database (the database is listed in Appendix 6). Finally, Cfont asks for the 'End of sequence (Hershey number).' Another Hershey character number is requested to indicate the sequence of characters to be mapped. The first Hershey character will be stored in the generated font as the ASCII code entered, and if the second Hershey number entered was greater than the first, then the next Hershey character stored as the following ASCII code, and so on.

For example, to create a font with the upper case Old English letters mapped to the normal ASCII codes, use the mapping command with the following three inputs: (starting ASCII) 65, (starting Hershey) 1457, (ending Hershey) 1482. The mapping command can be used many times to specify multiple noncontiguous Hershey sequences.

Example:

```
Command (one of: m,l,g,z,d,p,f,b,s,q,?)>m
Starting number of font set sequence (default 65)>65
Start of matching Hershey sequence (0 to 1593)>1457
End of sequence (Hershey number) (default 380)>1482
Command (one of: m,l,g,z,d,p,f,b,s,q,?)>
```

- Notes:**
- Once the mappings are complete they can be saved on disk with the *save* command.
 - Other commands useful during the mapping phase are the *zap*

Part C -- Creating Fonts

command which removes mappings, and the *display* command which shows the currently specified mappings.

- The default for the mapped-to (ASCII) number is one greater than the last character mapped to, or 65 if no characters have yet been mapped.
- See the *print* command for information on space characters in the Hershey database.
- If you accidentally enter this command give a Hershey character range from 1 to 0 to avoid adding any mappings.

Part C -- Creating Fonts

p -- Print

Print -- p

Print a range of Hershey characters on the printer.

Command

p

Description: The terminal is prompted for a range of Hershey characters and then these characters are printed on the printer using final draft mode and the current scale factors. Appendix 6 was printed using this command with all Cfont default values. See the *generate* command for a description of the alternative disk arrangements for 'Hershey.chr'. Some of the characters in Appendix 6 are space characters, e.g., 50, 51, and 52. Usually the highest numbered character in a short sequence like this is the widest space character and the smallest numbered is the narrowest. The *margins* command in Efont is generally a more convenient way to precisely specify space characters than trying to identify the appropriate space characters in the Hershey database.

Example:

```
Command (one of: n,l,g,z,d,p,f,b,s,q,?)>p
Not enough memory to print at that scale factor
Some characters may be clipped
Starting character (default 0) >380<
Ending character (default 380) >450<
                                characters will print on the printer
Command (one of: n,l,g,z,d,p,f,b,s,q,?)>
```

Notes:

- Depending upon the current scale factors and baseline, characters may be clipped at the top or the bottom when they are printed. This is to allow printing of a useful number of characters on a page. The characters will not be clipped when they are *generated* as long as they are contained in within the maximum character box (see section E.3 and the *baseline* command).
- If you accidentally enter this command give a character range from 1 to 0 to avoid printing any characters.

Part C -- Creating Fonts

-- Quit

Quit -- q

Exit Cfont.

Command	q
---------	---

Description: This immediately exits Cfont.

Example: Command (one of: n,l,g,z,d,p,f,b,s,q,?)>q
A>

Note: Cfont does not keep track of whether changes made in the mapping information have been saved to file so be sure to *save* the mapping information before exiting Cfont if you think you will need the mapping again.

Part C -- Creating Fonts

-- Save

Save -- s

Save the current mappings to a file.

Command

s

Description: Save the current mappings to a file so that they can be loaded with the *load* command. The terminal will be prompted for a file name and the file thus created is a normal ASCII file and can be edited with a text editor if desired.

Example:

```
Command (one of: m,l,g,z,d,p,f,b,s,q,?)>s
Mapping output file name>myfont.map
myfont.map already exists. Do you want to overwrite it (y or n)>y
Command (one of: m,l,g,z,d,p,f,b,s,q,?)>
```

Notes:

- This command is for saving a mapping file; it does not create a font file. The file name used here should be different than the one used in the *generate* command. We recommend using the file name extension '.map' as in 'myfont.map' for map file names.
- A map file for an Old English font is displayed in section C.2.

Part C -- Creating Fonts

-- Zap

Zap -- z

Remove some of the current, previously specified mappings.

Command

z

Description: The terminal will be prompted for a range of ASCII-to-Hershey mappings to be removed.

Example: Command (one of: n,l,g,z,d,p,f,b,s,q,?)>z
Start character of mappings to be deleted (0 to 127)>33
Ending character (default 0)>44
Command (one of: n,l,g,z,d,p,f,b,s,q,?)>

- Notes:**
- This command does not prompt for confirmation, but immediately removes the mappings.
 - If you accidentally enter this command give a character range from 1 to 0 to avoid deleting any mapping information.

Part C -- Creating Fonts

3.2 BATCH MODE OPERATION

Cfont can be executed from a single command line for use within a batch file (or CP/M submit file). This mode can be used to create several fonts at one time without having to wait at the terminal to answer the various Cfont prompts. This mode of operation requires the use of a previously created map file. To use Cfont in this manner type the command line

```
Cfont <fontfile> <mapfile> <x1> <x2> <y1> <y2> <baseline> <comments>
```

where <fontfile> is the desired output font file name, <mapfile> is the name of the mapping file for the font, <x1>, etc. are the scale factors, <baseline> is the baseline and <comments> is any descriptive commentary for the font. <comments> may contain spaces and should not be surrounded by double quotes.

The following transcript uses the 'olde.map' file distributed with Cfont to create a 20 point Old English font. The map file is also displayed below.

```
A>cfont olde20.fon olde.map 2 1 2 1 7 Old English, 20pt. 5/5/84
```

```
Cfont: Create Fonts, version n.nn (Epson FX)
```

```
Copyright (C) 1984 SoftCraft, Inc..
```

```
There are 98 characters specified in the font set.
```

```
Have to skip over 324 characters in the Hershey file so
```

```
it may take a while until the first dot appears on the screen.
```

```
.....
```

```
Now sorting the font file.
```

```
.....
```

```
olde20.fon written (Old English, 20pt. 5/15/84)
```

```
top row=2, bot row=-1, top blank=6, bot blank=6
```

```
The calculated font size is 20 points
```

```
A>
```

The file 'olde.map' contains the following mappings. The mapping numbers appear in pairs or triples separated by commas. The first number is the ASCII number and the second is the Hershey number to which the ASCII is mapped. If the second number is followed by a colon then the number following the colon is the end of the Hershey mapping range following the same convention as the *mapping* command.

```
24 1510, 30 909, 31 1536, 32 1510, 33 1526, 34 1540, 35 642, 36 641,  
37 638, 38 1530, 39 1529, 40 589:590, 42 1535, 43 1537, 44 1523,  
45 1536, 46 1522, 47 588, 48 1512:1521, 58 1524:1525, 60 888,  
61 1538, 62 889, 63 1527, 64 920, 65 1457:1482, 91 591, 92 324,
```


Part C -- Creating Fonts

93 592, 94 894, 95 878, 96 1528, 97 1483:1508, 123 593, 124 597,
125 594, 126 893 ;

You can easily make 10, 20 and 30 point fonts in Old English and Script
y using Cfont in the batch mode using the batch file (or CP/M submit
le) supplied on the Cfont distribution disk. Follow these directions.

) Copy the contents of the Cfont disk to a fresh system disk (i.e., on
MSDOS the file COMMAND.COM should be on the disk and in CP/M
the CP/M operating system should be on the disk's system tracks) and
place the new disk in drive A.

) Log to a drive containing at least 170k of free space.

) Type
a:mkfonts (on MSDOS), or
a:supsub mkfonts (on CP/M)

ptions:

) If you have a hard disk copy the contents of the Cfont disk to a
directory, log to that directory, type 'mkfonts' (on MSDOS) or 'supsub
mkfonts' (on CP/M).

) If you have an electronic memory with between 170k and 270k of free
space log to that drive in step 2.

) If you have an electronic memory with 270k of free space insert the
following step between steps 2 and 3 above.

2.5) copy a:hershey.chr (on MSDOS), or
pip m:=a:hershey.chr (on CP/M, where 'm' is the electronic
disk drive)



--- Notes ---

As
top
Ba
ch
CF
ne
De
Th
mc
De
bot
DC
sys
son
Do
ch
eq
Fa
W
giv
FI
fro
FI
Fo
or
Fo
pr
H
H
cr
C

Appendix 1 -- Glossary

ascender -- The stroke of any lowercase letter which extends above the top of an x, e.g., b, h.

baseline -- The location of the bottoms of most letters, e.g. A, x. Some characters descend below the baseline, e.g., g, p.

CP/M -- A popular operating system for microcomputers. The full, but never used name is Control Program/Microcomputer.

default -- The initial, pre-defined setting for a command or parameter. This is the setting that will take effect unless the user explicitly indicates a modified setting.

descender -- The stroke of any lowercase letter which extends below the bottom of the baseline, e.g., g, p.

MS-DOS -- 'Disk Operating System', a common name for the operating systems used by the IBM Personal Computer and its compatibles, sometimes called PCDOS.

dot -- An element of the dot pattern that creates the type image of a character. Also, often used in this manual as a unit of horizontal measure equal to 1/120th inch.

face -- Face typically refers to the boldness of a font, i.e., boldface type. We use face to differentiate between regular, bold, and italic forms of a given font style.

justify -- To place as many words on a line as possible by drawing words from subsequent lines until the end of a paragraph is reached.

flush left (or right) -- Type set to line up at the left (or right) margin.

font -- The complete assortment of type of one size, face, and orientation.

format -- The size, style, margins, printing requirements, etc. of a printed document.

hairline -- The thinnest part of a letter.

Hershey database -- The Hershey character database is a set of characters created by Alan V. Hershey for the National Bureau of Standards. The font program uses this database to construct new fonts.

Appendix 1 -- Glossary

Horizontal Measure -- A term used by Pfont to indicate a horizontal distance. A horizontal measure must include a 'real' number and a unit. For example, '1.25i' to indicate 1 1/4 inches.

Italic -- Normally, a style of type with letters slanting to the right, patterned on a compact manuscript hand. Used in this manual to differentiate any right-slanting font from its related vertical font.

Justify -- To space out lines by increasing the size of the spaces to make the right ends of the lines flush right.

Kerning -- To adjust the spacing between two characters to account for apparent spacing anomalies created by specific character adjacency combinations. For example, 'ox' looks more widely spaced than 'oo' even though the character margins are the same.

Letter-spacing -- To place additional space between each letter of a word to better balance a line or emphasize a word. Pfont does not do letter-spacing (look at a newspaper, it usually looks terrible).

Logged disk -- Many microcomputers have more than one disk drive. The notion of 'logged disk' is used to indicate on which drive to search for files. The currently logged disk is displayed as part of the prompt, e.g., '>' indicates that drive 'A' is the logged disk.

MS-DOS -- A popular operating system for 16-bit computer systems. This is a generic version of the IBM DOS (PCDOS) operating system.

Operating System -- The first (lowest) level of software used with a computer system. This allows the user to use the basic functions of the computer, and most importantly, to use other software (such as *Fancy Font*). Popular microcomputer operating systems are CP/M and DOS.

Point -- Unit of measurement equal to 1/6th inch.

Pixel -- Picture element or dot.

Pica -- Unit of measurement approximately equal to 1/72nd inch. Usually used for vertical measurement.

Slab -- Letters with thick and thin strokes and serifs.

Scale -- To change the size of an image by reducing or enlarging.

Appendix 1 -- Glossary

Sans Serif -- Letters without serifs and usually having strokes of apparently even thickness.

Serif -- A smaller line used to finish off the main stroke of a letter, usually at the top or bottom of the stroke.

em -- The vertical stroke of a letter.

Style -- Fonts come in different styles, for example Roman, which has serifs, and Sans Serif, which does not. Other styles appearing in the manual are Old English and Script.

Subscript, superscript -- These are fonts that are higher or lower than normal fonts. In order to do super or subscripts simply switch into one of these special fonts (e.g., 'romn8p.fon' for superscripts or 'romn8b.fon' for subscripts).

Vertical Measure -- A term used by Pfont to indicate a vertical distance. A vertical measure must include a 'real' number and a unit. For example, '1.25i' to indicate 1 1/4 inches.

Word wrap -- To move words from the current line to the next line when the current line becomes too long to be fully printed.

Baseline -- The height of lowercase letters without ascenders or descenders.

--- Notes ---

F
T
a
t
f
W
s
R
(
F
V
C
D
P
t
o
f
a
T
H
I

Appendix 2 -- Font Descriptions and Samples

Font files generally adhere to the following naming convention.

`<fontname>.fon`

The 'fon' extension is assumed by Pfont if no extension is given to the arguments to the '+fo' parameter. For example, '+fo romn12' is equivalent to '+fo romn12.fon'. `<fontname>` can be further refined to the following form:

`<style>{<face>}<size>{<option>}`

where {<...>} indicates an optional element. `<style>` examples are 'romn', 'sans', 'olde', and 'scrp' for Roman, Sans Serif, Old English, and Script respectively. `<face>` is optional and can be 'i' or 'b' for italic and bold. We are a little loose about the term italic. We use it to modify the Roman style, which is not strictly accurate but is in common usage. Worse, we use it to describe 'slanted' Sans Serif fonts. These are characters that are just slanted Sans Serif characters, but do not have the normal italic script-like curves.) `<size>` is the size of the font in printers points (generally the distance from the bottom of the lower case descenders to the top of the upper case letters, but sometimes adjusted for proper overall appearance). `<option>`, if present, is a single letter indicating a font modification such as 'b' for subscript fonts, 'p' for superscript fonts and 'f' for fixed width (monospaced) fonts.

The standard distribution package contains the most popular fonts and many more are available from the *Fancy Font* library of fonts. The standard package contains the following fonts:

romn12.fon	Roman (serif) 12 point font.
romni12.fon	Roman Italic 12 point.
romnb12.fon	Roman Bold 12 point.
romn12f.fon	Fixed width (monospace) 12 point.
romn8b.fon	Subscript Roman 8 point.
romn8p.fon	Superscript Roman 8 point.
romn18.fon	Roman 18 point.
romn24.fon	Roman 24 point.
specl20.fon	Special characters; shading; copyright, etc.

The 'mkfonts' batch (or submit) file will create fonts in a variety of sizes in the Old English and Script styles.

The following pages show samples of the distributed fonts. Other font styles can be created using Cfont and the Hershey database or obtained from the *Fancy Font* library.

Appendix 2 -- Font Descriptions and Samples

amples of the distributed fonts follow.

-- Roman regular --

The quick brown fox jumps over the lazy dog.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxy

1234567890!"#\$%&'()_~|-^{\}[]+*::<>?.,/

-- Roman italic --

The quick brown fox jumps over the lazy dog.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxy

1234567890!"#\$%&'()_~|-^{\}[]+::<>?.,/*

-- Roman bold --

The quick brown fox jumps over the lazy dog.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxy

1234567890!"#\$%&'()_~|-^{\}[]+*::<>?.,/

-- Roman fixed width --

The quick brown fox jumps over the lazy dog.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxy

1234567890!"#\$%&'()_~|-^{\}[]+*::<>?.,/

-- Script --

The quick brown fox jumps over the lazy dog.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxy

1234567890!"#\$%&'()_~|-^{\}[]+::<>?.,/*

-- Old English --

The quick brown fox jumps over the lazy dog.

ABCDEFGHIJKLMNOPQRSTUVWXYZ

abcdefghijklmnopqrstuvwxy

1234567890!"#\$%&'()_~|-^{\}[]+*::<>?.,/

Appendix 2 -- Font Descriptions and Samples

Many different sizes of various fonts are available from the font library. The Roman fonts are typically available in sizes from 6 points to 24 points.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.



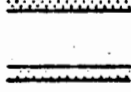
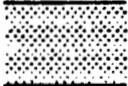

The quick brown fox jumps over the lazy dog.

The quick brown fox jumps over the lazy dog.

Appendix 2 -- Font Descriptions and Samples

following two pages the mapping between printed characters and the you type on your keyboard to print that character. The control characters (the 'control-X' key is shown here as '^X') can be entered using Pfont '\a' embedded command if your word processor does not allow to enter control characters into your text file. The program that produced these pages is called 'dispfont' and is available from the *Fancy Font* library. The font 'specl20.fon' contains characters used for shading and are mapped to keyboard keys as shown below. The following page shows the mappings for all of the Roman fonts and most of the other fonts available in the library.

SPECL20.FON
 Special chars, 20pt; 9/30/82
 Top row=2 Bottom row=-1
 Blank points (top)=7 Blank points (bottom)=2
 Page 1

(Key)	Ascii (Key)	Ascii (Key)	Ascii (Key)
() =	065 (A) = 	066 (B) = 	071 (G) = 
(R) = ↗	098 (b) =	103 (g) = 	104 (h) = 

Appendix 2 -- Font Descriptions and Samples

Ascii (Key)	Ascii (Key)	Ascii (Key)	Ascii (Key)
04 (^X) =	030 (^ ^) = ↑	031 (^_) = —	032 () =
03 (!) = !	034 (") = "	035 (#) = #	036 (\$) = \$
07 (%) = %	038 (&) = &	039 (') = '	040 (() = (
01 () =)	042 (*) = *	043 (+) = +	044 (,) = ,
05 (-) = -	046 (.) = .	047 (/) = /	048 (0) = 0
09 (1) = 1	050 (2) = 2	051 (3) = 3	052 (4) = 4
13 (5) = 5	054 (6) = 6	055 (7) = 7	056 (8) = 8
17 (9) = 9	058 (:) = :	059 (;) = ;	060 (<) = <
21 (=) = =	062 (>) = >	063 (?) = ?	064 (@) = @
25 (A) = A	066 (B) = B	067 (C) = C	068 (D) = D
29 (E) = E	070 (F) = F	071 (G) = G	072 (H) = H
33 (I) = I	074 (J) = J	075 (K) = K	076 (L) = L
37 (M) = M	078 (N) = N	079 (O) = O	080 (P) = P
41 (Q) = Q	082 (R) = R	083 (S) = S	084 (T) = T
45 (U) = U	086 (V) = V	087 (W) = W	088 (X) = X
49 (Y) = Y	090 (Z) = Z	091 ([) = [092 (\) = \
53 (]) =]	094 (^) = ^	095 (_) = _	096 (') = '
57 (a) = a	098 (b) = b	099 (c) = c	100 (d) = d
61 (e) = e	102 (f) = f	103 (g) = g	104 (h) = h
65 (i) = i	106 (j) = j	107 (k) = k	108 (l) = l
69 (m) = m	110 (n) = n	111 (o) = o	112 (p) = p
73 (q) = q	114 (r) = r	115 (s) = s	116 (t) = t
77 (u) = u	118 (v) = v	119 (w) = w	120 (x) = x
81 (y) = y	122 (z) = z	123 ({) = {	124 () =
85 () =	126 () =		

Appendix 3 -- Using *Fancy Font* with Word Processors

Fancy Font will print any standard ASCII file regardless of how it was produced. However, if your word processing software maintains its own printing system or includes its own special character codes in files, then it will not work properly with *Fancy Font*. Most word processing software, as well as data base management systems and spreadsheet programs produce standard ASCII files and thus work well with *Fancy Font*. In some cases, special *export* commands are available to turn a non-standard word processing file into a standard ASCII file -- consult your word processing manual.

Fancy Font has been designed to accommodate word processors that 'turn on' the 'high order bit' of certain characters. Thus, the *export* procedure mentioned above is not necessary for WordStar or similar word processors.

Currently available personal computer text editing and word processing software does not consider the different horizontal and vertical sizes of different fonts. Nor does it consider the widths of characters within a given font set. Therefore the formatting features of these text editors are not too useful when printing with *Fancy Font*.

To overcome this deficiency in existing personal computer software, *Fancy Font* provides its own horizontal and vertical formatting features, including word wrap and justification. Thus, it is advisable to disable all formatting functions of your word processor, and use those of *Fancy Font* instead.

The procedure for printing with *Fancy Font* is as follows:

1. Create a text file, possibly containing *Fancy Font* embedded commands for font selection and formatting.
2. Print the contents of the text file using Pfont, possibly specifying a number of printing parameters.

When creating the text file, the user should *not* use justification, centering, lineation or other formatting features of the text editing or word processing software.

If you are using WordStar, do the following:

Enter the *document* mode.

Turn justification *off* (Control-O J).

Use WordStar in the normal manner, however insert *Fancy Font* embedded commands to select fonts, center, justify, word wrap, and in general in place of *dot* commands. You will not actually see the results of these commands until you print the text. Do not use tabs (as WordStar expands these to spaces), but use the

Appendix 3 -- Using *FancyFont* with Word Processors

\t command instead.

When finished editing, save your file in the normal manner. Print the source text file you just created by using Pfont.

If you try to print a file created by the WordStar print feature, use the Pfont '-PG' parameter so that Pfont will not try to paginate your text. This is *not* recommended as WordStar will not be able to properly paginate your text due to different height and width *FancyFont* characters.

To summarize, WordStar and most other word processors should be used primarily for *editing* text and inserting *FancyFont* commands. *FancyFont* can then be used to *format and print* your text.

— Notes —

P

Th

me

lik

th

W

inf

—

3)

no

Se

tin

ch

re

'1

Th

th

at

—

—

—

—

—

—

—

—

—

—

—

—

—

—

—

Appendix 4 -- Pfont Messages

font Messages

This appendix lists the messages displayed by Pfont. Three types of messages are identified: 1) Error — an error has occurred that will most likely cause undesirable results. In some cases, *Fancy Font* will recover from the error properly; *Fancy Font* will almost always continue printing. 2) Warning — *Fancy Font* has encountered an unusual circumstance and displays information regarding the circumstance. You don't necessarily need to act but should consider if *Fancy Font* is behaving in the manner you expected. 3) Informational — *Fancy Font* is presenting some information that represents normal processing.

Several messages contain words from your text or words determined at the time the message is displayed. In this appendix, such words are characterized by bracketed words; for example, <number> is used to present some actual number and may appear in a displayed message as '0'.

The messages are listed in alphabetical order based on the first character of the first word in the message. If the first word is of the type described above, then the message will be found at the beginning of the list.

The Messages

— more — press any key to continue

Informational

Description: More information is to be displayed than can fit on the screen at one time. At the end of each screenfull this message is displayed.

Action: Press any key to allow the remaining information to be displayed. The profile 'FancFont.pro' allows you to indicate the size of your display to make sure this measure appears when appropriate.

PfontOx.ovl> system file not found.

Error

Type 'Control-C' to Quit, OR switch disks and type any other character to retry>

Description: CP/M users only, one of the PFont overlay files can not be located. Pfont is divided into several sections in order to leave as much room in memory as possible for loading of fonts. Thus there are several files that comprise PFont. Each of these files must reside either on the currently logged disk or on the disk in drive A.

Action: Make sure the indicated file is on the currently logged disk (switch disks if necessary) and type any character to continue. If

Appendix 4 -- Pfont Messages

you wish to quit, type a 'Control-C'.

<command> command absolute only; + or - ignored, assuming absolute.

Error

Description: The <command> indicated should not be followed by a value that includes a '+' or '-' sign. The command is intended to be absolute, not relative to some previous setting. The '+' or '-' sign is ignored.

<command> ignored.

Error

Description: The indicated <command> was not recognized as a valid command and thus could not be processed. The command was skipped and will have no effect on the current use of *Fancy Font*.

<command> not recognized as a command - ignored.

Error

Description: The <command> is not known by Pfont and hence cannot be processed. The indicated characters are ignored and not printed.

** too long for a font file name (maximum <size> characters)**

Error

Description: The contains too many characters and thus cannot be a valid font name. The maximum number of characters is <size>.

Action: Retype the using the appropriate number of characters.

<name> not a valid font file

Error

Description: The file <name> was found on the disk, but it is not a *Fancy Font* font file.

Action: Make sure the file is a good copy of a *Fancy Font* font file - recopy if necessary.

<name> not valid, '*' names not allowed

Error

Description: The indicated <name> contains the character '*' and is thus not a valid file name.

Action: Do not use a file name that includes a '*'.

<name>: No more room in memory (<number>)

Error

Description: The Pfont program has run out of computer memory and can not continue.

Action: Use fewer fonts and/or shorter substitution strings and header and footer lines. Your computer system must have a minimum of 128K bytes memory (DOS) or 64K bytes (CP/M).

Appendix 4 -- Pfont Messages

<name>: No more room in memory (<size>) *Error*

Description: Pfont has run out of available memory and can not continue. The program was attempting to find <size> memory characters for <name> and could not find enough memory.

Action: Use fewer fonts, substitution strings, header and footer lines and retry. If problems persist, contact SoftCraft, Inc.

<number> digits required for <command> - command ignored. *Error*

Description: Exactly the indicated <number> of digits is required for the indicated <command>. The command cannot be processed and thus has been completely ignored.

Action: Modify your text file, making sure to use the indicated number of digits.

<number> digits required for <command> command - assuming leading 0's. *Error*

Description: Exactly the indicated <number> of digits is required for the indicated <command>. Processing of the command continues, withing 0's inserted before the number to create the necessary number of digits.

<number> is not between <lower bound> and <upper bound> *Error*

Description: The indicated <number> is either less than <lower bound> or greater than <upper bound>.

Action: Use a number that is within the indicated range (inclusive).

<text> doesn't match expected values *Error*

Description: The indicated <text> doesn't match the names of any parameters and thus could not be processed by Pfont.

Action: Use a known parameter name. Pressing '?' will indicate the known parameter names.

<type> was expected *Error*

Description: An item of the indicate <type> was expected, e.g. Real, Horizontal Measure, etc.

Action: Use a valid element of the indicated type.

<word> not included in list of expected inputs *Error*

Description: The indicated <word> was not expected by Pfont and therefore could not be adequately processed.

Action: Retype the word, or use '?' to determine the list of expected input words.

Can't find the file: fancfont.pro *Error*

Description: The file FancFont.pro is neither on the currently logged

Appendix 4 -- Pfont Messages

disk nor is it on the disk in drive A. This file must be available in order for *FancyFont* to operate properly.

Action: Make sure the file FancFont.pro is on the currently logged disk or on the disk in drive A and retry.

Can't open '<file name>'.

Error

Type 'Control-Z' to Quit, or Switch Disks and type any other character to retry>

Description: The indicated <file name> is not available on the logged or named disk drive.

Action: Switch diskettes so that the indicated file can be found by Pfont. Press any key to continue (after switch diskettes) or press the Control and Z keys to quit (Control-C for CP/M).

enter command within center or right align command - ignored.

Error

Description: A center command appears following a center command. This combination cannot be processed by Pfont and hence the second center command is ignored and is not printed.

Action: Remove the second center command or end the first center command with a \h command or a new line.

Command Indicator set to <character>

Informational

Description: <character> is the *FancyFont* command indicator for this use of Pfont. Only characters preceded by this <character> will be processed as Pfont embedded commands.

Continuing input from parameter input file . . .

Informational

Description: The remainder of the parameter input file will now be processed. This occurs following a parameter error and after repair of the error.

Default Parameter: +<parameter>

Informational

Description: If no parameter is specified (i.e., no + or - preceding a parameter name), then the input will be assumed to be preceded by the indicated <parameter>.

Disk read error, end of file?

Error

Description: An error has occurred in the process of accessing information contained in a file on the disk.

Action: Check the file to see that it is complete and not damaged. If possible, make a new copy of the file and make sure that the end of the file is properly marked according to the standards of your operating system.

Appendix 4 -- Pfont Messages

Do you really wish to quit now ('y' or 'n')? *Informational*

Description: PFont can be interrupted if you type 'Control-Z' (CP/M users type 'Control-C'). After the interruption, this message is displayed to allow you to reverse your decision.

Action: If you indeed want to stop using PFont, press 'y'. Press 'n' to continue using PFont.

Ending Pfont. *Error*

Description: An error has occurred from which Pfont cannot recover, therefore Pfont is stopping.

Action: Repair the error and rerun Pfont.

Enter digit between '0' and '<number>' for horizontal draft mode: *Informational*

Description: In response to the 'Control-D' interactive request, Pfont needs to know the desired rough draft mode.

Action: Depress a numbered key between 0 and <number> inclusive to indicate the desired rough draft mode.

Enter lines to be printed using keyboard. *Informational*

Type 'Control-Z' 'Return' when done.

Description: Pfont is being used in its electronic typewriter mode (a file name of 'tty:' or 'con'). This message indicates that you should type lines of text to be printed. Whenever you press the 'Return' key, the line will be printed. To finish printing and end Pfont, press the 'Control-Z' combination and then press the 'Return' key. There must be no other typing on the ending line.

Enter printing parameters, for example: *Informational*

+FO Romn12 RomnB12 +FI File.ff

Then press RETURN (possibly more than once) to start printing.

Type '?' for assistance at any time.

This message provides an example of what you should type when using Pfont. '?' can be used frequently to indicate your choice of input.

Expecting <Parameter Value> - a <type> between <number> and <number>. *Informational*

Current Value: ...

Description: The value for the indicated parameter is expected. <type> and 'between <number> and <number>' provide additional clues as to what is expected. The <number>'s represent the inclusive bounds for the value expected. If the parameter in question has a current or default setting, it will be indicated following the words

Appendix 4 -- Pfont Messages

'Current Value:'.

Expecting name of parameter input file.

Error

Description: The name of a file containing Pfont parameters was expected and was not received.

Action: Try typing a different name, or a similar name that also includes a specific disk drive (e.g. A:name).

The 'looping' without Files, can't continue.

Error

Description: You have used the special file name 'loop:' to indicate repeated printing of files, but have not indicated any files to be printed.

Action: Use Pfont again, this time making sure to indicate a file to be printed before using 'loop:'.

The <file name> can't be opened

Error

Description: The <file name> is not available for use. *Fancy Font* was unable to find and use the file.

Action: Check that you have typed the name correctly; also be sure to prefix the file name with a disk drive name if not on the currently logged disk. For example, A:name or B:name.

The <name> not found

Error

Description: The indicated <name> could not be located on the logged disk drive.

Action: Try entering a different name, or similar name prefixed with a disk drive specification (e.g. B:name). If no success, stop using the *Fancy Font* program and make sure the exact <name> you are using is on the currently logged disk.

Font file <file name> not available -- can't continue.

Error

Description: The indicated font <file name> could not be located on your disk and therefore Pfont cannot continue.

Action: You have changed diskettes after indicating Pfont parameters and removed the diskette containing the <file name>. Rerun Pfont and do not switch diskettes. If necessary, copy fonts and/or files so that you won't need to swap diskettes.

Font file <name> not found

Error

Description: The font file <name> could not be located on your computer's disks.

Action: Check and retype the name, making sure to use a disk drive name prefix if the font file is not on the currently logged disk (e.g., A:name or B:name).

Appendix 4 -- Pfont Messages

Horizontal Mark <number> not set - command ignored. *Error*

Description: The horizontal mark number <number> has not been set by a \m command and thus cannot be used.

Action: Be sure to use \m<number> before using \hm<number>.

Horizontal draft mode <number> set. *Informational*

Description: The horizontal rough draft mode number <number> has been successfully set in response to the 'Control-D' interactive request.

Improper numeric argument for <command> - command ignored. *Error*

Description: The value following the indicated <command> does not conform to the expected value (for example, not a number). The entire command is ignored.

Input line too long. *Warning*

Description: Pfont has received too many characters on one input line. Subsequent characters will be ignored.

Action: Start a new input line.

Insert or adjust paper - then type any character to print next page. *Informational*

Description: Pfont has paused and is waiting for your approval before the next page is printed.

Action: Press any key on the keyboard and the next page will be printed.

Invalid printer type in the file fancfont.pro *Error*

Description: You have entered a printer type in the file FancFont.pro that is not recognized by your version of *FancyFont*.

Action: Inspect the file FancFont.pro, check that the printer type is properly specified. If trouble persists, copy the FancFont.pro file from your distribution diskette. You might also try a synonym or more generic name for your printer.

Invalid y density request: <number> *Error*

Description: The font you are using does not conform to the format of font file expected by your version of *FancyFont*.

Action: Recopy the font file from your master diskette; it is likely damaged.

Line length (<line length>) exceeds maximum length (<max length>), <number> characters. *Warning*

Description: The current line (displayed preceded by '***' following this message) does not fit on the print line. The length of the line

Appendix 4 -- Pfont Messages

(<line length>) is longer than the maximum <max length>. The line is <number> characters long.

Action: Use word wrap (\w) and Pfont will automatically move words between lines to avoid this problem. Alternate solutions are to reduce the number of characters to be printed on the line or to increase the line width (+LW).

Loading as font number <number> . . . in memory

Informational

Loading as font number <number> . . . on disk

Description: Pfont is preparing for printing by reading the character definitions from the file into memory. Printing is much faster if the font fits in memory. If the font does not fit, the message '... on disk' is displayed to indicate that the character definitions must be used directly from the disk. The is successfully loaded either in memory or on disk as font number <number> for use with the embedded \f command.

Make sure printer is 'On Line'

Warning

Description: Pfont is ready to start printing, but you must have your printer properly connected to your computer, have the printer plugged in, turned on and have the appropriate button or switch on your printer 'on line'.

No FF fonts -- command ignored.

Error

Description: You can not use the \ff command since the '+FO' parameter has not been used to indicate the correspondence between Fancy Font font numbers and font names.

Action: Specify the '+FO' parameter and rerun Pfont.

No more room in list (maximum <number>), <text> not used.

Error

Description: Too many items have been placed in a list of values for a Pfont parameter. The message indicates the maximum number of items in the list and the item that would not fit. The indicated item will be ignored.

Action: User fewer items, for example, fewer fonts or file names. In some causes you must use Pfont multiple times, in others you must combine multiple files into one file to continue. The Pfont list limits are reasonably large; thus this message may result from a usage error.

No unit specified for <command> command -- assuming '<unit>'. *Error*

Description: The indicated <command> should be followed by a value that includes a unit. The command was not followed by such a unit; so the indicated <unit> was assumed and processing continues

Appendix 4 -- Pfont Messages

using this unit.

No unit specified, assuming <unit>.

Warning

Description: A unit ('i', 'c', 'd', 'u') was expected for a horizontal or vertical measure. The unit was not found, however the indicated unit was used.

Not enough room for the font set arrays

Error

Description: The amount of available memory on your computer system is not adequate to hold the amount of information necessary to use all of the fonts you have requested, even if they reside on disk.

Action: Reduce the number of fonts for the document and retry. Some space is taken up by substitution strings, header and footer lines and file names (+SU, +HL, +FL, +FI); if these are cut down to a minimum you may be able to use more fonts.

Note: One or more lines printed with extra interline spacing. *Informational*

Description: There is extra white space between at least two printed lines in your document.

Action: This is an informational message and you need not correct the situation in most cases. The extra space is the result of the definition of the fonts used on the print lines. You must use the 'a' command in Efont to properly set the 'blank points' for a font. The blank points indicate the extra, unused white space in a font and is taken into account when calculating interline spacing. Additionally, fonts that have a very tall or very deep character must leave room for that character. Thus extra white space will be left for all characters of such fonts. Again, the 'a' command of Efont will indicate the tallest and deepest characters of a font. This message will also be displayed if you attempt excess negative vertical motion with either \v or \l when using -RV (see \l, \p, and RV parameter).

Nothing left to delete.

Warning

Description: There are no parameters as yet unprocessed during Pfont error recovery, thus you can not delete any parameters.

Number is too long

Error

Bad Integer in the profile, FANCFONT.PRO just before: ...

Description: The file FancFont.pro contains either an unexpected number, or a number that is too large.

Action: Inspect the file FancFont.pro, look for the text in the file displayed in the above message and determine if a reasonable number has been specified. If not, correct the number. Otherwise, make a copy of FancFont.pro from your distribution diskette and

Appendix 4 -- Pfont Messages

retry.

Page <page number>, <number> Lines *Informational*
Description: Page <page number> has been printed, it contained <number> lines of output.

Page length (<Vertical Measure>) is shorter than Top and Bottom margins (<Vertical Measure>). Page Length set to <Vertical Measure>. *Error*

Description: The page length specified with the '+PL' parameter is shorter than the sum of the lengths of the '+TM' and '+BM' parameters and thus Pfont can do no printing. Therefore, Pfont has set the page length to the indicated length.

Parameter Input File> *Informational*

Description: The name of a file containing Pfont parameters is expected.

Action: Enter the name of a file, followed by a press of the RETURN key. Make sure to prefix the name with a disk drive specification (e.g. A:name) if the named file is not on the currently logged disk drive.

ast last tab on line, tab ignored. *Error*

Description: A tab command (\t or 'Control-I') has been encountered while formatting that has no corresponding tab stop set. That is, there is not tab stop to the right of the position in the print line at which the tab was encountered.

Action: Modify tab stops with the '+TB' parameter, remove the tab from the text, or use \h or \i to achieve the desired horizontal motion.

lease answer 'y' or 'n': *Error*

Description: You have pressed a key other than 'y' or 'n' in response to a question requiring a yes or no answer.

Action: Press 'y' to answer yes, 'n' to answer no.

rint with Fonts, version x.x *Informational*

Copyright (C) 1984 SoftCraft, Inc.

Description: Introductory message. The number identifies this version of the program and helps you to determine whether you have the most recent revision.

rinting <file name> ... *Informational*

Description: The indicated <file name> is currently being formatted and printed. The <file name> may be 'tty:' or 'con' if the special electronic typewriter mode is in use.

Appendix 4 -- Pfont Messages

Printing terminated after page <page number> *Informational*
Description: Pfont is ending after the indicated <page number>. This occurs when the '+LP' parameter is used.

Printing will pause after current page. *Informational*
Description: In response to the 'Control-E' interactive request, Pfont will pause after printing the end of the current page.

Profile backspace string too long *Error*
Description: The file FancFont.pro allows you to customize how your computer system responds to a backspace character. This message indicates that the indicated backspace specification exceeds the maximum 6 character length.
Action: Modify the file FancFont.pro to contain a shorter backspace string specification and retry. If this problem persists, copy the original FancFont.pro file from your distribution diskette.

Real argument value *Partial*
Description: A real number is expected. That is, a number that may have a fractional part (e.g. 1.45, 2, .56).

Remaining input: <parameters> *Informational*
(plus rest of parameter input file)
Insert input or press 'RETURN' to continue
Control-D to delete next input word, Control-B to backup
Control-X to ignore all remaining input and continue
Description: An error in specifying Pfont parameters has occurred. The <parameters> have not yet been processed. The rest of the parameter input file (if any) also has not yet been processed. You have several options to repair the parameter error and continue to process all the remaining parameters.
Action: Type new input to replace the erroneous input or use Control-D, Control-X or Control-B to repair, ignore etc. the remaining input. A 'return' will continue processing the remaining input.

Result (<number>) of <command> command not between <number> and <number> - command ignored. *Error*
Description: The value resulting from the indicated <command> is not within the allowable limits. The resulting value is indicated by the first <number> and the second and third <number>s represent the limit. The entire command is ignored. The resulting <number> may be different than the value indicated in the <command> due to unit conversion or a relative command.

Appendix 4 -- Pfont Messages

Right Alignment command within Right Alignment command - ignored. *Error*

Description: A right alignment command appears within a right alignment command. This combination cannot be processed by Pfont and hence the second right alignment command is ignored and not printed.

Action: Remove the second right alignment command or end the first right alignment command with a \h or a new line.

Right Margin (<Horizontal Measure>) larger than maximum (<size>), reset to maximum. *Warning*

Description: The sum of the line width and the left margin exceeds the maximum printing width (<size>) for your printer. The line width has been changed to fit the <size> of your printer.

Action: If you have a wide printer, modify your profile to so indicate. For Epson printers, modify the file 'FancFont.pro' so the 'printer.type' is XX100 rather than XX80 (where XX is FX, RX, or MX).

High draft mode error *Error*

Description: An error has been encountered in reading information from a font file.

Action: It is likely that you are using a damaged font file. Make a new copy of the font file and retry.

Low draft mode error *Error*

Description: An error has been encountered in the process of accessing a character from a font that is stored on disk.

Action: It is likely that you have a damaged font file. Make a new copy of the font file, and retry.

Skipping to next file. *Informational*

Description: In response to a 'Control-F' interactive request, the remainder of the current file will be ignored and subsequent input text will be read from the next file in the '+FI' file list.

Skipping to page <page number>. *Informational*

Description: In response to a 'Control-N' interactive request, all pages until the page numbered <page number> will be skipped. ('Control-P' for CP/M).

Missing argument value *Informational*

Description: A string of characters is expected.

Appendix 4 -- Pfont Messages

ring number <digit> not defined -- command ignored. *Error*

Description: The substitution string (\s) numbered <digit> has not been defined with a +SU parameter and thus cannot be used. The \s command has been ignored.

Action: Use Pfont again, this time making sure to use +SU as a parameter to define all the \s substitution strings used in your text.

ie Fancy Font System -- loading Pfont . . . *Informational*

Description: Your system may take some time loading the Pfont printing program. This message indicates that Pfont is being loaded, and that you must wait until it is ready for operation.

Unexpected end-of-file *Error*

Description: *Fancy Font* has prematurely encountered the end of a file it is trying to use.

Action: Check that all font and other files have been properly copied. Attempt to determine the file causing the error, make a new copy of the file if possible, and retry.

Unknown FF Font -- font 0 selected. *Error*

Description: In switching from native fonts (\fn) to Fancy Font fonts (\ff), the current font number was set to a number not available for Fancy Font fonts. In order for printing to continue, Pfont is choosing to use Font 0 until instructed to do otherwise with a subsequent \f command.

Action: Switch fonts *before* switching from native font printing to Fancy Font printing.

Unknown font () -- font not changed. *Error*

Description: The font has not been specified with the '+FO' parameter and hence cannot be used. Pfont will not make the requested font change.

Action: Specify the '+FO' parameter and rerun Pfont.

Unknown string substitution command '<character>' -- command ignored. *Error*

Description: In a text file being processed by Pfont, there is a \s substitution command that is not recognized. Examples of valid commands are \sf, \s#. The <character> following \s in the file is not recognized and the entire \s command is ignored.

Vertical Mark <number> not set -- command ignored. *Error*

Description: The vertical mark number <number> has not been set by a \n command and thus cannot be used.

Action: Be sure to use \n<number> before using \hm<number>.

Appendix 4 -- Pfont Messages

Warning – Bottom Margin (<Vertical Measure>) less than Footer Margin (<Vertical Measure>). Warning

Description: The bottom margin ('+BM' parameter) is not large enough to contain the footer margin ('+FM' parameter). Results of printing will be unpredictable.

Action: Increase the size of the bottom margin or decrease the size of the footer margin and rerun Pfont.

Warning – No fonts specified, choosing native printer fonts. Warning

Description: The '+FO' parameter has not been used, therefore all printing will be with the fonts native to your printer rather than the high quality *Fancy Font* fonts.

Warning – Right Margin (LW + LM) too large to allow for slanted characters; Warning

suggest reduction by approx. %d dots.

Description: The right margin (sum of line width and left margin) is too large to allow slanted characters to fit on the line. Since Pfont allows characters to have negative margins, some characters (such as a Roman 'f' and most script and italic characters) extent slightly past the right margin. This message indicates that there may not be enough room for such characters to extend into the right margin.

Action: As recommended in the message, reduce either the line width or the left margin to by the indicated number of dots to allow room for character overhangs.

Warning – Top Margin (<Vertical Measure>) less than Header Margin (<Vertical Measure>). Warning

Description: The top margin ('+TM' parameter) is not large enough to contain the header margin ('+HM' parameter). Results of printing will be unpredictable.

Action: Increase the size of the top margin or decrease the size of the header margin and rerun Pfont.

Warning – no printing or display, 0 copies requested. Warning

Description: You have indicated '+NC 0' or '-NC' as a Pfont parameter. This results in no printing or processing and is most likely a mistake.

Action: Run Pfont again without the 'NC' parameter.

Warning: Native/Fancy font selection not at beginning of line – Warning
formatting errors likely.

Description: A selection of \ff for Fancy Font fonts or \fn for native printer fonts should occur at the beginning of a print line. If not,

Appendix 4 -- Pfont Messages

formatting problems are likely since Pfont will assume that all printing is to be done in the last type of fonts specified.

Action: Use a `\b` immediately preceding the `\ff` or `\fn` command.

Warning: this font is not compatible with this version of the program

Warning

Description: The font is for a different version of the program and will not work properly with the version of the program you are using.

Action: Use a different version of the program or a different font. In some cases, different font versions may be used to achieve unusual effects.

Word split, too long for print line.

Warning

Description: When using word wrap (`\w`), a word is so long that it cannot fit on a printed line without being split. Pfont splits the word, putting the first part on the current print line and the remainder on the next print line.

Action: Avoid using very long words and very short line lengths. Alternatively, hyphenate the word by placing `'- '` (a hyphen followed by a space) within the word.

.. for help, ... (or ...) for file input, ... to review parameters, Informational
'Control-V' for general help, 'RETURN' to end list or start processing, or
'Control-U' to erase input line. Type 'Control-Z' to abort and exit.]

Description: An informational message to remind you of the special characters used by Pfont. The first character can be used for help, the second and (third) to obtain parameter input from a parameter input file, and the fourth to review the current settings of all the parameters. The remaining lines indicate additional special characters. (CP/M users should press 'Control-C' instead of 'Control-Z').

Appendix 4 -- Efont Messages

font Messages

ready exists. Do you want to overwrite it

When you try to create a file (e.g., in the *save* or *edit* commands) and a file of that name already exists you will be offered the opportunity to overwrite the old file or give a new name. Answer 'y' to the prompt if you want to overwrite the old file or 'n' if you do not. In the latter case the program will request an alternative name.

blank points (top)=<integer>, blank points (bottom)=<integer>

This is an informational message indicating the 'blank points' values for a font. See section E.2 for a discussion of these values.

can't create <filename>; try another name.

If you try to create a file (e.g., in the *save* or *edit* commands) and that file cannot be created then this prompt will allow you to give another name. Most likely you are out of space on the disk or you typed an invalid file name. Give another file name, perhaps on a different drive (e.g., b:newfile.fon).

can't find the file: fancfont.pro

Efont needs to read the file 'fancfont.pro' when it starts up in order to determine what type of printer is being used. Make sure 'fancfont.pro' is either on the currently logged drive or on drive A.

can't open file: <filename>; try another name.

If, when trying to read a file (e.g., in the *load* or *edit* commands), that file cannot be opened this message and prompt will allow you to try a different file name. You may have mistyped the file name. Try to type it again. Be sure to include the drive specifier (e.g., b:myfont.fon) if the file is on a disk other than the currently logged drive.

can't swap in <filename>; is it on the default disk?

This arises on CP/M systems only. The Efont overlay files (EFONT*.OVL) must be on the current disk or on drive A. You should copy the overlay files to the current disk or to drive A, or log to the disk containing the overlay files before starting up Efont.

changing the right margin of character #<integer> to <integer>

Negative overall character widths are not allowed. When negative left or right character margins (see the *margins* command) would result in a character with negative width this message is displayed indicating that the right margin has been reset to make the character have width zero. No action is required.

Appendix 4 -- Efont Messages

Character 'A' is not the highest character

Top blank points is computed for a font only if 'A' is the tallest character and 'p' is the lowest. This message is displayed simply to inform you that these values are not being computed. See the *auxinfo* command and section E.2 for a discussion of blank points values.

Character 'p' is not the lowest character

Top blank points is computed for a font only if 'A' is the tallest character and 'p' is the lowest. This message is displayed simply to inform you that these values are not being computed. See the *auxinfo* command and section E.2 for a discussion of blank points values.

Character is too high, y=<integer>

During a *replace* command a character was found to have dots in it at a vertical location exceeding the maximum value. Section E.3 describes the maximum character size. The character will be clipped down to the allowable size. If the character is less than one inch in overall height, reduce the *ytop* value to an allowable value and *replace* again.

Character is too low, y=<integer>

During a *replace* command a character was found to have dots in it at a vertical location lower than the minimum value. Section E.3 describes the maximum character size. The character will be clipped down to the allowable size. If the character is less than one inch in overall height, increase the *ytop* value and *replace* again.

Character is too wide, x=<integer>

During a *replace* command a character was found to have dots in it at a horizontal location exceeding the maximum value. Section E.3 describes the maximum character size. The character will be clipped down to the allowable size. If the character is less than one inch wide you probably have blank columns at the left edge of the '.edf' file. Move the asterisks left to the left edge of the file and *replace* again.

Can't find any characters in that range to replace

The range of characters provided during the *replace* command did not correspond to any 'a<nnn>.edf' files on the current disk. Perhaps you need to be logged to a different drive (i.e., to the one containing the '.edf' files) before starting Efont.

Appendix 4 -- Efont Messages

Disk read error, end of file?

There was an error in trying to read from a font file. The disks containing the font files must remain in the computer throughout execution of Efont so make sure you haven't removed any disks after starting Efont. This message may also indicate a bad font file. See the 'Shouldn't happen; bad alloc' message for options in the latter case.

Disk write error. Out of space?

You are probably out of space on the disk to which a file is being saved. If this occurs during a *save* command try to save to a different disk (e.g., b:myfont.fon). If this occurs during a *edit* operation you will need to save the current font if necessary, get out of Efont, make space available or log to another disk, and then reenter Efont and try again.

Expecting an integer

This message is displayed when the program is reading from a file and is expecting to find an integer. If the message occurs immediately upon startup of the program then there is probably a syntax error in the profile file, 'fancfont.pro'. If you have recently modified this file check it against the original and make sure all integers look right (e.g., make sure you don't have 'oh' for 'zero') and make sure the commas and spaces are still present in the file. If this message occurs during a *replace* command the context of the message should identify where the problem is. Check the first three numbers in the replacement file. See the *replace* command for more discussion on '.edf' file format.

Invalid character, <hex> (hex), ignored; it should be an asterisk or a space.

The file read by the *replace* command should have only asterisks, spaces, carriage returns and line feeds in it. This message identifies an unexpected character in the replacement file. See the *replace* command for more discussion on file format.

Invalid command indicator, '\...'

Only a small number of Pfont embedded commands are allowed in the *teststring* command. See the *teststring* command for a list of allowable commands. You probably had a backslash '\' in the string to be printed followed by something other than one of the allowable commands.

Appendix 4 -- Efont Messages

Invalid printer type in the file fancfont.pro

The printer type in the file 'fancfont.pro' is not one of the allowable types. Check to see that the string of characters between 'printer.type' and the following comma is one of the allowable printer types (see Appendix 5 for the list of allowable types).

Invalid y density request: <integer>

This probably indicates a bad font file. See the 'Shouldn't happen; bad alloc' message for options.

Negative character widths are not allowed.

Negative overall character widths are not allowed. When negative left or right character margins (see *themargins* command) would result in a character with negative width this message is displayed indicating that the right margin has been reset to make the character have width zero. No action is required.

No more room

There is not enough memory left to *replace* any more characters. Try to save the font as it is, exit Efont, restart Efont, load the font and then continue replacements. If the font is simply too large to fit into memory (quite possible on CP/M machines) there is nothing that can be done; you must split the font in two and use it as two separate fonts (i.e., save the font as it is and do the rest of the replacements in another run of Efont and then save as another font).

Not a full ASCII font set

Top blank points is computed for a font only if 'A' is the tallest character and 'p' is the lowest. This message is displayed simply to inform you that these values are not being computed. See the *auxinfo* command and section E.2 for a discussion of blank points values.

Not a valid font file

All fonts begin with some identifying information to distinguish font files from other files. This information was not there for the font you just tried to load. This probably indicates a bad font file. The file may have been copied incorrectly from the original disk. If it was generated from Cfont you may have run out of space during the font generation phase (there would have been a message to that effect). Try to recopy the font from the original disk or regenerate it from Cfont.

Appendix 4 -- Efont Messages

Not computing blank top points

Top blank points is computed for a font only if 'A' is the tallest character and 'p' is the lowest. This message is displayed simply to inform you that these values are not being computed because either 'A' or 'p' is not defined in the font. See the *auxinfo* command and section E.2 for a discussion of blank points values.

Not enough memory for font editing

You do not have enough memory in your computer to even begin execution of Efont. There is nothing to do but try to run it on a computer with more memory available to application programs.

Not enough memory for the whole font: nothing loaded

The size of the font you tried to load is larger than the available memory. Sometimes a font can be loaded if the *load* command is the first command issued in a run of Efont, so you might try to start Efont afresh and load the font again.

Not enough room for the font set arrays

You do not have enough memory in your computer to even begin execution of Efont. There is nothing to do but try to run it on a computer with more memory available to application programs.

Number is too long

This message is displayed when the program is reading from a file and is expecting to find an integer. If the message occurs immediately upon startup of the program then there is probably a syntax error in the profile file, 'fancfont.pro'. If you have recently modified this file check it against the original and make sure all integers are less than 20 digits long. If this message occurs during a *replace* command the context of the message should identify where the problem is. Check the first three numbers in the '.edf' file.

Profile backspace string too long

The list of hex digits provided for 'backspace.output.string' in the profile file 'fancfont.pro' is too long; it must be four bytes or less. If you have modified this file recently check the syntax of the current file against the original and make sure the list of bytes provided for backspace string is length 4 or less.

Enough draft mode error

This probably indicates a bad font file. See the 'Shouldn't happen; bad alloc' message for options.

Appendix 4 -- Efont Messages

Shouldn't happen; bad alloc calculation

This probably indicates a bad font file. The file may have been copied incorrectly from the original disk. If it was generated from Cfont you may have run out of space during the font generation phase (there would have been a message to that effect). Try to recopy the font from the original disk or regenerate it from Cfont.

Shouldn't happen; the bitmap for 'A' is wrong

This probably indicates a bad copy of the Efont program. Try to recopy from the original disk. This message may occur intentionally (and without ill effect) when Efont is used in 'Auto Replace' mode.

Shouldn't happen; the bitmap for 'p' is wrong

This probably indicates a bad copy of the Efont program. Try to recopy from the original disk. This message may occur intentionally (and without ill effect) when Efont is used in 'Auto Replace' mode.

Sorry, no default here. Enter a number between <integer> and <integer>>

The previous prompt for a number does not allow for carriage return (or enter) alone as a response. That is, there is no default value for this prompt. You must type in a number explicitly.

The font height has changed. The new values are:

A *replace* or *zap* command has caused the overall height of the font to change (see the command descriptions for reasons why this happens).

There are no command line parameters required

You typed some parameters on the command line when you started Efont. All Efont commands are interactive and no parameters are required. Simply continue in Efont after this message.

Top row=<integer>, Bottom row=<integer>

See section E.2 for the meanings of these numbers for a font.

Unexpected end-of-file

There is something wrong near the end of the file 'fancfont.pro'. If you have recently modified this file check it against the original and make sure all integers look right (e.g., make sure you don't have 'oh' for 'zero') and make sure the commas and spaces are still present in the file.

Appendix 4 -- Efont Messages

Warning: this font is not compatible with this version of the program

Each printer type requires a different version of the programs (Pfont, Efont and Cfont). The font you tried to load was not made for this version of Efont.

Appendix 4 -- Cfont Messages

Cfont Messages

Actually, not enough memory to print at any scale factor

Your computer does not have enough memory to use the *print* command. There is nothing that can be done.

Already exists. Do you want to overwrite it

When you try to create a file (e.g., in the *save* or *generate* commands) and a file of that name already exists you will be offered the opportunity to overwrite the old file or give a new name. Answer 'y' to the prompt if you want to overwrite the old file or 'n' if you do not. In the latter case the program will request an alternative name.

Bad scale factors or baseline; they must be integers

This message occurs during automatic (command line) operation of Cfont. There is probably some error in the command line parameters. See section C.2 for a description of the required command line parameters.

Can't create <filename>; try another name.

If you try to create a file (e.g., in the *save* or *generate* commands) and that file cannot be created then this prompt will allow you to give another name. Most likely you are out of space on the disk or you typed an invalid file name. Give another file name, perhaps on a different drive (e.g., b:newfile.fon).

Can't find HERSHEY.CHR. Hershey file name>

Normally the file 'hershey.chr' should be on drive A or you should be logged to the disk containing the Hershey database when running Cfont. However, if this is not the case you will be prompted with this message for the location of the database file (e.g., 'm:hershey.chr' if you have the database file on an electronic disk).

Can't find the file: fancfont.pro

Cfont needs to read the file 'fancfont.pro' when it starts up in order to determine what type of printer is being used. Make sure 'fancfont.pro' is either on the currently logged drive or on drive A.

Can't open file <filename>; try another name>

If, when trying to read a file (e.g., in the *load* command), that file cannot be opened this message and prompt will allow you to try a different file name. You may have mistyped the file name. Try to type it again. Be sure to include the drive specifier (e.g., b:myfont.map) if the file is on a disk other than the currently logged drive.

Appendix 4 -- Cfont Messages

disk read error, end of file?

There was an error in trying to read from the temporary font file produced during the *generate* command. The disk containing the temporary font file must remain in the computer throughout execution of Cfont so make sure you haven't removed any disks after starting Cfont.

disk write error. Out of space?

You are probably out of space on the disk to which a file is being saved. If this occurs during a *save* command try to save to a different disk (e.g., b:myfont.fon). If this occurs during a *generate* operation you will need to save the current mappings if necessary, get out of Cfont, make space available or log to another disk, and then reenter Cfont and try again.

end of database file. Looking for continuation.

This should only arise on Osborne and other low-capacity disk systems. Cfont needs to find the file 'hershey1.chr' so respond to the prompt with the name of this file, including the drive specifier (e.g., c:hershey1.chr).

expecting an integer

This message is displayed when the program is reading from a file and is expecting to find an integer. There is probably a syntax error in the profile file, 'fancfont.pro'. If you have recently modified this file check it against the original and make sure all integers look right (e.g., make sure you don't have 'oh' for 'zero') and make sure the commas and spaces are still present in the file.

font set char no. must be between 0 and 127;

You tried to specify a mapping to an ASCII code greater than 127. Type another number, this time between 0 and 127.

have to skip over <integer> characters in the Hershey file so ...

This is just a warning that the next step of the *generate* or *print* command may take a long time.

Hershey character #<integer> is wider than its defining box.

This probably indicates that the Hershey database file, 'hershey.chr' has been corrupted. Try to copy it over again from the original disk.

Appendix 4 -- Cfont Messages

It any key to continue>

The screen freezes when it fills up with information. Type any key to display the next screenfull of information.

Invalid printer type in the file fancfont.pro

The printer type in the file 'fancfont.pro' is not one of the allowable types. Check to see that the string of characters between 'printer.type' and the following comma is one of the allowable printer types (see Appendix 5 for the list of allowable types).

Invalid density request: <integer>

This probably indicates a bad temporary font file. See the 'Disk read error' message.

No mappings deleted

The *zap* command was given a range of characters that did not include any of the currently mapped characters. If you really wanted to remove some of the mappings you probably mistyped one of the character range bounds and therefore nothing happened.

No mappings yet

There are no mappings from ASCII codes to Hershey character numbers currently in effect. Either none have been provided or all of them have been removed (with *zap*).

Not enough memory to plot characters

Your computer does not have enough memory to use the *generate* command. There is nothing to do but try to run it on a computer with more memory available to application programs.

Not enough memory to print at that scale factor

This is simply a warning that some of the characters may be clipped during the *print* command due to memory limitations. You may still be able to *generate* full-sized characters however.

Not enough room for the font set arrays

You do not have enough memory in your computer to even begin execution of Cfont. There is nothing to do but try to run it on a computer with more memory available to application programs.

Number is too long

This message is displayed when the program is reading from a file and is expecting to find an integer. If the message occurs immediately upon startup of the program then there is probably a syntax error in the profile file, 'fancfont.pro'. If you have recently modified this file

Appendix 4 -- Cfont Messages

check it against the original and make sure all integers are less than 20 digits long. If this message occurs during a *load* command the context of the message should identify where the problem is.

profile backspace string too long

The list of hex digits provided for 'backspace.output.string' in the profile file 'fancfont.pro' is too long; it must be four bytes or less. If you have modified this file recently check the syntax of the current file against the original and make sure the list of bytes provided for backspace string is length 4 or less.

ough draft mode error

This probably indicates a bad temporary font file. See the 'Disk read error' message.

me characters may be clipped

See the 'Not enough memory to print at that scale factor' message.

rry, no default here. Enter a number between <integer> and <integer>>

The previous prompt for a number does not allow for carriage return (or enter) alone as a response. That is, there is no default value for this prompt. You must type in a number explicitly.

font file is too large. The size is more than <integer> Kbytes

Font files currently have a maximum size of 64K bytes and you tried to *generate* a font larger than that. You must either reduce the scale factors and *generate* again or remove some of the characters from the mapping specification. One option is to cut a font into two parts by placing uppercase letters in one font and lowercase letters and punctuation in another font.

named file is not a valid mapping file. It looks ...

The *load* command is sometimes mistakenly used to load font files instead of mapping files into Cfont. See the description of the *load* command.

row=<integer>, bot row=<integer>, top blank=<integer>, bot blank=<integer>

See section E.2 for the meanings of these numbers for a font.

expected end-of-file

There is something wrong near the end of the file 'fancfont.pro'. If you have recently modified this file check it against the original and make sure all integers look right (e.g., make sure you don't have 'oh')

Appendix 4 -- Cfont Messages

for 'zero') and make sure the commas and spaces are still present in the file.

--- Notes ---

Appendix 5 -- Distribution File Descriptions

- prog.exe**> These are executable DOS programs. <prog> is one of Pfont, Efont, or Cfont.
- prog.com**> These are executable CP/M programs. <prog> is one of Pfont, Efont, or Cfont.
- overlay>.ovl** These are the CP/M overlay segments for either Pfont or Efont. Pfont overlays may reside on the CP/M logged in disk or on disk 'A', Efont overlays must reside on the CP/M logged in disk.
- font.hlp** This is a help file for Pfont. It should reside on the currently logged in disk or on drive 'A'. If not present, Pfont help will continue to work properly, except for the 'Control-V' general help.
- font.hle** This is a help file for Pfont following parameter input errors. It should reside on the currently logged in disk or on drive 'A'. If not present, '?' used following a '<E>' error prompt will provide no assistance.
- file.ff**> This is the conventional file name extension for *Fancy Font* (input) text files. The 'ff' extension is not required or specially recognized by any *Fancy Font* programs; it is strictly an informative convention.
- file.ffi**> This is the conventional file name extension for Pfont parameter files. For example, 'Pfont @letter.ffi myletter.ff' would be useful if 'letter.ffi' defined the standard letter format (page size, letterhead, etc.).
- hershey.chr** This is the Hershey character database used by Cfont. If it is on the logged in disk or on drive 'A', Cfont will find it automatically, otherwise Cfont will ask for the disk and file name where it can be found. The Osborne distribution package provides 'hershey.chr' in two halves (hershey.chr and hershey1.chr) due to disk capacity limitations.
- ancfont.pro** This is a configuration profile file, describing a few aspects of your computer system and printer. It is used by Pfont, Efont and Cfont to determine and adapt to the type of printer you are using, and by Pfont to adapt to the size of your display screen. The profile may be modified using your text editor (WordStar users, make sure to use *non-document* mode). See the end of this appendix for details.

Appendix 5 -- Distribution File Descriptions

<fontname.fon> These are the font files. The 'fon' extension is assumed by Pfont if no extension is given to the arguments to the '+FO' parameter. For example, '+FO Romn12' is equivalent to '+FO Romn12.fon'.
<fontname> adheres to the following conventions:

<style>{<face>}<size>{<supersub>}

where {<...>} indicates an optional element. <style> is an abbreviation for the style of the font, 'romn', 'sans', 'olde', and 'scrp' for Roman, Sans Serif, Old English, and Script respectively. <face> is optional and can be 'i' or 'b' for italic and bold. <size> is the size of the font in printers points (generally the distance from the bottom of the lower case descenders to the top of the upper case letters, measured in 1/72 inches). <supersub> is optional and is either 'p' for suPerscript or 'b' for suBscript.

The 'fancfont.pro' Profile

The file 'fancfont.pro' must be used in conjunction with all of the *Fancy Font* programs to identify the type of printer you are using. You may edit the contents of the profile, but be careful not to add any extraneous characters, blank lines or spaces.

The standard profile is configured for IBM computers and Epson '80' printers for DOS packages and your CP/M system and Epson '80' printers or CP/M packages. There are a few pre-configured profiles that you can rename 'fancfont.pro' rather than editing the existing profile. These are as follows:

fancfont.eps	all Epson printers
fancfont.gem	all Gemini printers, generic MSDOS computers
fancfont.dos	generic MSDOS and Epson printers
fancfont.ibm	IBM computers and Epson printers

To use one of these profiles, instead of the standard profile, on DOS systems, type:

```
copy fancfont.new fancfont.pro
```

On CP/M systems, type:

```
pip fancfont.pro=fancfont.new
```

where 'new' stands for one of the extensions listed above (e.g. 'gem').

Appendix 5 -- Distribution File Descriptions

The components of the profile are:

printer.type <type> {parallel} <type> is mx,fx,rx followed by
80 or 100 (e.g. mx80, fx100) or gemini10x or gemini15x for all
gemini printers; {parallel} is optional, it should be used with
IBM and compatibles to speed-up printing under DOS 2.0 and above.

number.of.columns.on.screen 80, normally 80, change number if desired

number.of.rows.on.screen 24, normally 24, change number if desired

backspace.key.input.code 8, normally 8, the decimal ASCII character code
for the backspace key on your keyboard

backspace.output.string 8 32 8; a list of decimal ASCII characters to be
sent to your display to delete the previous character

The Epson MX, RX and FX printers are all very similar; however specifying FX80 will change the operation of Pfont in two significant ways, 1) the default left margin (+LM) will be .75 inches and 2) reverse paper motion (+RV) will be enabled.

To be able to print the full width of wide printers, make sure to change your profile (e.g. mx100, fx100 or gemini15x) to indicate the type of printer, and also increase the line width (+LW) as desired.

--- Notes ---

Appendix 6 --- Hershey Character Database

400	D	E	F	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W
420	X	Y	Z	A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M	N	Ξ	O	Π	P
440	Σ	T	Υ	Φ	X	Ψ	Ω	A	B	C	D	E	F	G	H	I	J	K	L	M
460	N	O	P	Q	R	S	T	U	V	W	X	Y	Z	a	b	c	d	e	f	g
480	h	i	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	α
500	β	γ	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ
520	χ	ψ	ω	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o	p	q
540	r	s	t	u	v	w	x	y	z	ff	fi	fl	fm	fn	l	ε	θ	φ	ς	ff
560	fi	fl	ffi	ffl	ι				0	1	2	3	4	5	6	7	8	9	.	,
580	:	;	!	?	'	"	°	*	/	()	[]	{	}	<	>		-	
600	+	±	∓	x	·	÷	=	≠	≡	<	>	≤	≥	α	~	^	'	'	˘	'
620	'	'	'	c	u	∩	ε	→	↑	←	↓	∂	∇	√	∫	∫	∞	%	&	
640	@	\$	#	§	†	‡	∃	⊙	⊗	⊕	⊖	⊗	⊕	⊖	⊗	⊕	⊖	⊗	⊕	*
660	Ω	υ	Π	Σ	()	[]	{	}	()	√	∫	A	B	C	D	E	F
680	G	H	I	J	K	L	M	N	O	P	Q	R	S	T	U	V	W	X	Y	Z
700	A	B	Γ	Δ	E	Z	H	Θ	I	K	Λ	M	N	Ξ	O	Π	P	Σ	T	Υ
720	Φ	X	Ψ	Ω	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O	P
740	Q	R	S	T	U	V	W	X	Y	Z	x	a	b	c	d	e	f	g	h	i
760	j	k	l	m	n	o	p	q	r	s	t	u	v	w	x	y	z	α	β	γ
780	δ	ε	ζ	η	θ	ι	κ	λ	μ	ν	ξ	ο	π	ρ	σ	τ	υ	φ	χ	ψ

Appendix 6 -- Hershey Character Database

1200	а б в г д е ж з и й к л м н о п р с т у
1220	ф х ц ч ш щ ъ ы ь э ю я А В С D E F G H
1240	I J K L M N O P Q R S T U V W X Y Z A B
1260	C D E F G H I J K L M N O P Q R S T U V
1280	W X Y Z a b c d e f g h i j k l m n o p
1300	q r s t u v w x y z a b c d e f g h i j
1320	k l m n o p q r s t u v w x y z 0
1340	1 2 3 4 5 6 7 8 9 . , : ; ! ? ' ' & \$ /
1360	() * - + = ' " ° 0 1 2 3 4 5 6 7
1380	8 9 . , : ; ! ? ' ' & \$ / () * - + = ' "
1400	" ° ı Œ Ć Đ Ć Ğ Ğ Ğ Ğ Ğ Ğ Ğ Ğ Ğ Ğ Ğ Ğ Ğ Ğ Ğ Ğ Ğ
1420	Ğ Ğ
1440	m n o p q r ſ t u v w x y z A B C
1460	D E F G H I J K L M N O P Q R S T U V W X
1480	Y Z a b c d e f g h i j k l m n o p q
1500	r s t u v w x y z 0 1 2 3 4 5 6 7
1520	8 9 . , : ; ! ? ' ' & \$ / () * - + = ' "
1540	" ° H B Q D E F G H I J K L M N O P Q R
1560	S T U V W X Y Z a b c d e f g h i j k l
1580	m n o p q r s t u v w x y z

Appendix 7 -- Data File Formats

There are only two types of data files, the font files and the Hershey character database file.

FONT FILES

The font files have the following format.

<u>bytes</u>	<u>description</u>
0 to 29	Id string, 'Translation by SoftCraft, Inc.'. No trailing zero.
30	Font file version number.
31 to 32	Number of 'ink' data bytes in the file (see below). Low byte, high byte.
33 to n	Commentary text terminated with a zero. Variable length.
n+1	A constant relating to the X density of the font.
n+2	A constant relating to the Y density of the font.
n+3	Top blank points (see section E.2 and the Efont <i>auxinfo</i> command).
n+4	Bottom blank points.
n+5	Number of characters defined in the font, including spaces.
n+6 to n+15	Reserved but currently unused.
n+16 to n+24	These nine bytes define the height, width, and relative vertical and horizontal locations of the first character in the font. Immediately following these bytes are nine similar bytes for the next character, then nine for the next, and so on until all characters (byte n+5) are defined. The characters are arranged in increasing ASCII order. The nine bytes are defined as follows: 1 ASCII number. 2 to 3 Box width (left margin + right margin + ink width). A two byte integer (low byte, high byte). 4 to 5 Left margin (the starting location of the ink inside the box). A two byte integer (low byte, high byte). 6 to 7 Ink width. A two byte integer (low byte, high byte). 8 Ink height in rows (8 points, 24 dots per row for the Epson MX and FX version). 9 Row number (offset by 128) of ink bottom. The top row is this value plus the ink height. Section E.3 describes these parameters.

Appendix 7 -- Data File Formats

HERSHEY FILE

The Hershey character database has the following format. The first two bytes in the file are zero and are not used. Each character in the database is defined by a list of plotting directives and is best explained in terms of a pen plotter. The first two bytes in the character description are the width bytes and are not used by Cfont except for space characters. Each subsequent plotting directive is a two-byte entry that is either the x-coordinate of a new pen location or a special indicator. The special indicators are octal value 40000 specifying 'pen up' and 0 indicating the end of the character. Values other than these are coordinates to move the pen to, drawing a line if the pen was down, or just moving if the last value was 'pen up' (any pen movement implies that the pen is lowered at the end of the movement regardless of its state before the move). The coordinates are expressed as 'y_{data}' and 'x_{data}' bytes, in that order, and are plotted by Cfont at 'x_{coord}' and 'y_{coord}' according to the following equations.

$$x_{\text{coord}} = (x_{\text{data}} - 64 * x1/x2)$$
$$y_{\text{coord}} = (68 - y_{\text{data}} * y1/y2)$$

where x1, x2, y1, and y2 are the scale factors as described in the Cfont *scale* command. x_{coord} should be centered around 0 for the best possible allowable range of scaling factors, and the margins will be automatically set to 1 (they can be changed with Efont). y_{coord} equal to zero will be set at the baseline as specified by the Cfont *baseline* command (default value of 0). The above equations are modified slightly for character numbers greater than 88. The constant, 68, in the equation for y_{coord} is as follows: characters 0 through 88, 68; 89 through 396, 73; 397 through 672, 70; 673 through 1075, 73. The left and right margins are always set to one except as follows: characters 0 through 215, 1; 216 through 241, 0; 242 through 1075, 1; 1076 through 1101, 0; 1102 on, 1.

Appendix 8 -- ASCII Conversions

	<u>char</u>	<u>dec</u>	<u>oct</u>	<u>hex</u>		<u>char</u>	<u>dec</u>	<u>oct</u>	<u>hex</u>
null	␣	0	0	0		␣	64	100	40
	A	1	1	1		A	65	101	41
	B	2	2	2		B	66	102	42
	C	3	3	3		C	67	103	43
	D	4	4	4		D	68	104	44
	E	5	5	5		E	69	105	45
	F	6	6	6		F	70	106	46
bell	G	7	7	7		G	71	107	47
backspace	H	8	10	8		H	72	110	48
tab	I	9	11	9		I	73	111	49
linefeed	J	10	12	A		J	74	112	4A
	K	11	13	B		K	75	113	4B
formfeed	L	12	14	C		L	76	114	4C
return	M	13	15	D		M	77	115	4D
	N	14	16	E		N	78	116	4E
	O	15	17	F		O	79	117	4F
	P	16	20	10		P	80	120	50
	Q	17	21	11		Q	81	121	51
	R	18	22	12		R	82	122	52
	S	19	23	13		S	83	123	53
	T	20	24	14		T	84	124	54
	U	21	25	15		U	85	125	55
	V	22	26	16		V	86	126	56
	W	23	27	17		W	87	127	57
	X	24	30	18		X	88	130	58
	Y	25	31	19		Y	89	131	59
	Z	26	32	1A		Z	90	132	5A
escape		27	33	1B	left bracket		91	133	5B
	\	28	34	1C	backslash	\	92	134	5C
]	29	35	1D	right bracket]	93	135	5D
	^	30	36	1E	circumflex	^	94	136	5E
	_	31	37	1F	underscore	_	95	137	5F
space		32	40	20	backquote	`	96	140	60
exclamation	!	33	41	21	a	a	97	141	61
quote	"	34	42	22	b	b	98	142	62
hash	#	35	43	23	c	c	99	143	63
dollar	\$	36	44	24	d	d	100	144	64
percent	%	37	45	25	e	e	101	145	65
ampersand	&	38	46	26	f	f	102	146	66
apostrophe	'	39	47	27	g	g	103	147	67
left paren	(40	50	28	h	h	104	150	68
right paren)	41	51	29	i	i	105	151	69
asterisk	*	42	52	2A	j	j	106	152	6A
plus	+	43	53	2B	k	k	107	153	6B
comma	,	44	54	2C	l	l	108	154	6C
minus	-	45	55	2D	m	m	109	155	6D
period	.	46	56	2E	n	n	110	156	6E
slash	/	47	57	2F	o	o	111	157	6F
0	0	48	60	30	p	p	112	160	70
1	1	49	61	31	q	q	113	161	71
2	2	50	62	32	r	r	114	162	72
3	3	51	63	33	s	s	115	163	73
4	4	52	64	34	t	t	116	164	74
5	5	53	65	35	u	u	117	165	75
6	6	54	66	36	v	v	118	166	76
7	7	55	67	37	w	w	119	167	77
8	8	56	70	38	x	x	120	170	78
9	9	57	71	39	y	y	121	171	79
colon	:	58	72	3A	z	z	122	172	7A
semicolon	;	59	73	3B	left brace	{	123	173	7B
less than	<	60	74	3C	vertical rule		124	174	7C
equal	=	61	75	3D	right brace	}	125	175	7D
greater than	>	62	76	3E	tilde	~	126	176	7E
question	?	63	77	3F	delete	DEL	127	177	7F

Appendix 9 -- Parameter and Command Summary

Pfont Embedded Formatting Commands

<u>Font Selection Commands</u>	page	<u>Horizontal Motion Commands</u>	page
<digit> Select font number	17	\h±<hm> Move to horizontal position	
Select previous font		on current line	21
Select native printer fonts		m<digit> Move to marked position	
Select Fancy Font fonts		\i±<hm> Short horiz. relative move	23
<digit> Select horizontal rough		\m<digit> Mark current horizontal	
draft	15	position	29
Underline on/off	39	\t Move to next tab stop (^I)	37
Continuous underline on/off	47	\- Backup width of previous character	
		(^H)	46
<u>Format Control Commands</u>		<u>Substitution Commands</u>	
Temporary no justify/word wrap	12	\a<3 digit> Substitute ASCII char.	11
Center text	13	\s<digit> Substitute corresponding	
Justify on, 'smooth' right margin	24	+SU substitution string	35
Justify off, 'ragged' right margin	26	f Current file name	
Right align text	34	# Current page number	
Word wrap on, fill and wrap lines	42	<u>Vertical Motion Commands</u>	
Word wrap off	44	\v±<vm> Move vertically on page	41
<hm> Indent from left margin	49	m<digit> Move to vertical position	27
<hm> Indent from right margin	50	\l±<vm> Set interline spacing	27
<u>Miscellaneous Commands</u>		\n<digit> Mark current vert. position	30
Reverse background on/off	45	\p Start new page (^L)	32
Comment, ignore text	52	\q<vm> Conditional new page	33

optional + or -; <hm> is horizontal measure; <vm> is vertical measure, for example: 1.25i
 for <hm>- and <vm> are 'i' inches; 'c' centimeters; 'd' dots; 'u' user defined (see +UU)

<u>Efont Commands</u>	page	<u>Cfont Commands</u>	page
Load a font set	118	m Enter Hershey to font mappings	148
Save a font set	125	l Load Hershey to font mappings	
Print expanded form of chars.	121	from file	147
Edit a set of characters (create		g Generate font	145
'edf' file for each)	117	z Remove previously specified	
Replace a set of characters (from		mappings (zap)	153
'edf' files)	123	d Display current mappings, etc.	142
Display expanded character	115	p Print selected Hershey characters	150
Modify margins for set of chars.	119	f Set character scaling factors	143
Test font by printing chars.	126	b Set baseline (to move chars up	
Remove a set of characters (zap)	128	or down)	140
Move characters up or down (use		s Save current mappings to file	152
with 'e' and 'r')	127	q Quit	151
Display & modify font info.	113		
Quit			

Appendix 9 -- Parameter and Command Summary

Pfont Printing Parameters

<u>Font Selection Parameters</u> (^ signifies 'Control' char.)	default	page
FO Font file names, list of up to 10 (15 DOS) fonts	[off]	71
RD Rough draft selection, integer between 0 and 6 (roughest)	[off]	93
SD Screen display, display on screen instead of printing	[off]	97

Print Control Parameters

NF No header or footer line on first page	[off]	85
PP Pause between pages, to allow paper loading	[off]	91
PF Process form feeds (^L characters as \p)	[on]	86
FP First page to begin printing, integer between 1 and 9999	[off]	73
LP Last page to print, integer between 1 and 9999	[off]	81
NC Number of copies to print, integer between 1 and 9999	[1]	84

Page Setup Parameters

HL Header line, list of 1 or 2 strings (2 if odd/even headers)	[off]	75
FL Footer line, list of 1 or 2 strings (2 if odd/even footers)	[off]	68
PN Initial page number for \s#, integer between 0 and 9999	[1]	89

Horizontal Formatting Parameters (<hm> is 'horizontal measure'; e.g. 1.25i)

TB Set tab stops, list of up to 15 <hm>, 1 <hm> for even tabs	[.8i]	101
LW Line width from left margin, <hm> between 0 and 8 (13.6) inches	[6.5i]	82
LM Left margin, <hm> between 0 and 8 (13.6) inches	[1i]	79

Vertical Formatting Parameters (<vm> is 'vertical measure'; e.g. .75i)

HM Margin from top of page to header, <vm> between 0 and 425 inches	[.25i]	77
FM Margin from bottom of page to footer, <vm> between 0 and 425 inches	[.25i]	70
TM Margin from page top to text, <vm> between 0 and 425 inches	[.75i]	102
BM Margin from bottom of page to text, <vm> between 0 and 425 inches	[.75i]	63
PL Total page (paper) length, <vm> between 0 and 425 inches	[11i]	88
PG Enable pagination	[on]	87
RV Enable reverse paper motion (if printer supports, [on] for FX80)	[off]	95
SP Spacing between lines, <vm> between -425 and 425 inches	[.045i]	98

Miscellaneous Parameters

FI Names of files to be printed, list of up to 15 file names (tty:,con,loop:)	--	66
CF Concatenate files	[on]	64
SU Strings to substitute for \s, list of up to 10 strings	[off]	99
S Initialization string, string inserted at beginning of each file	[off]	78
JU User defined unit for 'u', list of 1 or 2 real numbers	[off]	103
CI Specify alternate command indicator, a single character	[\]	65

'During Printing' Commands

Z (^C for CP/M) Stop printing
N (^P) Immediately print next page
D Select horizontal rough draft
E Pause after current page
F Immediately print next input file

Special Characters

? Display quick help (try ^V)
@ Read parameters from file
> End a list or start printing
& Display current parameter settings
^U, ^H Line delete, Character delete