



# Sftware

A Manual for the Standalone Utilities

Copyright © 1990-92 by Norman Walsh  
All Rights Reserved

Version 1.1

## NO WARRANTY

THESE PROGRAMS ARE DISTRIBUTED FREELY. THESE PROGRAMS IS PROVIDED "AS IS" WITHOUT WARRANTY OF ANY KIND, EITHER EXPRESSED OR IMPLIED, INCLUDING BUT NOT LIMITED TO, THE IMPLIED WARRANTIES OF MERCHANTABILITY AND FITNESS FOR A PARTICULAR PURPOSE. THE ENTIRE RISK AS TO THE QUALITY AND PERFORMANCE OF THE PROGRAMS IS WITH YOU. SHOULD THE PROGRAMS PROVE DEFECTIVE, YOU ASSUME THE COST OF ALL NECESSARY SERVICING, REPAIR OR CORRECTION.

IN NO EVENT WILL ANY COPYRIGHT HOLDER BE LIABLE TO YOU FOR DAMAGES, INCLUDING ANY GENERAL, SPECIAL, INCIDENTAL OR CONSEQUENTIAL DAMAGES ARISING OUT OF THE USE OR INABILITY TO USE THE PROGRAMS (INCLUDING BUT NOT LIMITED TO LOSS OF DATA OR DATA BEING RENDERED INACCURATE OR LOSSES SUSTAINED BY YOU OR THIRD PARTIES OR A FAILURE OF THE PROGRAMS TO OPERATE WITH ANY OTHER PROGRAMS), EVEN IF SUCH HOLDER OR OTHER PARTY HAS BEEN ADVISED OF THE POSSIBILITY OF SUCH DAMAGES.

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 Sftware 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 Sftware 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

Sftware is “shareware,” and the author encourages archive maintainers to post Sftware 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 Sftware.

---

## Hardware Requirements

In order to use Sftware, 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 Sftware is probably a little pointless without one! Note: the original LaserJet printer does not have the ability to use softfonts; the Sftware 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 Sftware, but it does provide a user-interface for the Sftware 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 Sftware installed.

---

## Software Requirements

In addition to Sftware, 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 conversions

---

### 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 Sftware utilities perform the same filename resolution strategy 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,<sup>†</sup> 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 forward 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.

---

<sup>†</sup> This may be a case of creeping featurism . . .

---

**Laser Printer**

The Laser printer selection helps Sftware decide if font compression should be enabled. Later versions of Sftware 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 Sftware 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 by 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 separated 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 Sftware 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-existent).

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 verbosity 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 Sftware 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 Sftware 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.COMD
SfShell SwapFile=SFSHELL. $$$
SfShell MsgFile=SFSHELL.MSG
SfShell ExecOutput=Window
;
SfShell FontListSize = 35
SfShell ActionListSize = 35
;
; The Pattern and Style lines are for Sfx
;
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 SflLoad 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 Sftware 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 Sftware 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 Sftware 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 character. 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 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. The default value for this parameter is 16384.

---

#### */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 Sftware utilities, SfView is interactive. When you run SfView, a something like the following will be displayed:

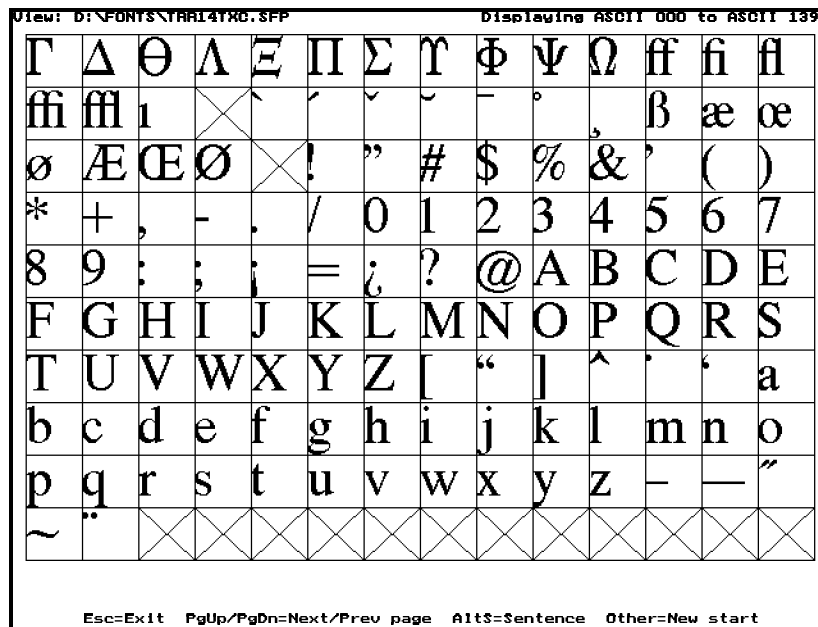


Figure 9.1. SfView grid display

---

### 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:

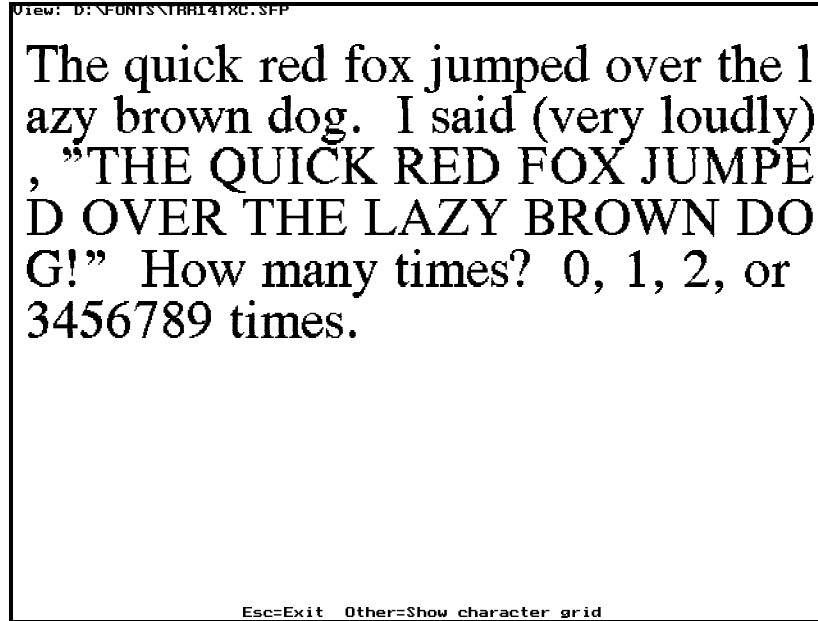


Figure 9.2. SfView sentence display

---

**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 ("U")  
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	2*	21	29	100	40 ↑
Ctrl-A	1	P	1	20	26	29	112	116
Ctrl-B	1	P	2	29	26	31	120	124
Ctrl-C	1	P	0	20	27	29	112	116
Ctrl-D	2	P	1	30	25	34	112	124 ↓
Ctrl-E	2	P	1	20	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 Sftware 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 incompatibilities 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`.

---

`/-`

The `/-` option forces a “space” character to exist. It may be particularly useful with this effect. For more information, consult the technically speaking section of the Sfx chapter.

---

`/range`

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

---

### 15.4. Technically Speaking

This effect forms the heart of several effects in Sfx. 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 noticeable 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 offsets, this works fine; however when the offsets become larger than the widths of the strokes that make up the character, 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* THREEED /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 Hollow Three-D Drop Shadow Effect

---

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

---

### 31.1. Usage

`SFFX fontmask outmask HOLLOW3D /X:# /Y:# [[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.

---

### 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 hollowed version of the original character is painted back in.

---

## 33. The Vertical Fade/Mist Effect

---

Fading a font with this effect “smudges” out the top or bottom edge of each character.

---

### 33.1. Usage

SFFX *fontmask outmask* VFADE /to:% *[[options]]*

---

### 33.2. Parameters

---

/to

The /to parameter determines what percentage of the character is faded out. A fade factor of 100% applies the fade all the way down each character so that 100% black saturation is achieved in the last row of pixels. Fade factors below 100% apply the fade more rapidly so that a 100% black saturation is achieved before the bottom of the character. Conversely, fade factors above 100% draw the fade out so that it never reaches saturation.

---

### 33.3. Options

---

/back

By default, a vertical fade begins with 0% black on the top row of the character and proceeds towards 100% on the bottom row (at a rate determined by “fade percent.” See above). If backwards fading is selected, the fade begins with 0% black on the bottom row of the character and proceeds towards 100% on the top row.

---

/range

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

---

### 33.4. Technically Speaking

See the technically speaking section for the Horizontal Fade/Mist effect. The vertical fade algorithm is a natural analog of the horizontal fade algorithm.

---

**Round to 8-dots**

The region used to generate the pattern must be an even multiple of eight dots wide. Repeat the smallest region to the right until it is a multiple of eight dots wide. You must repeat the entire pattern (for example, if the region is 6 dots wide, you will have to repeat it until it is 24 dots wide). In this case the smallest region a multiple of eight dots wide is this:

		*				*	
	*		*		*		*
*				*			

---

**Use zeros and ones**

Redraw the pattern using zeros for “off” dots and ones for “on” dots. If the pattern is more than eight dots wide, write the zeros and ones of each row in groups of eight as you copy the pattern. In our example, the result is this:

								=	00000000
		*				*		=	00100010
	*		*		*		*	=	01010101
*				*				=	10001000

---

**Convert to number**

Treat each group of eight digits in each row like a binary number. Using a calculator or a conversion chart (there is a conversion chart in the online help facility for SfShell), change each eight digit binary number into a decimal number. If the rows have more than one group of eight digits, separate the resulting decimal numbers with commas. Our example becomes:

00000000 = 0  
00100010 = 34  
01010101 = 85  
10001000 = 136

---

**Rewrite**

Use the decimal numbers to create the pattern command. Optionally, you may wish to add the pattern to the configuration file (as described below). The decimal numbers for each row are separated by commas and the rows are separated by semicolons. The pattern we set out to create can be specified as follows:

0;34;85;136

Remember that you can use preview to look at the pattern before creating a font with it. This is a good way to check that you did the conversion correctly.

---

## 35. Sftware Registration

---

Registering shareware is an investment. Your registration will provide the support and encouragement required to continue the development of Sftware. The Sftware utilities represent an investment of more than two years of my time and effort. You get the results of this toil for a fraction of what a commercial package would cost. Plus, you get the benefits of a try-before-you-buy license agreement. If you continue to use the Sftware utilities, you are required to register them.

Return the enclosed order form with your check or money order today!

---

### 35.1. Usage

The SPS-Reg registration program requires key information that will be mailed to you when you register the Sftware utilities. You cannot make any use of the program until you mail in your registration.

Sftware is provided under a lifetime registration policy. Your registration contributes directly to the future growth of Sftware. Every registration is good for all future versions of Sftware. Register once. Register now!

---

## 37. Glossary

---

<b>ASCII</b>	ASCII stands for the American Standard Code for Information Interchange. Text files are usually referred to as being “plain ASCII” if they contain no additional formatting information. The CONFIG.SYS and AUTOEXEC.BAT files on your boot disk are examples of a plain ASCII files. The spreadsheets, database files, or word processing documents produced by large application programs are generally <i>not</i> plain ASCII.
<b>baseline</b>	The baseline is an imaginary line upon which each character rests. Characters that appear next to each other are (usually) lined up so that their baselines are on the same level. Some characters extend below the baseline (“g” and “j”, for example) but most rest on it.
<b>bitmap</b>	<p>A bitmap is an array of dots. If you imagine a sheet of graph paper with some squares colored in, a bitmap is a compact way of representing to the computer which squares are colored and which are not.</p> <p>In the context of softfonts, the dots are always black and white. In a bitmapped softfont, every character is represented as a pattern of dots in a bitmap. The dots are so small (300 dots-per-inch, usually) that they are indistinguishable on the printed page.</p>
<b>bounding box</b>	Every character in a bitmapped softfont is represented as a pattern of dots in a rectangular grid. The bounding box is an imaginary box just large enough to hold the character. The box is as wide as the widest row of dots and as tall as the tallest column of dots.
<b>character</b>	A character is an individual symbol in a font. The letter “A” is a character. So is a period. All of the printed symbols that can appear in a font are characters. They can also be called glyphs.
<b>child process</b>	When one program directly runs another program (as when SfShell runs SfFx to perform a requested special effect), the program that is run (SfFx in this case) is called the child and the program that did the running (SfShell) is called the parent.
<b>command line</b>	When you type a command at the DOS prompt, you are entering information on “the command line.” Command line parameters and command line options are things that you type after the name of the command. For example, if you type “ <b>edit letter.txt</b> ”, “ <b>edit</b> ” is the command and “ <b>letter.txt</b> ” is a command line parameter.
<b>decimal</b>	Decimal refers to the number base composed of ten symbols (0-9). Normal, ordinary math is performed in decimal (which can also be referred to as base 10).
<b>device</b>	A device is a special piece of hardware that exists (either physically or logically) and can send and/or receive data. Your printer is a device. So is your modem. Your computer also includes several logical devices (for example, the <b>NUL</b> device which is an infinite sink and an empty source—that means you can always write to it (it never fills up) and you can never read from it (you always get “end-of-file”)).
<b>download</b>	Downloading is the process of transferring information from one device to another. This transferral is called downloading when the transfer flows from a device of (relatively) more



<b>memory, extended</b>	Extended memory is memory above the 1 megabyte boundry in your computer. Sftware cannot directly use extended memory. Many programs exist which map extended memory as expanded memory. For more information about expanded memory, see <i>EMS</i> .
<b>memory, main</b>	Main memory is the DOS memory below 640K in your computer. This is the area where normal DOS programs run. The DOS <code>chkdsk</code> program, for example, reports the amount of main memory that is free.
<b>octal</b>	Octal refers to the number base composed of eight symbols (0-7). Octal is sometimes called base 8.
<b>pathname</b>	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 <code>tr100.sfp</code> is the name of a file in the directory <code>\fonts</code> on drive <code>d:</code> , then <p style="text-align: center;"><code>d:\fonts\tr100.sfp</code></p> is the pathname of <code>tr100.sfp</code> .
<b>scalable font</b>	A scalable font, unlike a bitmapped font, is defined mathematically and can be rendered at any requested size (within reason). Sftware can download and show scalable fonts but other manipulations (including on-screen previewing) are not possible (at this time).
<b>selection sequence</b>	Your laser printer can print many different fonts. Some of the fonts are built in, some may come from a cartridge and many can be downloaded. In order to tell the laser printer which font you want text to be printed in, you must send it a selection sequence. The selection sequence describes, in a well defined, precise manner, the typeface, symbol set, height, width, style, and degree of boldness that you want.
<b>softfont</b>	A softfont is a bitmapped or scalable description of a typeface or font. They can be downloaded to your printer and used just like any other printer font. Unlike built-in and cartridge fonts, softfonts use memory inside your printer. Downloading a lot of softfonts may reduce the printers ability to construct complex pages.
<b>symbol set</b>	The symbol set of a font describes the relative positions of individual characters within the font. Since there can only be 256 characters in any font, and there are well over 256 different characters used in professional document preparation, there needs to be some way to map characters into positions within the font. The symbol set serves this purpose. It identifies the “map” used to position characters within the font.
<b>typeface</b>	A typeface is generic term for a collection of symbols with a similar style. Times Roman and Helvetica are typefaces. Contrast with font.

---

## Table of Contents

---

1. Introduction .....	1
2. Conventions Used in This Manual .....	4
3. Configuring Sftware 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 Tilt Effect .....	50
31. The Hollow Three-D Drop Shadow Effect .....	51
32. The Filled Three-D Drop Shadow Effect .....	52
33. The Vertical Fade/Mist Effect .....	53
34. Patterns .....	54
35. Sftware Registration .....	57
36. Contacting the Author .....	58
37. Glossary .....	59

# Sfware Order Form

Name: \_\_\_\_\_ Phone: (\_\_\_\_\_) \_\_\_\_\_-

Company: \_\_\_\_\_ Email: \_\_\_\_\_

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	_____
<b>Utility bundles:</b>			
_____ SFShell (menu interface shell) and <i>all</i> 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  
#42I Southwood Apts  
Amherst, MA 01002