

Notice:

This document is written and maintained by Small Planet Software. All rights reserved. No part of this publication may be reproduced, photocopied, stored in a retrieval system, or transmitted, in any form or by any means except those provided for by the shareware license agreement of the accompanying software.

Copyright © 1991 by Small Planet Software

All Rights Reserved

Although every reasonable precaution has been taken in the preparation of this document, no warranty of any kind is made with regard to the use of this material, including, but not limited to, the implied warranties of merchantability or fitness for a particular purpose. No patent or copyright liability is assumed with respect to the use of the information contained herein or the use of the accompanying software.

Acknowledgments:

I would like to thank Tom Bruhns and Philippe Weil for their invaluable assistance. They were my primary guinea pigs as the product went through alpha and early beta testing. Without their patience and helpful comments, neither this software nor this manual would be what they are today.

All trademarks used within this document are the trademarks of their respective owners.

Packing List

Sfware is distributed in four archive files. The archives are named **SFWverP1**, **SFWverP2**, **SFWverD1**, and **SFWverD2**. In each file, the *ver* is replaced by the Sfware version number. Every archive contains a file called **PACKING.xx** that lists the files that should be present in the archive. Please make sure that you have complete archives before you proceed to install Sfware.

The “P” archives contain the Sfware programs and *both* are required in order to install Sfware. The “D” archives contain documentation. The “D1” archive documents the SfShell interface, the “D2” archive documents the individual utilities. Please refer to the file **PRINT-DOC** in **SFWverP1** for instructions describing how to print the documentation.

1.2. Installation

Making Backups

Like any software package, it is always advisable to make backup copies of the distribution diskettes or distribution archives. This is especially important if you use an “on the fly” compression program to compress executable files (e.g. PkLite). Sfware cannot be registered after it has been compressed—you will need the original programs in order to register Sfware. There is no compelling reason not to compress the programs after you have registered them.

Hard Drive Installation

Create a subdirectory on your hard drive for the Sfware utilities; it does not matter what drive you install onto or what you name the directory. For the purpose of this manual, the directory **D:\SF** is assumed.

Copy all of the files from the distribution diskette (or from the distribution archive) into the Sfware directory.

Floppy Disk Installation

Copy all of the files from the distribution diskette (or from the distribution archive) onto a floppy disk. This manual assumes that Sfware has been installed in the directory **D:\SF** but it is not necessary to install Sfware onto a hard disk.

Due to space limitations on floppy disks, it may not be possible to place all of the files on a single diskette. If is the case, it is recommended that you put **SFSHELL.EXE** and **SFSHELL.HELP**, on one floppy and all of the other utilities on a second floppy. If you do not plan to use the shell, you will not need the SfShell files on a diskette.

If you use a floppy-only system, you will only be able to use the SfShell program if you have sufficient expanded memory (EMS) for SfShell to use a swapping space when it runs the other utility programs. SfShell requires either sufficient EMS *or a non-removable* disk for swapping.

Splitting the Sfware utilities across two floppies does not present any real technical difficulties (aside from the location of swapping space) but you should read the *Configuration* chapter carefully to make sure that you have set things up properly. In particular, you will need to tell SfShell where the utility programs are located.

2. Conventions Used in This Manual

The standalone utilities have a steeper learning curve than the SfShell interface. However, if you are comfortable with command line utilities in general, the Sftware utilities should not be too difficult to master. The advantages of mastering the learning curve are that the Sftware utilities have a few more options when used from the command line than they do from within the shell and the command line utilities allow you to work with groups of files efficiently.

In terms of getting something done, all of the command line utilities are run in the same general way. Basically, you type:

UTILITY *some-parameters some-options*

wherever you type DOS commands[†].

The sections that follow will describe the conventions used in this manual for describing what *some-parameters* and *some-options* are and how to enter them.

2.1. Typographic conventions

typewriter

Typewriter type is used within this manual to denote explicit words or commands or file-names that you type exactly the way they appear in this manual. In this manual, **FRUIT** means you type **F R U I T**, whereas a *fruit* (italics are described below) might mean **apple**, or **pear**, or *any* specific fruit.

italics

Italics are used to name a general “class” of things. If a command in this manual contains a word in italics, you should replace that word with a concrete example of “one of those things” when you type the command. For example, a *fontname* in this manual means any valid, existing softfont; you should type the name of an existing font file.

Occasionally, italics are used for emphasis (as they are in general typography) but it will be clear from the context when that is the case.

[[brackets]]

The stylized square brackets denote optional parameters. You should only type what appears within the brackets when you want to use the associated optional feature.

[†] This is generally the DOS prompt, but many user-friendly interfaces to DOS exist and if you are using one of them, you can type these commands wherever you would type any general DOS command (like **DIR** or **WP**).

2.3. How parameters and options are used

Parameters and options come in two flavors: those with associated values and those without. Parameters (or options) without associated values are either present or absent; that provides all that is required to interpret the option. The `/w` option on the DOS `DIR` command is an example of a parameter without an associated value. If the `/w` option is present, it means print a wide directory listing; if it's absent, don't print a wide directory listing. Many Sfware utilities have parameters (or options) without associated values.

Parameters (or options) that do have associated values have the general form:

/parameter-name:value

In this case, parameters (or options) are introduced by a forward slash and separated from the value by a colon. Spaces are not allowed anywhere in the parameter-name or the value.

In the descriptions of parameters and options that have values associated with them, the sample value given in the description tries to indicate what kind of value is expected. These fall into four general categories: filenames or fontnames, numbers, patterns, and ranges. Filenames and fontnames are described above, patterns are described in the pattern chapter, and ranges are described in the range chapter. Numbers are indicated by `#` or `%` in the command description. Numbers indicated by `%` should be in the range 0-100.

Parameters and options can be abbreviated to their shortest unambiguous name. For example, the `/device:filename` option can be abbreviated to `/d:filename` if no other option begins with "d". (It could also be abbreviated to `/de`, `/dev`, etc.)

2.4. Common Options

There are a handful of options that almost all of the Sfware utilities share. These options and their meanings are described below. Every utility indicates whether or not it accepts these options without repeating the description each time. This convention, in conjunction with the conventions mentioned above, makes the process of describing each utility much easier and directs the reader more precisely to the information that is specific to each utility.

`/@:`

The `/@` option specifies an input filename. The input file contains Sfware commands. The format of the input file is discussed in the configuration chapter.

`/!:`

The `/!` option specifies a message filename. If a message filename is given, the Sfware utilities will write warning and error messages to this file. It is used primarily by the SfShell program to capture error and completion messages.

`/ $`

The `/ $` option displays registration information for the Sfware utilities. If you are using an unregistered program, this information will be displayed automatically. Please register your shareware!

3. Configuring Sftware and Using Command Files

In order to make Sftware easier to use, all of the programs read a configuration file each time they are executed. This configuration file gives you the flexibility to assign default values to many of the options and parameters of each program.

3.1. Name of the configuration file

All of the utilities can share the same configuration file. However, rather than hardcoding the name of the configuration file, Sftware relies on the existence of a DOS environment variable to determine the name of the configuration file. Each Sftware utility expects the DOS environment variable **SFCFG** to name the complete drive, path, and filename of a suitable configuration file. For example, if you make a configuration file called **SF.CFG** and you put it in the **D:\SF** directory, the DOS command **SET SFCFG=D:\SF\SF.CFG** would be appropriate.

If the DOS environment variable **SFCFG** is undefined, each of the utilities looks for a configuration file with the same name as its executable file and the extension **.CFG**. For example, **SFFX.EXE** looks for **SFFX.CFG**.

Special note for DOS 2.xx users

In versions of DOS prior to version 3.xx, it was not possible for a program to find out the name or directory of its executable file. If **SFCFG** is undefined, the utilities will look in the current directory for configuration files. It is especially important to define **SFCFG** if you are not using DOS 3.xx or later.

3.2. Using SfConfig

Frequently, the most difficult part of installing new software is the task of configuring it to work correctly in your system. This may be true of Sftware as well. In an effort to make the *initial* configuration as painless as possible, Sftware comes with the SfConfig program. SfConfig should be run *after* the **SFCFG** environment variable, discussed above, has been set.

SfConfig will create a configuration file initialized with appropriate defaults and allow you to select, interactively, the laser printer that you use, the print device that you use, and name of your softfont directory. These are the most site-specific configuration options.

SfConfig can be run again to change any one of these values; it will not change anything else in the configuration file that you have changed manually since the first time that you used SfConfig.

The following three settings can be made from within SfConfig:

3.4. Compress

Usage: *program* COMPRESS=YES or NO
Used by: SFFx, SFRotate

The Sftware utilities that write new softfont files use this flag to determine if the softfonts should be written in PCL4/5 compressed format or in the older, non-compressed format. Compression can produce very dramatic decreases in the amount of disk space required for a softfont. However, the compressed fonts are only recognized by LaserJet printers that are PCL4 compatible. The LaserJet Series II *is not* PCL4 compatible. Note, however, that the Sftware utilities provide for decompression “on the fly” in most cases. Please consult the section about downloading fonts for more information.

3.5. Device

Usage: *program* DEVICE=*filename*
Used by: SfLoad, SfShow

Names the output device for Sftware utilities that interact directly with the printer. The most common value is **LPT1**, but any DOS file or device name may be used.

3.6. Esc

Usage: *program* ESC=YES or NO
Used by: SfDir

Controls how SfDir displays font information. If **ESC** is **YES**, escape sequences are displayed by default. Otherwise a readable, text description is displayed by default.

3.7. FontDir

Usage: *program* FONTDIR=*directory*
Used by: SfShell, SFCmpr, SFFx, SfLoad, SFRotate, SfShow

Names the DOS subdirectory where HP LaserJet softfonts are located. This is the default input and output directory for Sftware utilities that read or write softfont files.

3.8. FontExtn

Usage: *program* FONTEXTN=*ext*
Used by: SfShell, SFCmpr, SFFx, SfLoad, SFRotate, SfShow

Names the default filename extension for softfont files. If you specify either an input softfont name or an output softfont name that does not include an extension, the Sftware utilities will append this extension to the filename. Note: it is possible to specify that a file should not have *any* extension by ending the filename with a period.

The following table lists all of the graphics cards and the modes associated with them. In general, it is not necessary to specify a graphics mode since the highest resolution mode is selected by default:

Card	Mode	Resolution	Palette
CGA	0	320x200	0
CGA	1	320x200	1
CGA	2	320x200	2
CGA	3	640x200	3
CGA	4	640x200 (default)	2 colr
MCGA	0	320x200	0
MCGA	1	320x200	1
MCGA	2	320x200	2
MCGA	3	320x200	3
MCGA	4	320x200	2 colr
MCGA	5	640x480 (default)	2 colr
EGA	0	640x200	16 colr
EGA	1	640x350 (default)	16 colr
EGA64	0	640x200	16 colr
EGA64	1	640x350 (default)	4 colr
EGAMONO	3	640x350 (default)	2 colr
HERC	0	720x348 (default)	2 colr
ATT	0	320x200	0
ATT	1	320x200	1
ATT	2	320x200	2
ATT	3	320x200	3
ATT	4	640x200	2 colr
ATT	5	640x400 (default)	2 colr
VGA	0	640x200	16 colr
VGA	1	640x350	16 colr
VGA	2	640x480 (default)	16 colr
PC3270	0	720x350 (default)	2 colr
IBM8514	0	640x480	256 colr
IBM8514	1	1024x768 (default)	256 colr

3.14. Numbers

Usage: *program* **NUMBERS**=*base*

Used by: SfShow

The **numbers** parameter is used by SfShow to select the numeric base of the numbers printed around the reference grid. Valid options are **hex**, **oct**, **dec**, and **none** for hexadecimal (base

3.19. Sentence

Usage: *program* **SENTENCE**=*string*
Used by: SfShell, SfView, SfShow

Identifies the sentence to be displayed on graphical font preview screens and printed on the reference page. The default sentence is: The quick red fox jumped over the lazy brown dog. I said (very loudly), “THE QUICK RED FOX JUMPED OVER THE LAZY BROWN DOG!” How many times? 0, 1, 2, or 3456789 times.

3.20. SwitchChar

Usage: *program* **SWITCHCHAR**=*char*
Used by: SfShell, SfCmpr, SFFx, SFLoad, SfRotate, SfShow, SfView, SfInfo

Identifies the switch character. It must be set to either “/” or “-”. Under MS-DOS, “/” is recommended. For more information about the uses of the switch character, consult the “technically speaking” note at the end of the *Conventions* chapter.

3.21. Style *name*

Usage: **STYLE** *name*=*number*
Used by: SfShell, SfShow, SFFx

The “style” of a softfont is one of the font parameters that is used to distinguish between two otherwise identical softfonts. The values defined by HP are “upright,” “italic” and “oblique.” Using SFFx to create variations on a font can potentially create two fonts that are indistinguishable from each other. For example, “hollowing” a softfont does not change *any* of its font characteristics. The **STYLE** parameter tells the SFFx what style value to use in the font header for each effect. In this way, it will always be possible to tell the old and new fonts apart. The name of the style must be one of the following: **Bold**, **Fix**, **Fill**, **Fill3d**, **HalfTone**, **Hollow**, **Hollow3d**, **Invert**, **Mirror**, **OutLine**, **Prop**, **Resize**, **Reverse**, **Shade**, **Shadow**, **Stripe**, and **Threed**. The style value can be any number between 0 and 255. If the effect changes some other characteristic of the font, it is not necessary to change the style; this is indicated with a style value of 0.

3.22. Typefaces

Usage: *program* **TYPEFACES**=*filename*
Used by: SfShell, SfInfo, SfShow

The typefaces parameter names the file that lists typeface names. Every softfont has a typeface number. A name is associated with each typeface number; this is the name displayed by SfShell in the typeface column, and by SfInfo and SfShow. Because the number of typefaces is growing and is subject to change, you can supply an additional typeface list that identifies any and all typeface numbers. Sftware is distributed with the file **TYPEFACES.LST** that contains all of the Hewlett Packard typeface names defined as of PCL5. If you have an old or non-standard softfont, this name may not accurately reflect the style of the characters contained in the font.

3.24. Command Files

The standalone utilities, whether they are run from the command line directly or invoked automatically by SfShell, accept all of their input on the command line. Since this imposes a severe limit on the amount of input that can be provided, the utilities also accept input from a command file.

SfShell creates a command file automatically to communicate with the standalone utilities and you can use them outside of SfShell, but understanding what they are and how they work is not important to using Sftware. Feel free to skip this section.

The standalone utilities accept the name of the command file on the `/@:filename` option.

If a command file is used, the utility will read commands from the file as if they were typed as parameters. The format of the command file is simple: each line should begin with an asterisk followed by the name of the utility followed by a space. The rest of each line is interpreted exactly as if it were typed on the command line. Because each line identifies which utility it is for, the same command file can be passed to several utilities. Each utility will only use the lines that are intended for it.

For example, the command file below downloads several fonts:

```
*sfload tr* /expand
*sload tr* /expand /landscape
*sload logo.sfp /expand
```

If this command file is saved as `AUTOLOAD.CMD`, I would tell SfLoad to execute it by entering:

```
SFLOAD /@:AUTOLOAD.CMD
```

In general, this ability is of little use beyond downloading fonts (every morning, for example). However, the SfShell program makes extensive use of this feature to pass parameters to child processes when it executes the individual utilities to perform actions for the user.

<hr/> /portrait	The portrait option rotates the softfont to portrait orientation before downloading it. This option has no effect if the font is already portrait.
<hr/> /landscape	<p>The landscape option rotates the softfont to landscape orientation before downloading it. This option has no effect if the font is already landscape.</p> <p>Note: downloading both orientations <i>does not</i> imply that you will be able to use both orientations on the same page. The LaserJet Series II printer, for example, cannot print both portrait and landscape fonts on the same page.</p>
<hr/> /temp	The temp option specifies that downloaded fonts should be made temporary. SFLoad usually makes fonts permanent. Note: temporary fonts are automatically deleted every time a printer reset command is used so this option is of limited usefulness.
<hr/> /id:#	The id option sets the starting font-id number. The default font-id is zero. Every font must have a unique id number; if more than one font matches the <i>fontmask</i> , the first font will have the specified id and every subsequent font will have an id number one greater than it's predecessor.
<hr/> /device: <i>file-name</i>	The /device option specifies the output device. The output device is most frequently LPT1 . SFLoad uses the specified filename (which may name a device) as the destination for downloaded fonts.
<hr/> Other Options	The SFLoad program accepts the /@ , /! , /replace , /quiet , /verbose , and /\$ options.

The range command comes in three flavors:

5.2. `/range:c-c`

This is the "letter-to-letter" syntax, i.e. `/range:a-z` limits the scope to the lowercase alphabet, `/range:A-E` limits it to the uppercase letters A, B, C, D, and E.

5.3. `/range:nn-nn`

This is the "ascii value-to-ascii value" syntax, i.e. `/range:33-47` limits the scope to the punctuation symbols between "!" and "/", inclusive. In this syntax, the ascii value must be padded with zeros to make it at least two digits long. For example, the range `/range:0-9` is in letter-to-letter form and limits the scope to the decimal digits, ascii values 48 to 57, inclusive. The range `/range:00-09` limits the scope to the control characters NUL to HT (tab).

5.4. Mixed forms

The forms can be mixed, `/range:00-@` limits the scope of the effect to the first sixty-five ascii values, the null character (ascii 0) to @ or 00 to 64.

The range effect is limited to a specific, contiguous subset of the ASCII character set. That is, you can specify any single range but you *cannot* specify an "exception range" (e.g. do all the characters *except* the lowercase letters) or two or more discontinuous ranges (e.g. do all the upper case letters *and* all the lower case letters).

5.5. Technically Speaking

Most fonts do not contain a real blank space character. The LaserJet printer moves over by the default HMI everytime it encounters a character that does not exist in the current font; most fonts rely on the fact that the default HMI is exactly one space wide. This can create an unpleasant, choppy appearance if a special effect (e.g. halftoning) is applied that modifies the white background of each character.

The SFFx program accepts the special option "`/_`" to correct this problem. The "`/_`" option forces SFFx to create a physical blank space character if the font does not already contain one.

7. SFROTATE: Rotating Fonts

Softfonts come in two orientations: portrait and landscape. Newer LaserJet printers are capable of “automagic” internal font rotation but older LaserJets and some compatibles do not have this ability. Sftware provides the ability to convert from one orientation to the other. The SfRotate program can convert from one orientation to the other.

7.1. Usage

SFROTATE *fontmask* `[[outmask]]` `[[options]]`

7.2. Options

outmask

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

Other Options

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

/device: *file-name* The **/device** option specifies the output device. The output device is most frequently **LPT1**. SfShow prints to the specified filename.

/refset: *set* For decorative or special purpose fonts, it may be helpful to have an additional reference character printed (in plain ASCII) next to each symbol in the chart. The **/refset** option allows you to specify the character set to use for reference marks on the printout. If reference marks are used, the reference character for each position in the font will be printed in the upper right hand corner of each cell on the grid. For example, using the **/refset:8u** option 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 *do not* want reference characters to be printed, use **/refset:none**.

/numbers: *base* The **/numbers** option allows you to specify the numeric base of the reference numbers printed around the grid. Valid options are **hex**, **oct**, **dec**, and **none** for hexadecimal (base 16), octal (base 8), decimal (base 10) and no reference numbers respectively. The **numbers** configuration variable (discussed in the configuration chapter) is provided to specify a default number base.

Other Options The SfShow program accepts the **/@**, **/!**, **/replace**, **/quiet**, **/verbose**, and **/\$** options.

8.3. Technically Speaking

When multiple reference pages are required, SfShow attempts to use the minimum number of pages, however, there are a few “hidden” constraints on how it selects the first character for each page. In particular, it will not skip characters on any single page (i.e. if the font defines ABCEFG but not D, SfShow will not print ABCEFG on a reference page without an intervening blank space where the D would be if it was defined. It wouldn't be difficult to provide this option but it would make numbering the grid much more difficult.

Esc

You can leave at any time by pressing **Esc**.

PgUp/PgDn

Pressing **PgDn** moves the range of characters displayed forward by one “screenfull.” If ASCII 255 is currently in the display, pressing **PgDn** has no effect. Pressing **PgUp** moves the range of characters displayed backward by one screenfull. If ASCII 0 is currently in the display, pressing **PgUp** has no effect.

Alt + A

If you are displaying the font in a graphics mode that has the same number of pixels-per-inch both horizontally and vertically across the display, the **Alt+A** key combination is not available.

If the number of pixels-per-inch horizontally and vertically is not the same, (i.e. the display has a non-square aspect ratio) it is impossible to display the characters without some distortion because the softfont *is* defined with the same number of pixels-per-inch both horizontally and vertically.

There are two kinds of distortion: stretch-distortion and “reduced resolution” distortion. If *every* pixel of each character is displayed, the letters will be stretch-distorted by the fact that the pixels are “closer together” on the screen in one direction than the other. Alternatively, some rows or columns of pixels can be removed to avoid stretch distortion; characters drawn this way suffer from distortion because they are printed at a reduced resolution.

The **Alt+A** key-combination alternates between these two types of distortion.

10. SFINFO: Displaying Softfont Information

If you have difficulty printing a particular font, SfInfo can help pinpoint the source of the problem. SfInfo displays the contents of the softfont header and the header of each character in the font. In addition, SfInfo examines the font looking for possible printer incompatibilities. New printers have a much more relaxed opinion about what constitutes a valid font. A font that works on a LaserJet III may not work on a Series II; SfInfo will be able to tell you why.

10.1. Usage

SFINFO *fontmask* [*options*]

Other Options

The SfInfo program accepts the /\$ option.

10.2. Running SfInfo

Running SfInfo displays a panel something like the following:

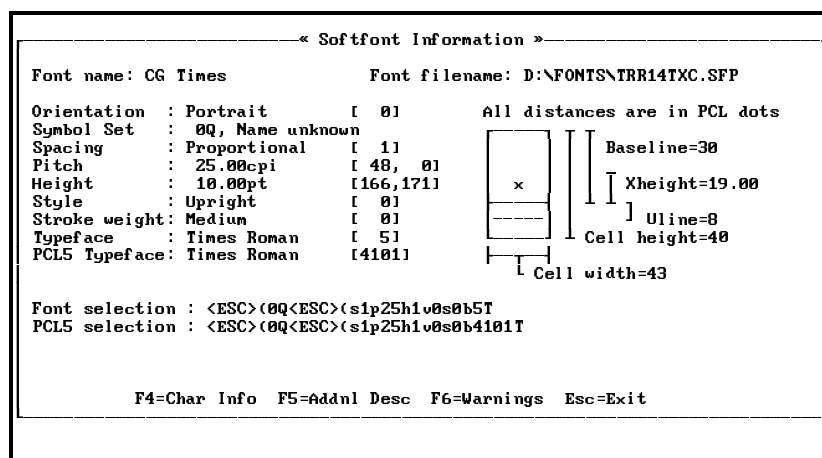


Figure 10.1. SfInfo Main Panel

Esc

You can leave at any time by pressing **Esc**.

11. SFDIR: Softfont Directory Listings

The Sfdir program provides a useful alternative to the standard DOS DIR command for softfonts. Sfdir prints the font characteristics of each softfont that matches the *fontmask*.

11.1. Example

```
Directory of D:\FONTS\*. *
ARB010AAA ESP Port 10J Fix 12.00cpi 9.96pt Up Medium Courier
TRB1CTXC SFL Land 0Q Pro 12.00pt Up Bold TnsRmn
CMR10 SFP Port 0U Pro 9.96pt Up Medium TnsRmn
TRB1CTXC SFP Port 0Q Pro 12.00pt Up Bold TnsRmn
TRR00USC SFP Port 0U Pro 6.00pt Up Medium TnsRmn
TRR0WUSC SFP Port 0U Pro 8.00pt Up Medium TnsRmn
TRR14TXC SFP Port 0Q Pro 10.00pt Up Medium TnsRmn
TRR1CTXC SFP Port 0Q Pro 12.00pt Up Medium TnsRmn
TRR1KTXC SFP Port 0Q Pro 14.00pt Up Medium TnsRmn
TRR1KUSC SFP Port 0U Pro 14.00pt Up Medium TnsRmn
TRR1KXXC SFP Port 0Q Pro 14.00pt Up Medium TnsRmn
UNB1CTXC SFP Port 0Q Pro 12.00pt Up Bold Univers
UNR00USC SFP Port 0U Pro 6.00pt Up Medium Univers
UNR0WUSC SFP Port 0U Pro 8.00pt Up Medium Univers
UNR14USC SFP Port 0U Pro 10.00pt Up Medium Univers
UNR1CTXC SFP Port 0Q Pro 12.00pt Up Medium Univers
UNR1KUSC SFP Port 0U Pro 14.00pt Up Medium Univers
NB010AAA USP Port 0U Pro 9.96pt Up Medium Typeface# 210
VGA20 SFP Port 10U Fix 20.00cpi 1.68pt Up Medium Courier
VGA20SH SFP Port 10U Fix 20.00cpi 1.68pt 14 Medium Courier
20 Font(s) 268420 bytes
```

11.2. Usage

SFDIR *fontmask* [*options*]

11.3. Options

/esc

If Sfdir is run with the */esc* option, it prints the LaserJet escape sequence required to select each font instead of a textual description of the font characteristics. In the escape sequence, a raised dot is used to represent the ESC character (ASCII 27d).

/noesc

With the */noesc* option, Sfdir prints a textual description of the font characteristics for each font that matches the *fontmask*. This is generally the default.

Other Options

The Sfdir program accepts the */quiet*, */verbose*, and */\$* options.

13. The Fixed Spacing Effect

Fixed spacing uses the same width for each character in the font. This is the opposite of proportional spacing in which each character is given a width appropriate to its appearance. In a fixed spaced font, all characters have the same width. The fixed spacing effect creates a fixed spaced font from a proportionally spaced font. This can be useful if you need to line up columns of characters, for example, although it's generally better to use a font specifically designed for fixed spacing.

13.1. Usage

SFFX *fontmask outmask* **FIX** `[[options]]`

13.2. Options

There are no options for this effect.

13.3. Technically Speaking

In the fixed spaced version of the font, all characters have the maximum cell width. Bitmaps that are narrower than the maximum cell width are adjusted to print as if they were centered in a box as wide as the maximum cell width.

15. The Halftone Effect

Halftoning a font can produce a wide variety of results. It is one of the most general effects in SFFx's repertoire. In brief, it allows you to specify the fill patterns for both the foreground and the background of two different regions of each character. This can create, for example, half-inverted characters.

15.1. Usage

```
SFFX fontmask outmask HALFTONE /start:# /stop:#  
/blackbot:pattern /whitebot:pattern [options]
```

15.2. Parameters

Every character is divided into two areas, a selected area and a non-selected area. Within each area, two shading patterns are applied—one to the currently black portion of the character cell (the character itself) and one to the white portion of the character cell (everything else).

Please refer to the *Patterns* chapter elsewhere in this manual for more information about patterns.

/start

The **/start** parameter specifies where the selected area begins. This value should be expressed as a percentage from the top of the tallest character in the font. For example, specifying 25 begins the selected area 1/4 of the way down from the top of the character, similarly, 50 selects a position halfway down the character and 67 selects a position 67% of the way down from the top of the character.

/stop

By analogy with **/start**, this option specifies where the select region ends. The **/stop** value should be larger than the starting value. The area between the start position and the stop position is the “selected region” of the character.

/blackbot

The **/blackbot** pattern replaces the foreground (black) area of the selected region. The **/blackbot** parameter can be abbreviated to **/bbot**.

/whitebot

The **/whitebot** pattern replaces the background (white) area of the selected region. The **/whitebot** parameter can be abbreviated to **/wbot**.

16. The Horizontal Fade/Mist Effect

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

16.1. Usage

`SFFX fontmask outmask HFADE /to:% [[options]]`

16.2. 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 across each character so that a 100% black saturation is achieved in the last column of pixels. Fade factors below 100% apply the fade more rapidly so that a 100% black saturation is achieved before the edge of the character. Conversely, fade factors above 100% draw the fade out so that it never reaches saturation.

16.3. Options

`/back`

By default, a horizontal fade begins with 0% black on the left edge of the character and proceeds towards 100% on the right edge (at a rate determined by “fade percent.” See above). If backwards fading is selected, the fade proceeds from right to left instead of left to right.

`/range`

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

16.4. Technically Speaking

The fade effect examines each pixel in the bitmap and decides randomly if the pixel should be turned off. In any given column, $100 * \text{ColumnNumber} * (\text{FadePercent} / 100) / \text{CharacterWidth}$ percent of the pixels are turned off.

18. The Invert Effect

Inverting a character creates a “reverse video” effect. However, the choice of patterns in this effect can dramatically change the result.

18.1. Usage

SFFX *fontmask outmask INVERT* `[[options]]`

18.2. Options

Both of the options for this effect are patterns. See the *Patterns* chapter elsewhere in this manual for more information.

/black

This pattern replaces all of the black areas of the character.

/white

This pattern replaces all of the white areas of the character.

/range

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

18.3. Technically Speaking

See the technically speaking section of the halftone effect for more information.

20. The Mist Effect

Misting a font “smudges” each character.

20.1. Usage

SFFX *fontmask outmask* MIST /mist:% *[[options]]*

20.2. Parameters

/mist

The mist percent determines what percentage of the character is misted (faded) out. Larger mist percentages remove more pixels than smaller ones. A 100% (or larger) mist percent removes all trace of the character.

20.3. Options

/range

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

20.4. Technically Speaking

This effect is identical to the horizontal and vertical fade effects with the exception that the fade percentage is calculated once and does not vary for each row or column in the bitmap.

22. The Proportional Spacing Effect

Proportional spacing is the opposite of fixed spacing. In a proportionally spaced font, each character is only as wide as its printed image, plus a small border. The proportional spacing effect creates a proportionally spaced version of a fixed spaced font.

22.1. Usage

`SFFX fontmask outmask PROP`

22.2. Options

There are no options for this effect.

22.3. Technically Speaking

In the proportionally spaced version of the font, all characters are four dots wider than the natural width of the bitmaps required to print each character (two dots on each side). Note: in many fonts, conversion from proportional spacing to fixed and back to proportional will yield a proportionally spaced font that is not as attractive as the original font since conversion to fixed spacing effectively destroys any special spacing information. For example, in many fonts the tail of a lower case letters like “j” and “g” are allowed to “hang back” below the character that precedes them. When a font is converted from fixed spacing to proportional spacing, there is no way to insert this kind of aesthetic hint automatically.

24. The Reverse Effect

Reversing a font creates backwards characters.

24.1. Usage

SFFX *fontmask outmask* REVERSE `[[options]]`

24.2. Options

`/range`

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

24.3. Technically Speaking

The reverse effect simply rotates each bitmap through its center. The left offset and delta-x values of each character are adjusted to keep the correct amount of space “in front of” and “behind” each character.

26. The Shade Effect

Shading a font replaces all off the “black” areas of a font with the specified shading pattern. This effect changes dramatically depending on the pattern that you select.

26.1. Usage

`SFFX fontmask outmask SHADE /pattern:pattern [[options]]`

26.2. Parameters

`/pattern`

All of the black areas of each character are replaced by the specified pattern. Please refer to the chapter on patterns elsewhere in this manual for more information about patterns.

26.3. Options

`/range`

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

26.4. Technically Speaking

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

28. The Stripe Effect

Striping places alternating white and black horizontal lines across each character in the font.

28.1. Usage

`SFFX fontmask outmask STRIPE /black:# /white:# [[options]]`

28.2. Parameters

`/black` Selects the width (in dots) of the black stripes.

`/white` Selects the width (in dots) of the white stripes.

28.3. Options

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

28.4. Technically Speaking

In each character, the stripes are adjusted so that a black stripe begins at the baseline. This assures that the stripes will line up when characters are placed next to each other. Note: a similar effect with vertical stripes can be created with the shade effect using an appropriate pattern.

30. 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.

30.1. Usage

`SFFX fontmask outmask HOLLOW3D /X:# /Y:# [options]`

30.2. 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-Offset`

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

30.3. Options

`/range`

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

30.4. 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.

32. The Vertical Fade/Mist Effect

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

32.1. Usage

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

32.2. 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.

32.3. 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.

32.4. 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.

34. 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!

34.1. 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.

36. Glossary

ASCII	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.
baseline	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.
bitmap	<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 still white.</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>
bounding box	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.
character	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.
decimal	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).
device	A device is a special piece of hardware that exists (either physically or logically) and can be communicated with. Your printer is a device. So is your modem. Your computer also includes several logical devices (for example, the NULL device which is an infinite sink and a null source—that means you can always write to it (it never fills up) and you can never read from it).
download	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 power to one of (relatively) less power. Sending new fonts to your printer so that it “learns” how to print characters in that font is called downloading.
EMS	EMS memory (also called LIM EMS) is expanded memory in your computer. EMS exists outside of normal DOS main memory. You must have a device driver to provide support for EMS. Sftware uses EMS memory to store font and action lists as well as for swapping when SfShell runs the other utilities.
file	A file is a collection of information stored on your disk. All the data that you ever save to disk is saved in a file. You can write to files and read from files.

scalable font	A scalable font, unlike a bitmapped font, is defined mathematically and can be rendered at any requested size (within reason). Software can download and show scalable fonts but other manipulations (including on-screen previewing) are not possible.
selection sequence	Your laser printer can print in 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 of font that you want.
softfont	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.
symbol set	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.
typeface	A typeface is generic term for a collection of symbols with a similar style. Times Roman and Helvetica are typefaces. Contrast with font.

