Small Planet Software

Sfware

A Manual for the Standalone Utilities

Version 1.0

NO WARRANTY

Every reasonable effort has been made to assure the quality and completeness of these programs. If you have any questions, comments or suggestions, or if you believe that you have found a bug, please contact the author at the address given at the end of this document.

1. Introduction

The Sfware softfont utilities from Small Planet Software are MS-DOS programs distributed under a shareware license agreement. These utilities provide extensive font manipulation capabilities for HP LaserJet softfonts.

The Sfware utilities allow you to download, rotate, compress, expand, view, and perform special effects on softfonts. The effects provided include bold, fill, convert to fixed spacing, halftone, hollow, invert, mirror, outline, convert to proportional spacing, resize, reverse, shade, shadow, slant, stripe, three-d, hollow-three-d, and filled-three-d. The effects can be tailored and customized for any font with various parameters and shading patterns. All of these features are provided by a set of standalone utilities. The utilities have a standardized interface to make learning how to use them as painless as possible. Additionally, the features are available through an integrated menu interface. The menu interface is provided by the SfShell program. SfShell is described in another manual—this manual documents the standalone utilities.

1.1. Getting Started

Sfware is “shareware,” and the author encourages archive maintainers to post Sfware for downloading; you may have received your copy from almost anywhere and almost anyone. Please make sure that you have a complete distribution before you try to install Sfware.

Hardware Requirements

In order to use Sfware, you will need a PC, PS/2, or close compatible with 150KB or so of free memory. A hard disk is recommended, but not required. A LaserJet+ or compatible printer is not actually *required*, but Sfware is probably a little pointless without one! Note: the original LaserJet printer does not have the ability to use softfonts; the Sfware utilities cannot help you use fonts with the original LaserJet printer.

If you want to use the SfShell program, you will need 300-400KB of *free* memory and either a hard disk, a ram disk, or sufficient expanded memory for swapping. You cannot use a removable (floppy) drive for swapping. The shell is not required for any of the features available in Sfware, but it does provide a user-interface for the Sfware utilities that is less intimidating than the command line.

If you are running a version of DOS prior to 3.30, read the configuration section carefully since you may have to do a little more work to get Sfware installed.

Software Requirements

In addition to Sfware, you must have at least one HP LaserJet softfont file.

1.3. Initial Configuration

Before you can use Sfware, you must run SfConfig to establish an initial configuration. Please follow the 'quick start' instructions in the **READ.ME** file or read the Configuration chapter before trying to run Sfware.

2.2. Semantic convensions

Parameter

Parameters are additional input, given on the command line, that you are required to enter. You cannot omit any parameters.

Option

Options are additional input, given on the command line, that you are not required to enter. You can omit any or all options.

filemask

A *filemask* is any valid DOS filename that may contain wildcard characters. The wildcard characters are interpreted exactly the way that DOS interprets them. Consult your DOS reference if you are unsure about what constitutes a valid filename or how you can use wildcard characters to select groups of files.

filename

A *filename* is any valid DOS filename. A *filename* must name a single specific file, it cannot contain wildcard characters.

fontmask

A *fontmask* is exactly like a *filemask* except that it must name one or more softfont files. The files whose names match a *fontmask* should contain LaserJet softfonts.

fontname

A *fontname* is exactly like a *filename* except that it must name a LaserJet softfont.

outname or *outmask*

An *outname* or *outmask* is any valid DOS filename or filemask. The file(s) specified for an *outname* or *outmask* will be created if they do not exist. In most cases, existing files with the same name will be replaced if they exist (the `replace` option determines whether or not confirmation is required before replacing an existing file).

When an outmask is used to decide what the name of each output file should be, the Sfware utilities perform the same filename resolution stratagy that DOS uses. First, any uses of '*' within the outmask are resolved to a string of '?'s. Then each character from the input filename is compared against the character at the same position in the outmask. If the outmask contains a '?' at that position, the character from the input filename is used for that position in the output filename. Otherwise the character from the outmask is used in the output filename.

`/replace`

If the `/replace` option is used, existing files will be replaced without warning. Not surprisingly, this option is only available on utilities that create files. If the `/replace` option *is not* used, then the utility may or may not replace files without warning depending on the setting of the "replace" option in the configuration file. Please consult the chapter on configuration files for more information about the `replace` configuration variable.

`/verbose`

All of the Sfware utilities print regular progress messages. The `/verbose` option causes many utilities to print more detailed progress messages.

`/quiet`

The `/quiet` option suppresses some informative messages. For example, the `/quiet` option will suppress the %-complete messages in SfLoad.

Technically Speaking

Many chapters end with a "technically speaking" section. This section describes, more technically, what Sfware does. You don't have to read it unless you want to. If you find the material in the technically speaking sections intimidating, just ignore it. On the other hand, if you find that something is not performing exactly the way that you thought it was supposed to, this section may help you figure out why Sfware is doing something other than what you expected.

The "real" technically speaking section of this chapter concerns the "switch character" used to introduce command-line options and parameters.

Because it was easy to implement,[†] Sfware can be told to accept either the forward slash ("/") or the hyphen ("-") as the switch character. This is controlled by the "`SwitchChar`" configuration parameter.

If you use the hyphen as the switch character, you *cannot* enter any filename that contains a hyphen because the hyphen will be interpreted as the start of another option. However, you can switch between switch characters on the command-line. If you enter `/-` when the foward slash is the switch character, the hyphen becomes the switch character and, conversely, if you enter `-/` when the hyphen is the switch character, the forward slash becomes the switch character.

[†] This may be a case of creeping featurism . . .

Laser Printer

The Laser printer selection helps Sfware decide if font compression should be enabled. Later versions of Sfware may make more use of this option.

Printer Output

The most common selection for printer output is `LPT1`. However, you can select LPT1-4, PRN, or any valid file or device for printer output.

Font Directory

Most users keep all of their softfonts in one directory. If this is the case, you can tell Sfware always to look for fonts in that directory regardless of what directory you are currently in. This becomes the default font directory. However, even if you do select a default font directory, you can still override it and use any directory you want be selecting a new directory with F4 in SfShell or by specifying a `fontdir` on the command line.

3.3. Format of the configuration file

The configuration file is a plain ASCII text file, and it should be edited with a program that will not insert extra formatting characters when the file is saved (I recommend Multi-Edit by American Cybernetics).

Each line of the configuration file is divided into three parts as follows:

```
programid parameter=value
```

The `programid` is seperated from the `parameter` by one or more spaces and the `parameter` is separated from the `value` by an equal sign (=). The `programid` is optional but the `parameter` and the `value` are required (actually, the `value` can technically be empty or blank but that is exactly the same as not defining it at all).

Individual Sfware programs use the combination of `programid` and `parameter` as a key to lookup the default value of each parameter. Any configuration line that does not contain a `programid` automatically matches all `programid`'s for that parameter. Case is insignificant in the `programid` and `parameter`.

A simple example should make everything clearer. Given the following configuration file:

```
PROG APPLE=1
PROG ORANGE=2
APPLE=3
OTHER ORANGE=4
```

The value of `PROG APPLE` is 1, `PROG ORANGE` is 2, `ANYTHING-ELSE APPLE` is 3, `OTHER ORANGE` is 4, and `ANYTHING-ELSE ORANGE` is undefined (blank, or non-existant).

Each possible configuration parameter is described in the sections that follow. The section header lists only the `parameter` if the `programid` is the name of the utility program that uses it. For configuration parameters that make special use of the `programid`, both parts are listed. The parameters are listed in alphabetical order by `parameter`.

3.9. GraphBack

Usage: *program* `GRAPHBACK=`*number*
Used by: SfShell, SfView

Controls the background color in graphics mode. The following colors can be used (they must be selected by *number*): 0=black, 1=blue, 2=green, 3=cyan, 4=red, 5=magenta, 6=brown, 7=light gray, 8=dark gray, 9=light blue, 10=light green, 11=light cyan, 12=light red, 13=light magenta, 14=yellow, and 15=white.

3.10. GraphCard

Usage: *program* `GRAPHCARD=`*cardname*
Used by: SfShell, SfView

Tells SfShell what kind of graphics card you are using. By default, SfShell tries to determine what kind of graphics card you have and adjust accordingly. However, if it makes the wrong choice, you can force SfShell to select one of the following: `CGA`, `MCGA`, `VGA`, `EGA`, `EGA64`, `EGAMONO`, `IBM8514`, `ATT`, `HERC`, and `PC3270`.

A complete list of available graphics resolutions for each card/mode is available under the section on “`GraphMode`”.

3.11. GraphForg

Usage: *program* `GRAPHFORG=`*number*
Used by: SfShell, SfView

Controls the foreground color in graphics mode.

3.12. GraphGrid

Usage: *program* `GRAPHGRID=`*number*
Used by: SfShell, SfView

Controls the color of the gridlines in the graphics display.

3.13. GraphMode

Usage: *program* `GRAPHMODE=`*number*
Used by: SfShell, SfView

Controls the graphics mode number for the selected graphics card. It is impossible for SfShell to know if you have selected a reasonable graphics mode. The results of using an incorrect or invalid graphics mode are undefined (and unpredictable!).

16), octal (base 8), decimal (base 10) and no reference numbers respectively. The default value is **hex**.

3.15. Pattern *name*

Usage: `PATTERN` *name=pattern-string*
Used by: SfShell, SfFx

The **pattern programid** introduces named patterns. Any pattern that you plan to use more than once or that is very complex should probably be saved in the configuration file. There is a whole chapter devoted to patterns and pattern strings. Please consult that chapter for more information about patterns.

The pattern created in the pattern chapter could be saved in the configuration file with the name **zig-zag** by placing the following line in the configuration file:

```
PATTERN ZIG-ZAG=0;34;85;136
```

3.16. Quiet

Usage: *program* `QUIET=YES` or `NO`
Used by: SfDir

Controls the degree of verboseness of messages from SfDir. In the future, other utilities may use this flag for the same purpose.

3.17. RefSet

Usage: *program* `REFSET=`*symbol-set*
Used by: SfShow

If the reference set is defined, the reference character for each position in the font will be printed in the upper right hand corner of each cell on SfShow's grid. For example, setting **sfshow refset=8u** would make SfShow print the reference characters with the 8U symbol set. You *must* select a symbol set that is available in your laser printer's line-printer font. If you don't want reference characters to be printed, use **sfshow refset=none**.

3.18. Replace

Usage: *program* `REPLACE=YES` or `NO`
Used by: SfCmpr, SfFx, SfLoad, SfRotate, SfShow

Each of the Sfware programs that can create new files use this flag to determine if existing files should be destroyed without warning. If you set this flag to `YES`, you can shoot yourself in the foot; if you leave it `NO`, the Sfware utilities will always make sure the safety is on!

The typefaces file is a plain text file. Each line should begin with a typeface number (typeface numbers 0 through 511 are valid as of PCL5; earlier printers only recognize typefaces numbered 0 through 255). The rest of the line is the typeface name. Lines that begin with a semicolon are ignored. The typeface numbers must be entered, one per line, in ascending order.

3.23. Sample File

This is a sample configuration file. This sample does not contain all of the possible configuration variables because many require defaults that are system-specific (graphics cards, program filenames, etc.) and many repetitious lines have been deleted.

```
Device      = LPT1
FontDir     = .
FontExtn    = SFP
Replace     = No
Compress    = No
;
SfShow Sentence=The quick red fox jumped over the lazy brown dog.
;
SfShell CommandFile=SFSHELL.CMD
SfShell SwapFile=SFSHELL.$$$
SfShell MsgFile=SFSHELL.MSG
SfShell ExecOutput=Window
;
SfShell FontListSize = 35
SfShell ActionListSize = 35
;
; The Pattern and Style lines are for SfFx
;
Pattern DarkSaw         = 255/127/62/28/8/128/193/227/247
Pattern LightSaw        = 128/65/34/20/8/0/0/0
Pattern NarrowBackslash = 136/68/34/17
Pattern TightSaw        = $6B/$DD/$B6/$6B/$DD/$B6
Pattern DecoSlash       = $D2/$69/$B4/$5A/$2D/$96/$4B/$A5
Pattern Cross           = $11/$BB/$EE/$BB
;
; Styles defined by HP:
;
; 0 = Upright
; 1 = Italic
; 2 = Oblique
;
Style Fill      =  5
Style Halftone  = 15
Style Hollow    =  3
Style Invert    = 14
Style Mirror    = 13
```

4. SFLOAD: Downloading Fonts

The SfLoad program downloads fonts to the LaserJet printer. Downloading fonts "teaches" the LaserJet printer how to print a particular font.

4.1. Usage

`SFLOAD` *fontmask* ⟦*options*⟧

4.2. Options

/image

Downloading a softfont as an image has two advantages: first, it is the fastest method and second, it should work for softfont formats that the Sfware utilities are not otherwise equipped to handle. For example, if a new laserjet printer, the Series IV perhaps, is developed with a new kind of softfont, downloading will continue to work with that new printer as long as you select the image option. The image option is the default.

If the image option is so great, why would I use anything else? Good question. There are two possible reasons. First, softfonts, especially large softfonts, take up a lot of disk space. The LaserJet IIP, III, and IIIP printers all support softfont compression (which provides substantial disk space savings for large softfonts). However, the LaserJet Series II printer does not support compression. If you have a LaserJet Series II printer and you always use Sfware to download your softfonts, you can still take advantage of the substantial space savings of softfont compression: compress all of your softfonts on disk and expand them when they are downloaded. You can't use the image option if you want to expand them when they are downloaded.

In a similar manner, softfonts can be rotated as they are downloaded if your laser printer does not support auto-rotation of fonts.

/expand

When the expand option is used, softfonts that are in PCL4 compressed format are expanded as they are being downloaded to the printer. This allows you to keep compressed softfonts on disk even if your printer does not support softfont compression.

/compress

When the compress option is used, softfonts are compressed using the PCL4 compression format as they are being downloaded to the printer. I can't think of a single good reason to use this option. It is provided only to satisfy the author's compulsive desire to provide the greatest possible flexibility.

5. SFFX: Special Effects

Because they are many and varied, each special effect is described in its own chapter (later in this document). All of the effects are produced by the SfFx program. All of the effects are created in the same general way; basically you enter:

`SFFX` *fontmask outmask effect-name effect-parameters effect-options*

The font masks select which font(s) will be used as the source and destination fonts. The *effect-name* indicates which effect to perform.

The SfFx program accepts the `/@`, `/!`, `/replace`, `/quiet`, `/verbose`, and `/$` options.

Every effect with the exception of the proportional and fixed spacing effects also accepts the `/range` parameter.

An online summary of the parameters and options of each effect is available by typing `SFFX HELP` *effect*. You can also type `SFFX HELP /RANGE` and `SFFX HELP /PATTERN`.

In the following chapters, where each effect is discussed, there are no examples of the effects because it was necessary to limit the number of fonts used in this document. This reflects a limitation in some LaserJet printers that prohibits printing more than sixteen different fonts on a given page.

A second document called `EXAMPLES.PCL` is included in the Sfware distribution. This file contains examples of some of the effects that are possible. The corresponding document `EXAMPL16.PCL` is provided for those printers that cannot print more than sixteen fonts per page.

Technically, every character within a softfont is defined within a rectangle. The rectangle is subdivided into squares like a sheet of graph paper. Inside the rectangle, some of the squares are black and some are white. Because the squares are very small, the effect of printing them on a sheet of paper is that they form the lines and curves that make up each charcter. In the descriptions of effects that follow, it is sometimes necessary to describe the way that "squares" within the rectangle are manipulated. The region of the grid that defines the character (the black dots on the "graph paper") is referred to as either the *black area* or the *foreground*, and the other "squares" are referred to as the *white area* or the *background.*

5.1. Ranges

Because the range option is available on almost every effect, it is described once here rather than repeating it for every effect.

The range option is available on all of the effects except proportional and fixed spacing. Specifying a range limits an effect to certain, specific characters. For example, you could limit the range of an effect to all of the uppercase letters.

6. SFCMPR: Compressing Fonts

Compression, available in the LaserJet IIP and subsequent printers, allows you to keep softfonts in a compressed format on disk. For fonts with large point sizes, this can achieve a *very* significant space savings.

6.1. Usage

`SFCMPR` *fontmask* ⟦*options*⟧

6.2. Options

outmask

If an *outmask* is not specified, the input fontname will be used (i.e. by default, SfCmpr replaces the input font with a compressed or expanded version of the same font).

`/expand`

The `/expand` option expands compressed fonts. If the input font is not compressed, this option has no effect. If neither `/expand` nor `/compress` are specified, `/compress` is assumed.

`/compress`

The `/compress` option compresses fonts. If the input font is already compressed, this option has no effect. If neither `/expand` nor `/compress` are specified, `/compress` is assumed. Only PCL4 compatible LaserJet printers can use fonts in compressed format.

Other Options

The SfCmpr program accepts the `/@`, `/!`, `/replace`, `/quiet`, `/verbose`, and `/$` options.

8. SFSHOW: Showing Fonts

Showing a font with SfShow creates a reference page that displays every character in the font. The reference page includes all of the font characteristics, the font selection sequence and a chart of all of the characters in the font.

What's to Show?

Every softfont can contain up to 256 different characters numbered from 0 to 255. Most fonts don't define all 256 different characters. The character chart is a grid that has "spaces" for each of the possible characters. If there are some character positions in the font that are not used, the spaces for those characters will be blank in the chart.

For small fonts, the character chart is a 16x16 grid on a single page. If the font is larger than about 36pt (or has some very tall or very wide characters), the characters may be too large to fit into the spaces in a 16x16 grid. In this case, multiple reference pages may be printed for the font. When multiple reference pages are required, SfShow attempts to use the minimum number of pages.

8.1. Usage

`SFSHOW` *fontmask* ⟦*options*⟧

8.2. Options

Downloading Options

In order to create a reference page, SfShow must first download the softfont. The following options control how each font is downloaded—they have precisely the same meaning as the SfLoad options with the same names: **`/image`**, **`/compress`**, **`/expand`**, **`/portrait`**, and **`/landscape`**.

`/id:`#

The **`/id`** option allows you to specify the starting font-id number that should be used for the temporary font downloaded in order to print the reference page. Since temporary fonts are deleted automatically whenever a printer reset is performed, it is not generally necessary to use this option.

`/gridoff`

The /gridoff option supresses grid lines on the reference page.

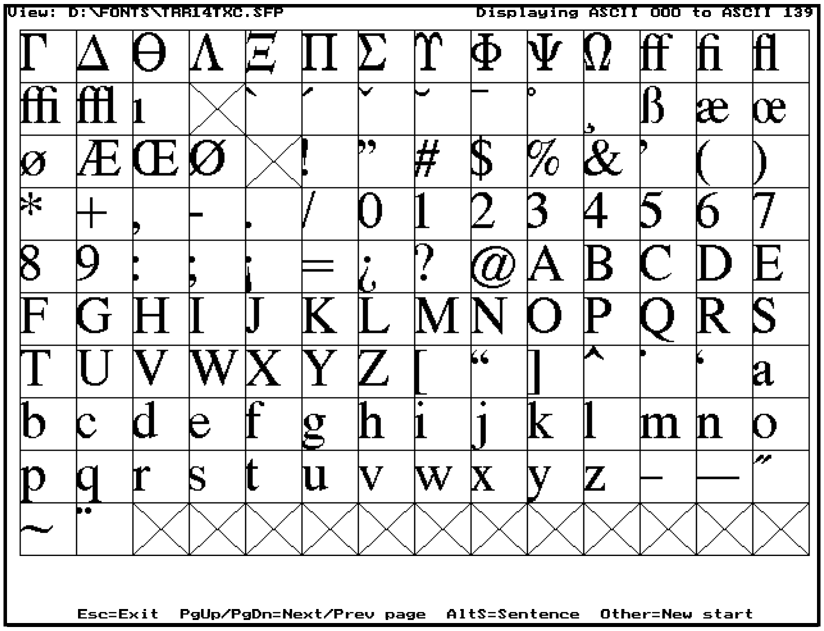
9. SFVIEW: Viewing Fonts

Viewing a font is the on-line equivalent of printing a reference page. SfView displays every character in the font on a grid similar to the printed output of SfShow. It can also display a sentence in the font. A graphics adapter is required to view fonts. The following adapters are supported at this time: CGA, MCGA, VGA, EGA, EGA (Mono), PC3270, IBM 8514, AT&T, and Hercules.

9.1. Usage

`SFVIEW` *fontfile* ⟦*options*⟧

Unlike the other Sfware utilities, SfView is interactive. When you run SfView, a something like the following will be displayed:



```
View: D:\FONTS\TRR14TXC.SFP                    Displaying ASCII 000 to ASCII 139
```

Γ	Δ	Θ	Λ	Ξ	Π	Σ	Υ	Φ	Ψ	Ω	ff	fi	fl
ffi	ffl	ı	☒	`	´	ˇ	˘	¯	˚	¸	ß	æ	œ
ø	Æ	Œ	Ø	☒	!	”	#	$	%	&	’	(	)
*	+	,	-	.	/	0	1	2	3	4	5	6	7
8	9	:	;	¡	=	¿	?	@	A	B	C	D	E
F	G	H	I	J	K	L	M	N	O	P	Q	R	S
T	U	V	W	X	Y	Z	[	“	]	^	˙	‘	a
b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	–	—	˝
~	¨	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒	☒

```
Esc=Exit  PgUp/PgDn=Next/Prev page  AltS=Sentence  Other=New start
```

Figure 9.1. SfView grid display

9.2. Options

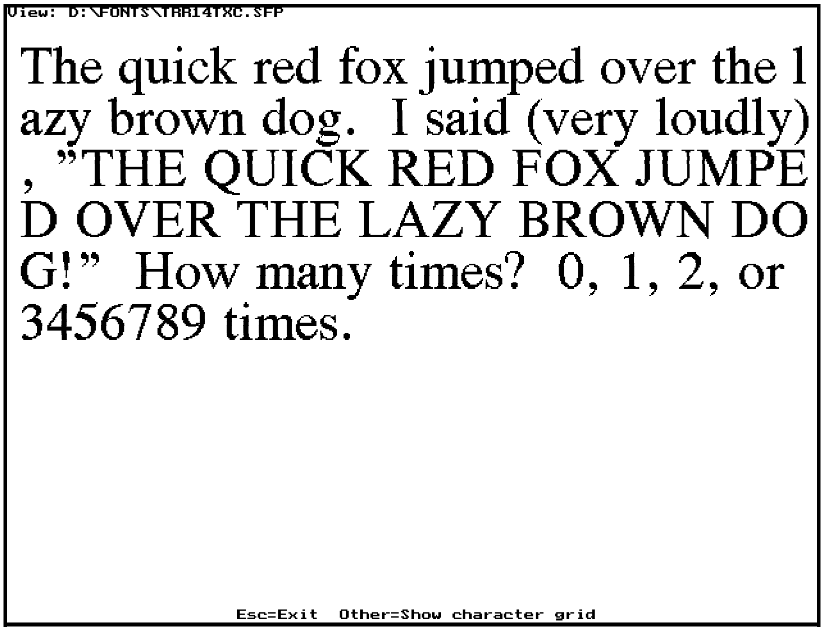
The SfView program accepts the /$ option.

9.3. Running SfView

The keys described below allow you to change the range of characters displayed, the quality of the display, and the format of the display.

Alt + S

Sometimes it is more useful to look at a font in the context of a sentence than it is to look at each individual character. This allows you to see how the characters interact with each other on the "printed page." The Alt+S key-combination alternates between the grid display and the sentence display. The sentence display looks like this:



```
View: D:\FONTS\TRR14TXC.SFP
```

The quick red fox jumped over the l
azy brown dog. I said (very loudly)
, "THE QUICK RED FOX JUMPE
D OVER THE LAZY BROWN DO
G!" How many times? 0, 1, 2, or
3456789 times.

```
Esc=Exit  Other=Show character grid
```

Figure 9.2. SfView sentence display

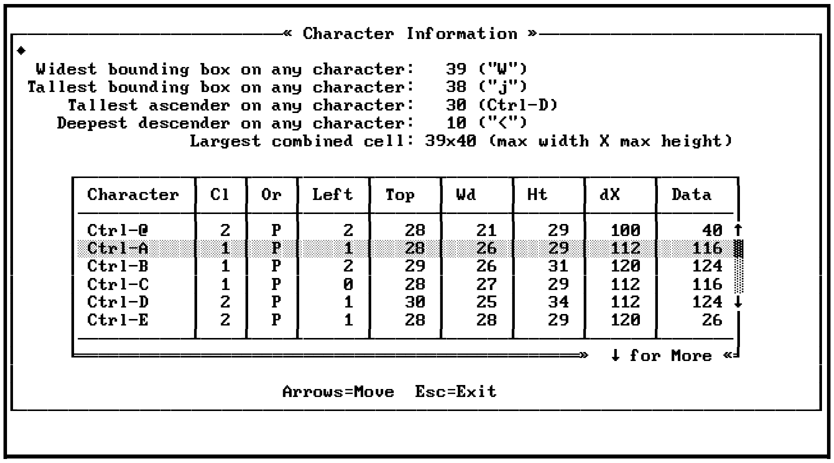
Other

Pressing any other key changes the range of characters displayed to begin with the key you pressed.

F4

Pressing F4 displays character information for the font. The dimensions of the largest character in the font are summarized and a scrolling list of the characters in the font is displayed.

The additional info panel looks like this:



```
                          « Character Information »
◆
  Widest bounding box on any character:   39 ("W")
 Tallest bounding box on any character:   38 ("j")
     Tallest ascender on any character:   30 (Ctrl-D)
    Deepest descender on any character:   10 ("<")
                  Largest combined cell: 39x40 (max width X max height)

      Character   Cl   Or   Left   Top   Wd    Ht    dX    Data

      Ctrl-@       2    P      2    28    21    29   100     40 ↑
      Ctrl-A       1    P      1    28    26    29   112    116
      Ctrl-B       1    P      2    29    26    31   120    124
      Ctrl-C       1    P      0    28    27    29   112    116
      Ctrl-D       2    P      1    30    25    34   112    124 ↓
      Ctrl-E       2    P      1    28    28    29   120     26

                                                 » ↓ for More «

                        Arrows=Move  Esc=Exit
```

Figure 10.2. Additional Character Information Panel

The scrolling list of characters displays the class, orientation, left-offset, top-offset, width, height, delta-X, and data sizes of every character in the font. These are technical measurements in the softfont and can be ignored by most users.

The left-offset, top-offset, width, and height fields are PCL coordinate system dots. The delta-X field is in 1/4 dot units. The data size is in bytes. For compressed fonts (class 2 characters), this is the data size of the compressed character, *not* the expanded character.

F5

Pressing F5 displays any additional information present in the font header. The most common use of this area is font copyright information. The special effects program in Sfware uses this area to describe what effects have been performed on the font.

Not all fonts have additional information in the header.

F6

When the font is scanned, it is frequently possible to recognize that it is not "valid" for some printers. The LaserJet III printer (and, presumably, printers that follow it) have a very relaxed set of guidelines as to what constitutes a valid font. Older printers, the LaserJet Series II in particular, have very stringent requirements. SfInfo recognizes these incompatabilities and will display a warning message for each problem that it finds. If the problem can easily be corrected, the appropriate action is described.

12. The Bold Effect

Emboldening a font makes it appear darker on the page. Adding a large amount of boldness to a font will cause it to blur and become difficult to read. In professional typography, the characters in a bold version of a font have different shapes and proportions. This is beyond the ability of Sfware. A normal font made bolder with Sfware will not look the same (and probably will not look as good) as a real bold version of the original font.

12.1. Usage

`SFFX` *fontmask outmask* `BOLD` ⟦*options*⟧

12.2. Options

/bold

The `/bold` parameter controls how much darker the characters are made to appear. The larger the number, the darker the characters will be. For best results, the amount of boldness should be small with respect to the size of the font. It's difficult to define "small" in this context; one-tenth of the point size of the font (or less) is probably a good estimate. Experiment and see what looks most pleasing to the eye.

/range

The range option limits the scope of the effect. Please refer to the SfFx chapter for more information.

12.3. Technically Speaking

The bold effect locates "edge" pixels (that is, pixels that are on a border of the character) by scanning horizontally across each row of pixels. Every time a pixel position is found that is currently off and adjacent to, but not surrounded by, pixels that are on, the pixel is turned on. This has the effect of adding pixels to the border of the character. The appropriate font and character parameters are updated so that the original character shape (now surrounded by a border) prints in the same position as the original. In other words, the left offset is incremented by one, the baseline is incremented by one, and the character bounding box is expanded. If an amount of boldness greater than one is specified, the above algorithm is iterated to produce the correct amount of boldness.

14. The Fill Effect

Filling a font creates outlined characters filled with a user-specified pattern.

14.1. Usage

`SFFX` *fontmask outmask* `FILL /pattern:`*pattern* ⟦*options*⟧

14.2. Parameters

/pattern

Patterns can be specified directly or by using names defined in the configuration file. The *Patterns* chapter describes how to create patterns; the *Configuration* chapter describes how to save and name patterns.

14.3. Options

/range

The range option limits the scope of the effect. Please refer to the SfFx chapter for more information.

14.4. Technically Speaking

Patterns are described in more technical detail in the *Patterns* chapter.

15.3. Options

/blacktop

The **/blacktop** pattern replaces the black areas of the non-selected region. Please refer the the *Patterns* chapter elsewhere in this manual. The **/blacktop** option can be abbreviated to **/btop**.

/whitetop

The **/whitetop** pattern replaces the white area (everything in the cell that isn't black) of the non-selected region. The **/whitetop** parameter can be abbreviated to **/wtop**.

/range

The range option limits the scope of the effect. Please refer to the SfFx chapter for more information.

15.4. Technically Speaking

This effect forms the heart of several effects in SfFx. For example, the "shade" effect is nothing more than the halftone effect applied to a selected region from 0% to 100% of the character! If you understand the concept of a pattern (discussed in the *Patterns* chapter), it shouldn't be too difficult to understand the halftone effect.

Note: in any effect that changes the background pattern, it may be necessary to turn off "kerning" within the word processor or other program that you use to print the font. Normally, causing two characters to overlap by a small amount (for example a capital "T" followed by a lowercase "o") is not noticable because they only overlap in the "white" background. However, after you have changed the background to a pattern other than plain white, the effect of overlapping two characters by even a small amount may be undesirable.

17. The Hollow Effect

Hollowing a font produces an unfilled outline of each character.

17.1. Usage

`SFFX` *fontmask outmask* `HOLLOW` ⟦*options*⟧

17.2. Options

/range

The range option limits the scope of the effect. Please refer to the SfFx chapter for more information.

17.3. Technically Speaking

The hollow effect and the outline effect are very closely related. The only difference is the placement of the outline. In the hollow effect, the existing perimeter of each character is left in place and the interior is "scooped out". For the outline effect, the entire character is erased and a new perimeter is added just around the character. In effect, an outlined character is a hollowed bold character (see the technically speaking section of the bold effect for more details).

It should also be noted that the hollow and fill effects are closely related. A hollowed character is a filled character with a pattern of 0.

19. The Mirror Effect

Mirroring a font creates characters that are upside down.

19.1. Usage

`SFFX` *fontmask outmask* `MIRROR` ⟦*options*⟧

19.2. Options

/adj

The `/adj` parameter changes the relative position of the (virtual) mirror across which each character is rotated. A value of zero specifies that the mirror is on the baseline, values larger than zero move the mirror above the baseline, smaller values move it below.

/range

The range option limits the scope of the effect. Please refer to the SfFx chapter for more information.

19.3. Technically Speaking

If you plan to use a font and its mirror to create a special display effect (by placing one above the other), you may find that the descenders on the original font overlap the "descenders" (now ascenders!) on the mirrored font. This is where it is helpful to change the mirror baseline adjustment. By making the adjustment roughly equal to the number of pixels in the descenders of the original font, you can move the mirrored font "down" a little so that the mirrored descenders don't overlap the descenders on the original font.

21. The Outline Effect

Outlining a font produces an unfilled outline of each character.

21.1. Usage

`SFFX` *fontmask outmask* `OUTLINE` ⟦*options*⟧

21.2. Options

/range

The range option limits the scope of the effect. Please refer to the SfFx chapter for more information.

21.3. Technically Speaking

See the technical discussion of the hollow effect for more information.

23. The Resize Effect

Resizing a font creates characters that are larger or smaller than the same characters in the original font. The characters can be scaled uniformly (creating more or less accurate renditions of the original characters with the same proportions) or non-uniformly (creating elongated or widened characters).

23.1. Usage

`SFFX` *fontmask outmask* `RESIZE /width:`*%* `/height:`*%* ⟦*options*⟧

23.2. Parameters

/width

The `/width` parameter specifies the width of each resized character as a percentage of its original size. Values less than 100 make the characters narrower, while values larger than 100 make them wider.

/height

Like the width, the new height specifies the height of each resized character as a percentage of its original height.

23.3. Options

/range

The range option limits the scope of the effect. Please refer to the SfFx chapter for more information.

23.4. Technically Speaking

In practice, this effect has few uses. Unlike more modern font scaling technology (which relies on mathematical descriptions of each character) SfFx has only the bitmap description of each character to work with. As a result, gross changes in the size of a character create "jagged" edges and *very* poor quality letters. Making fonts larger generally works better than making them smaller. As a rule of thumb, you probably won't like the results if you try to resize a font by more than a factor of two. If you hold one dimension constant (100%), it may be possible to stretch or compress the other dimension by a larger factor without significant loss of detail.

25. The Shadow Effect

Shadowing attempts to produce the effect that you would get if all you could see on the page were the shadows from an embossed image of the original character. It's a bit difficult to describe, but it is one of my favorite effects.

25.1. Usage

`SFFX` *fontmask outmask* `SHADOW /x:`# `/x:`# ⟦*options*⟧

25.2. Parameters

/X

The **/x** parameter controls the width (in dots) of the shadow to the right or left of the original character. Positive values create a shadow on the right hand side of the character, while negative values create a shadow on the left. This value should be small relative to the total width of the character.

/Y

By analogy with the **/X** parameter, the **/y** parameter controls the height of the shadow above or below the character. Positive values create shadows below the character, negative values above. This value should be small relative to the total height of the character.

25.3. Options

/range

The range option limits the scope of the effect. Please refer to the SfFx chapter for more information.

25.4. Technically Speaking

The effect is produced by moving a copy of the character over and down by the specified amounts and then removing all dots that fall within the original character (including all of the original character). For small widths, this works fine; however when the widths become larger than the widths of the strokes that make up the chacter, the effect falls apart.

27. The Slant Effect

Slanting is a poor-man's version of italics. In practice, italic fonts are not *just* slanted versions of the upright characters. But slanting will suffice in a pinch and it does allow you to produce oblique characters (slanted backwards), which are occasionally useful.

27.1. Usage

`SFFX` *fontmask outmask* `SLANT /deg:`# ⟦*options*⟧

27.2. Parameters

/deg

The `/deg` parameter specifies the amount of slant in degrees. A positive value causes the characters to slant toward the right. A negative value causes the characters to slant toward the left.

27.3. Options

/range

The range option limits the scope of the effect. Please refer to the SfFx chapter for more information.

27.4. Technically Speaking

This effect is produced by calculating how far over each row of pixels must be moved in order to produce a slant of the requested angle. Using a little bit of trigonometry, it is easy to calculate how far over the top row must be moved. Each row below the top must be moved over some fraction of the total height of the character. Rows below the baseline must be moved in the opposite direction.

Considering that this algorithm does nothing more than slide rows of dots back and forth, it should be easy to see that large slant values may produce jagged, non-contiguous characters.

29. The Three-D Drop Shadow Effect

Three-D drop shadows create a patterned shadow-image of each character that appears to be below the original. It is possible to change the apparent “distance” of the shadow by changing the offsets used to create it.

29.1. Usage

`SFFX` *fontmask outmask* `THREED /X:`# `/Y:`# `/pattern:`*pattern* ⟦*options*⟧

29.2. Parameters

/X

The `/x` parameter controls the distance (in dots) of the shadow to the right or left of the original character. Positive values create a shadow on the right hand side of the character, negative values create a shadow on the left.

/Y

By analogy with the `/X` parameter, the `/y` parameter controls the distance of the shadow above or below the character. Positive values create shadows below the character, negative values above.

Pattern

The pattern specified is applied to the areas used in the shadow.

29.3. Options

/range

The range option limits the scope of the effect. Please refer to the SfFx chapter for more information.

29.4. Technically Speaking

The original character is moved left or right and up or down by the distances specified. If necessary the character cell is enlarged to accommodate the new character. The character is then shaded with the specified pattern and the original character is painted back into the character cell at its original position.

31. The Filled Three-D Drop Shadow Effect

Filled Three-D drop shadows are simply a combination of the three-d drop shadow effect and the fill effect. It is a limitation of the algorithms used to create the three-d drop shadow effect that it is not possible to create a filled three-d character. This effect is provided to circumvent that limitation.

31.1. Usage

`SFFX` *fontmask outmask* `FILL3D /X:`*#* `/Y:`*#* `/pattern:`*pattern* ⟦*options*⟧

31.2. Parameters

/X

The `/x` parameter controls the distance (in dots) of the shadow to the right or left of the original character. Positive values create a shadow on the right hand side of the character, negative values create a shadow on the left.

/Y

By analogy with the `/X` parameter, the `/y` parameter controls the distance of the shadow above or below the character. Positive values create shadows below the character, negative values above.

/pattern

The pattern specified is applied to the original character.

31.3. Options

/range

The range option limits the scope of the effect. Please refer to the SfFx chapter for more information.

31.4. Technically Speaking

This option is exactly the same as the three-d drop shadow effect except that the shadow is always solid black and instead of painting the original character back into the cell, a pattern-filled version of the original character is painted back in.

33. Patterns

33.1. What are patterns?

Patterns change the appearance of many effects. A pattern is a rectangular region of on-and-off dots that is repeated across and down to cover the region being filled. The pattern is specified as a series of numbers separated by commas and semicolons. The binary representation of the numbers separated by commas indicates the dots that are on and off in each row and semicolons separate rows.

33.2. How do I create one?

Creating a new pattern is not difficult. The best way to begin is with a piece of graph paper and a pencil. Experiment until you have something that you like and then follow the directions below.

For example, suppose that you wish to create a zig-zag pattern. Here is one possibility:

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

Isolate a "generating region"

Isolate the smallest region that can be used to generate the pattern. This region, when repeated to the right and down, should create the entire pattern. In this case, the smallest acceptable region is this:

		*	
	*		*
*			

Note: there is frequently more than one smallest region that will produce the pattern. I have intentionally chosen this region because it is *not* the upper-left hand corner. Usually the upper-left hand corner contains a generating region, but not always.

33.3. Saving the pattern

Alternatively, the patterns may be saved in the configuration file and selected by name. Read the *Configuration* chapter for more information.

33.4. Previewing Patterns

It is possible to preview any pattern by running SfFx with the following options:

```
SFFX /PREVIEW pattern
```

One “screenfull” of the pattern will be displayed.

33.5. Technically Speaking

The fact that patterns are used for so many effects makes it apparent that Sfware really needs a pattern editor and a better mechanism for storing patterns. These are planned additions but Sfware is already beginning to suffer from “creeping featurism” (in the author’s opinion, at least) and it has been decided that these changes will just have to wait until the next release.

However, in view of the fact that creating patterns by the above method is very tedious, a simple program (`PATTERN.EXE`) has been added to Sfware that eliminates most of the “hard parts.” Please consult the file `PATTERN.DOC` for more information.

35. Contacting the Author

35.1. By Mail

You can reach the author by mail at the following address:

Norman Walsh
#421 Southwood Apts
Brittany Manor Dr
Amherst, MA 01002

35.2. Electronically

If you have access to electronic mail, the fastest way to reach the author is to send electronic mail to **walsh@cs.umass.edu**. In this case, electronic mail implies access to Internet domains (through BITNET or UUCP, for example). This is possible from CompuServe and from several of the large national BBS's as well.

filemask A filemask is a DOS filename which may include the "wildcard" characters `*` and `?`. The wildcard characters in a filemask allow you to select a group of files. Please consult your DOS reference for more information about wildcard characters.

font A font is a collection of symbols that have similar characteristics. The symbols in a font have a fixed typeface, size, weight, style and symbol set. For example, upright, bold Times Roman at 10pt is a font. Contrast with typeface.

fontdir In the context of this manual, a fontdir is the filemask (optionally including a path) that identifies LaserJet softfont files. For example, if you keel all of your softfonts in the directory `d:\fonts` then `d:\fonts\A*.SFP` is one example of a valid fontdir. The canonical font directory would be `d:\fonts\*.*`.

hexadecimal Hexadecimal refers to the number base composed of sixteen symbols (0-9,A-F). Hexadecimal is frequently used in computing because 256 different values can be represented in only two digits. Hexadecimal is sometimes called base 16.

kerning Kerning refers to slight changes in the spacing between characters. Some letter combinations ("AV" and "To", for example) appear farther apart than others because of the shapes of the individual letters. Many sophisticated word processors move these letter combinations closer together automatically (compare "AV" with "AV", for example).

laserjet Laserjet is a trademarked name for laser printers used by Hewlett Packard. In this document, it simply means an HP LaserJet printer or a compatible laser printer from some other manufacturer.

mask See *filemask*.

memory, expanded See *EMS*.

memory, extended Extended memory is memory above the 1 megabyte boundry in your computer. Sfware cannot directly use extended memory. Many programs exist which map extended memory as expanded memory. For more information about expanded memory, see *EMS*.

memory, main Main memory is the DOS memory below 640K in your computer. This is the area where normal DOS programs run. The DOS **chkdsk** program, for example, reports the amount of main memory that is free.

octal Octal refers to the number base composed of eight symbols (0-7). Octal is sometimes called base 8.

pathname A pathname is a filename (please consult your DOS reference for more information about what constitutes a valid DOS filename) with its associated drive and path. For example, if `tr100.sfp` is the name of a file in the directory `\fonts` on drive `d:`, then

```
d:\fonts\tr100.sfp
```

is the pathname of `tr100.sfp`.

Table of Contents

1. Introduction 1
2. Conventions Used in This Manual 4
3. Configuring Sfware and Using Command Files 8
4. SFLOAD: Downloading Fonts 17
5. SFFX: Special Effects 19
6. SFCMPR: Compressing Fonts 21
7. SFROTATE: Rotating Fonts 22
8. SFSHOW: Showing Fonts 23
9. SFVIEW: Viewing Fonts 25
10. SFINFO: Displaying Softfont Information 28
11. SFDIR: Softfont Directory Listings 30
12. The Bold Effect 31
13. The Fixed Spacing Effect 32
14. The Fill Effect 33
15. The Halftone Effect 34
16. The Horizontal Fade/Mist Effect 36
17. The Hollow Effect 37
18. The Invert Effect 38
19. The Mirror Effect 39
20. The Mist Effect 40
21. The Outline Effect 41
22. The Proportional Spacing Effect 42
23. The Resize Effect 43
24. The Reverse Effect 44
25. The Shadow Effect 45
26. The Shade Effect 46
27. The Slant Effect 47
28. The Stripe Effect 48
29. The Three-D Drop Shadow Effect 49
30. The Hollow Three-D Drop Shadow Effect 50
31. The Filled Three-D Drop Shadow Effect 51
32. The Vertical Fade/Mist Effect 52
33. Patterns 53
34. Sfware Registration 56
35. Contacting the Author 57
36. Glossary 58

Sfware Order Form

Name: ____________________ Phone: (____) ____–______

Company: ____________________

Mailing Address: ____________________

____

City: ____________________ State: ______ Zip: ____________

Please pay by check or money order, do not send cash through the mail. Make all checks payable to Norman Walsh.

Individual utilities:	Quantity	Price Each	Total
______ SfFx (softfont special effects)	______	$25	______
______ SfCmpr (softfont compression)	______	$10	______
______ SfLoad (download fonts)	______	$10	______
______ SfRotate (landscape/portrait conversion)	______	$10	______
______ SfShow (print summary page)	______	$10	______
______ SfView (preview font on screen)	______	$10	______
______ SfDir (directory enhancement for fonts)	______	$ 5	______
______ SfInfo (complete font information)	______	$ 5	______
Utility bundles:			
______ SfShell (menu interface shell) and *all* utilities	______	$60	______
______ All of the utilities (without SfShell)	______	$40	______
______ Any three utilities (excluding SfFx, select above)	______	$20	______

Subtotal: ______

Massachusetts residents, please add appropriate sales tax: ______

Total: ______

Complete this form and return it with your payment to:

Norman Walsh
#421 Southwood Apts
Amherst, MA 01002