Notice:

Acknowledgments:

I would like to thank Tom Bruhns and Philippe Weil for their invaluable assistance. They were my primary gunea 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.

Packing List

Sfware is distributed in four archive files. The archives are named `SFW`*ver*`P1`, `SFW`*ver*`P2`, `SFW`*ver*`D1`, and `SFW`*ver*`D2`. 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 `PRINTDOC` in `SFW`*ver*`P1` 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.HLP`, 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

2.1. Typographic convensions

`typewriter`

Typewriter type is used within this manual to denote explicit words or commands or filenames 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.

boldface

Boldface is used to highlight words that appear in selection lists. It is roughly analogous to the way `typewriter` text is used to indicate things you should type; **boldface** indicates things you should select off of a list of choices.

[[*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.

2.2. Sections

Captured Screens

Many of the chapters contain "captured screens" to provide a context for the discussion of the choices available. These captured screens are taken directly from version 1.1 of SfShell.

3. Configuring Sfware

In order to make Sfware 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, Sfware relies on the existance of a DOS environment variable to determine the name of the configuration file. Each Sfware 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 Sfware as well. In an effort to make the *initial* configuration as painless as possible, Sfware 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. ActionListSize

Usage: *program* `ACTIONLISTSIZE=`*number*
Used by: SfShell

Controls the amount of memory reserved for the font action list. Each time you choose to do something to a font (download it, compress it, perform a special effect, etc.) that choice gets added to an action list. The actions in the action list get performed when you press F10 in SfShell. The `ACTIONLISTSIZE` can be large, but it is advantageous to keep it relatively small unless you have a lot of expanded memory (EMS). If it is too large, it will be written to disk which may have a considerable impact on program performance (especially on response time).

3.5. CommandFile

Usage: *program* `COMMANDFILE=`*filename*
Used by: SfShell

Specifies the name of the SfShell command file. The command file is used to communicate between SfShell and the utility programs. The command file can also be saved for later use to automatically re-run the selected actions.

3.6. Compress

Usage: *program* `COMPRESS=YES` or `NO`
Used by: SfFx, SfRotate

The Sfware 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 Sfware utilities provide for decompression "on the fly" in most cases. Please consult the section about downloading fonts for more information.

3.7. Device

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

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

3.13. 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.

3.14. 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`".

3.15. GraphForg

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

Controls the foreground color in graphics mode.

3.16. GraphGrid

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

Controls the color of the gridlines in the graphics display.

3.17. 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!).

3.18. MsgFile

Usage: *program* `MSGFILE=`*filename*
Used by: SfShell

SfShell forces all of the Sfware utilities to write error and completion messages to the message file that you specify. When you leave SfShell, this file will be displayed to give you a summary of the things that you did.

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

3.20. 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
```

3.21. Quiet

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

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

3.22. 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

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.

3.27. SfCmpr, SfFx, SfLoad, SfRotate, SfShow

Usage: *program name=filename*
Used by: SfShell

If the executable files for the Sfware utilities are kept in a different directory or drive than the SfShell executable (for example, if you are using the two-floppy disk setup described in the getting started chapter), these parameters should name the respective executable files. The filename given should be a complete filename with drive, path and extension. For example, if SfShell is in your utilities directory but you keep the other Sfware utilities in the directory `D:\SF`, then `SFSHELL SFCMPR` in the configuration file should be defined like this:

```
SFSHELL SFCMPR=D:\SF\SFCMPR.EXE
```

And analogously for all the other utilities.

3.28. SwapFile

Usage: *program* `SWAPFILE=`*filename*
Used by: SfShell

The swapfile parameter names the file that SfShell should use for a swapfile if it cannot swap to EMS. The swapfile filename should be a complete filename with drive and path. The swapfile *must* be on a non-removable medium. If you specify a swapfile on a removable medium, SfShell will not be able to swap and you will not be able to use the shell very effectively.

3.29. Typefaces

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

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, SfShow and SfDir. 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. Sfware 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.

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.

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 Sfware. 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 utilites. Each utility will only use the lines that are intended for it.

For example, the command file below downloads several fonts:

```
*sfload tr* /expand
*sfload tr* /expand /landscape
*sfload 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.

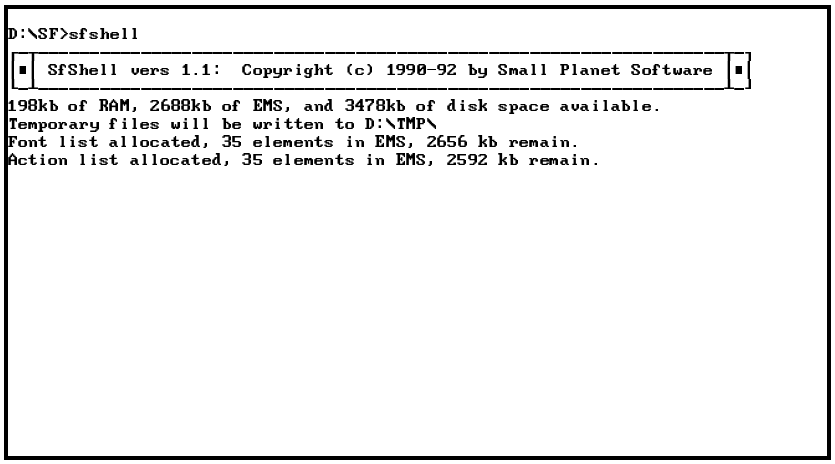


Figure 4.1. Initialization message

The column headings across the top of the main menu describe the primary characteristics of each font.

Heading	Font characteristic displayed
`Typeface`	Typeface name of the font
`Fontname`	The fontname in the softfont file
`Sty`	Style of the font
`Set`	Font symbol set
`O`	Font Orientation, Portrait or Landscape
`Bold`	Degree of "boldness" of the font
`Height`	Font size in points
`Pitch`	Font pitch (characters-per-inch) for fixed pitch fonts
`Filename`	Name of the softfont file

SfShell attempts to display informative names for each characteristic. However, if the value of a characteristic falls outside the bounds expected by SfShell, the numerical value of the characteristic will be printed in square brackets.

Either the typeface or the fontname can be displayed in the first column. The Tab key alternates between them. The fontname is assigned by the font designer and stored in the softfont header. Some fonts may not have a readable fontname.

For proportionally spaced fonts, the pitch is listed as `n/a` because the font has no fixed pitch. For scalable fonts, the height is listed as `n/a` for the same reason. The great majority of actions that can be selected for softfonts apply to bitmapped (non-scalable) fonts only. For example, SfShell cannot perform any special effects on scalable fonts. SfShell can download scalable fonts and show (print font summaries of) scalable fonts.

4.3. Selecting fonts

The highlight bar is used to select a font. You can only select one font at a time. The arrow keys move the highlight bar.

If you want to perform the same action on several fonts, you must select each font in turn and apply the action. If you want to perform more than one action on a single font, simply reselect the font.

4.4. Changing Directories

The initial font directory, *fontdir*, is either the default font directory specified in the configuration file or is selected with a command line option when running SfShell. You can change the current font directory by pressing F4 while the main menu is displayed.

4.6. Pulling the Trigger

After you have selected an action (as described above) for a softfont, pressing F10 will cause SfShell to perform the action. You can select more than one effect for more than one font before you press F10. If you do not press F10 before you leave SfShell, *no actions will be performed.* Later chapters describe exactly what happens when press F10 but you do not need to know how your actions are performed if you are always going to use the shell.

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.

Portrait

The **Portrait** option rotates the softfont to portrait orientation before downloading it. This option has no effect if the font is already portrait.

Landscape

The **Landscape** option rotates the softfont to landscape orientation before downloading it. This option has no effect if the font is already landscape.

Note: downloading both orientations *does not* 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.

6.1. 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.

Pressing F2 on any of the special effect panels that support the range option will present a list like the following:

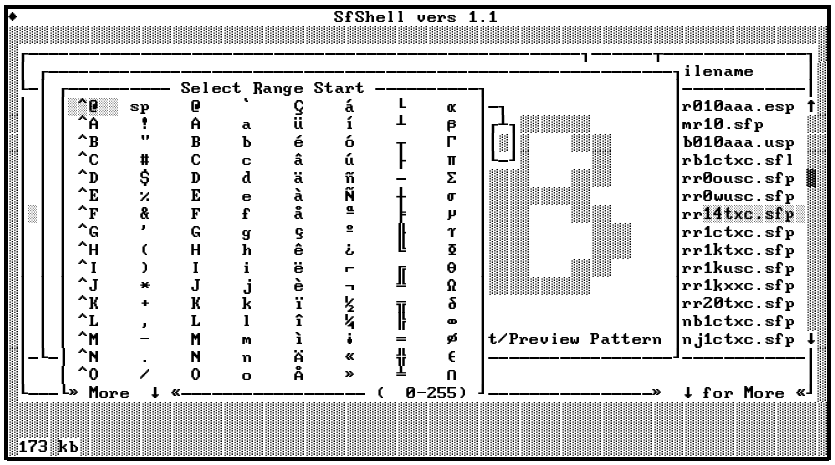


Figure 6.2. The Range selection menu

Use the arrow keys (and PgUp and PgDn) to move the highlight bar to the desired character then press Enter. The first time that you press Enter, you will be selecting the first character of the range and the second time you will be selecting the last character of the range.

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 discontiguous ranges (e.g. do all the upper case letters *and* all the lower case letters).

6.2. 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.

In several places, SfShell inserts a physical space for you to circumvent this problem. There is no way to control this action from within SfShell, but if you run SfFx directly you can have complete control.

8. 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. Sfware provides the ability to convert from one orientation to the other. The actual rotation is performed by the SfRotate utility.

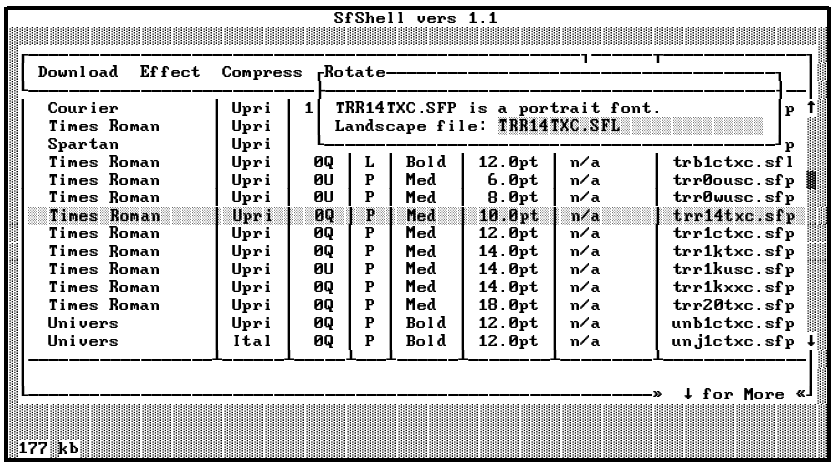


Figure 8.1. The Rotate Panel

The panel indicates the name of the current font and its orientation. You must enter the name of the font file which will contain the rotated font.

The default filename is the same as the original filename with the extension `SFL` if the resulting font will be landscape and `SFP` if the resulting font will be portrait.

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**.

No grid

The **No grid** option supresses grid lines on the reference page.

No Refset

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. 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. The **No Refset** option turns off the reference characters for this font.

You must specify the reference set in the configuration file.

9.2. Technically Speaking

When multiple reference pages are required, SfShell 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, SfShell 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 (read "impossible").

The reference numbers (printed around the chart) can be printed in hexadecimal, decimal or octal or they can be turned off. The `numbers` option (discussed in the configuration chapter) is provided to control this feature. At present, this option cannot be changed from within SfShell.

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.

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:

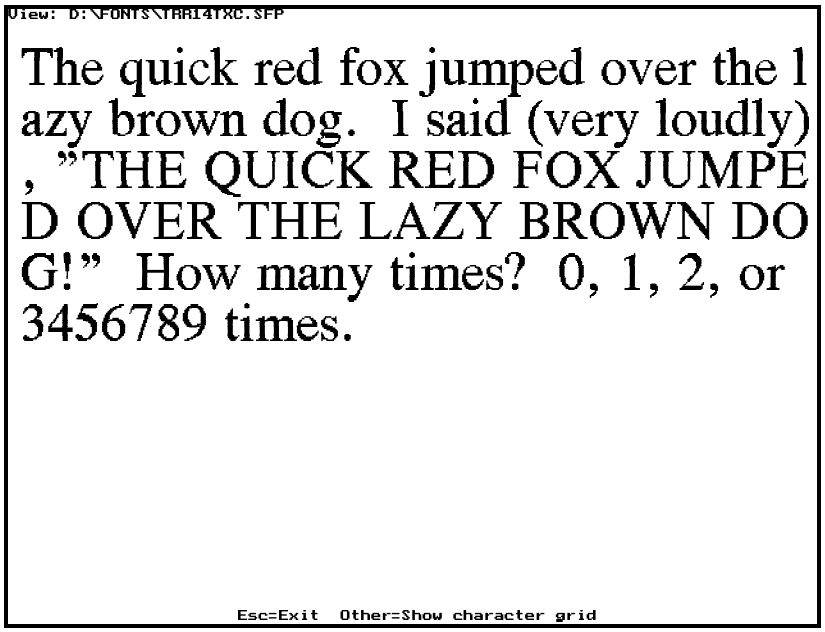


Figure 10.2. The view sentence display

Other

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

The additional info panel looks like this:

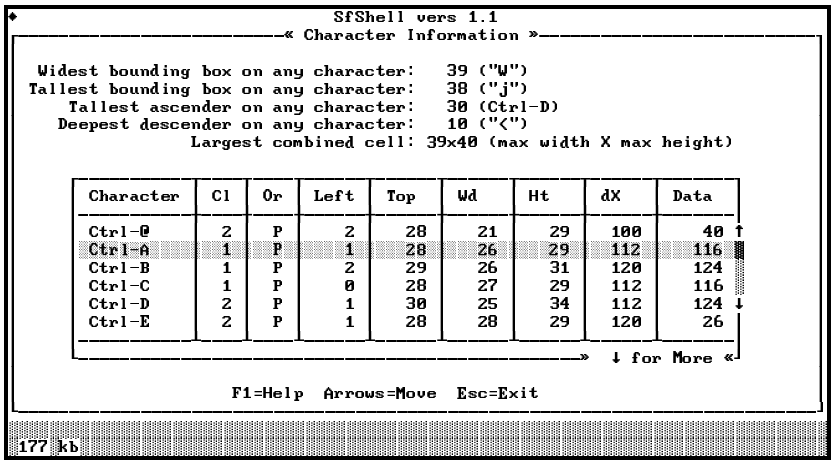


Figure 11.2. Additional Character Information Panel

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

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

F5

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

Not all fonts have additional information in the header.

F6

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

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.

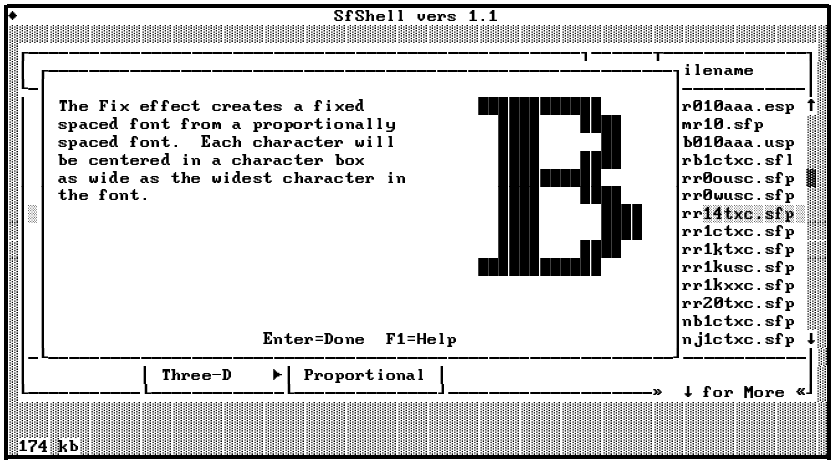


Figure 13.1. The Fixed Spacing panel

13.1. Options

There are no options for this effect.

13.2. 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.

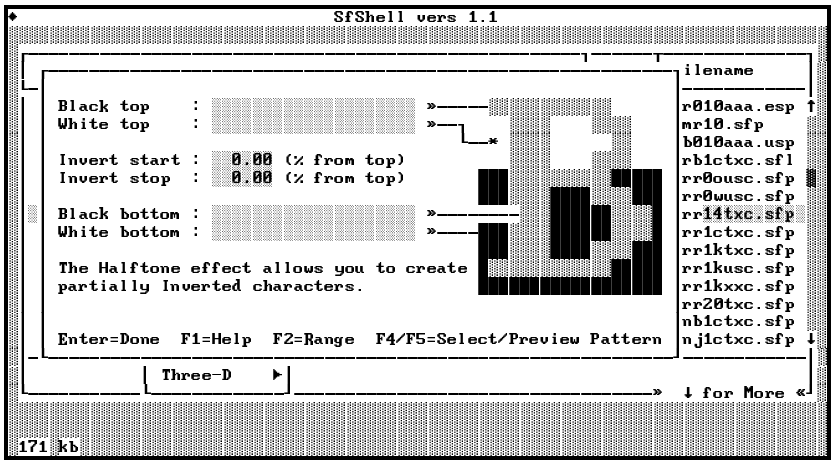


Figure 15.1. The Halftone panel

15.1. Options

Every character is divided into two areas, a selected area and a non-selected area. These areas are referred to as as the "top" area and the "bottom" area because that is the way they are drawn in the reference panel. 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). The reference panel displays a font that is being halftoned with the following parameters: the black top is using a pattern of 170;85, the white top is using a pattern of 0, the invert start is 50%, the invert stop is 100%, the black bottom is using a pattern of 0, and the white bottom is using a pattern of 170;85.

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

Black top

The black top pattern replaces the black areas of the non-selected region.

White top

The white top pattern replaces the white area (everything in the cell that isn't black) of the non-selected region.

16. The Horizontal Fade/Mist Effect

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

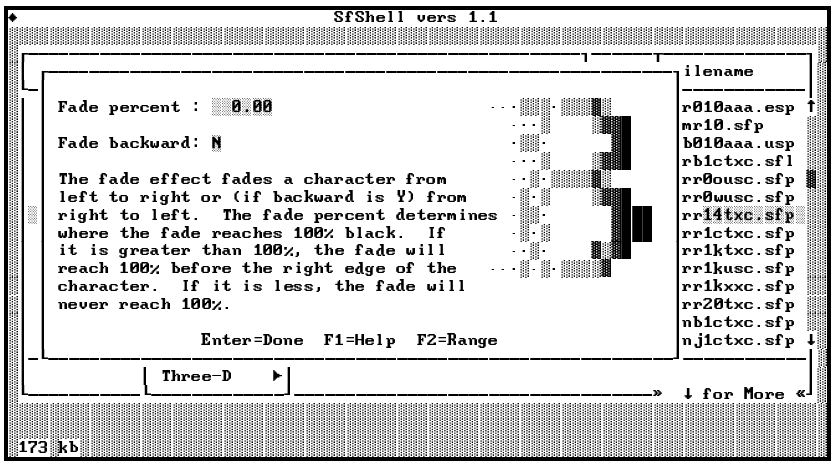


Figure 16.1. The Horizontal Fade panel

16.1. Options

Fade Percent

The fade percent 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. Conversly, fade factors above 100% draw the fade out so that it never reaches saturation.

Fade Backward

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.

16.2. 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*ColumnNumber*(FadePercent/100)/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.

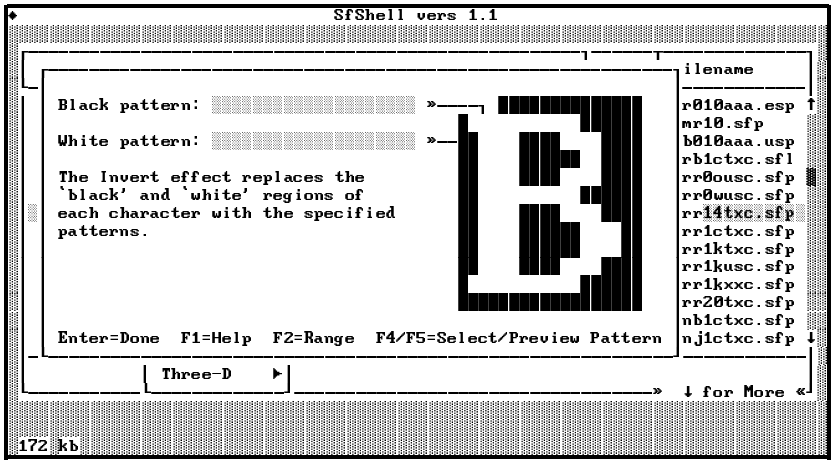


Figure 18.1. The Invert panel

18.1. Options

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

Black pattern

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

White pattern

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

18.2. Technically Speaking

Since the black and white patterns default to black and white, respectively, in practice, if neither the black nor the white pattern is specified, inverting has no effect.

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

20. The Mist Effect

Misting a font "smudges" each character.

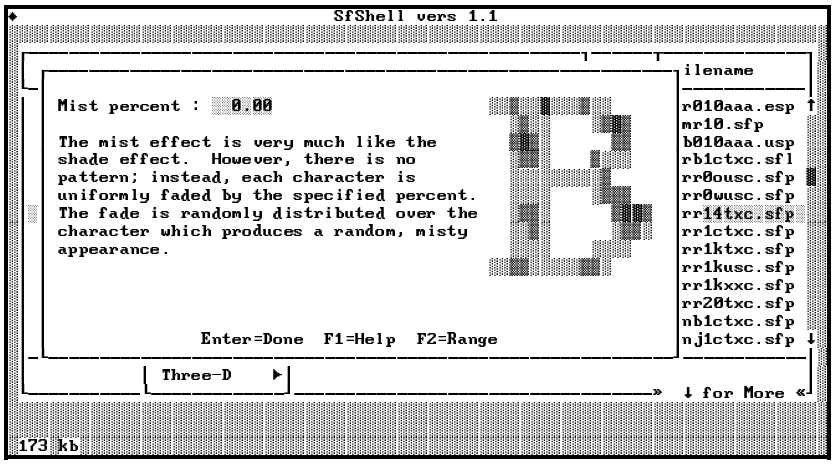


Figure 20.1. The Mist panel

20.1. Options

Mist Percent

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.

20.2. 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.

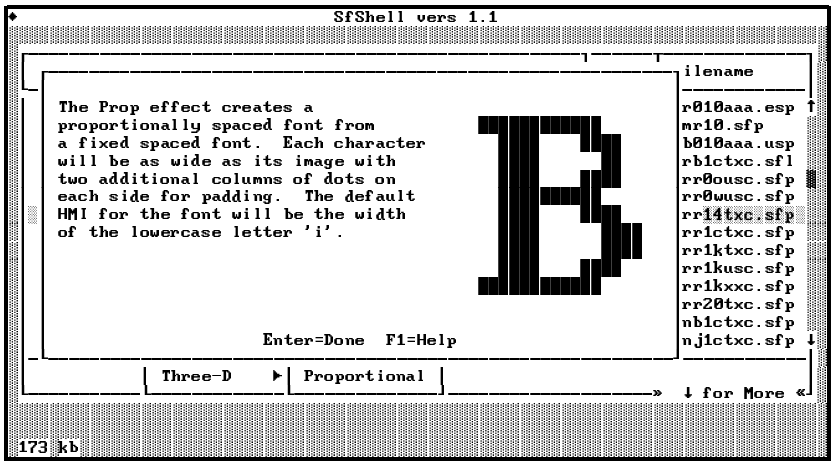


Figure 22.1. The Proportional Spacing panel

22.1. Options

There are no options for this effect.

22.2. 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.

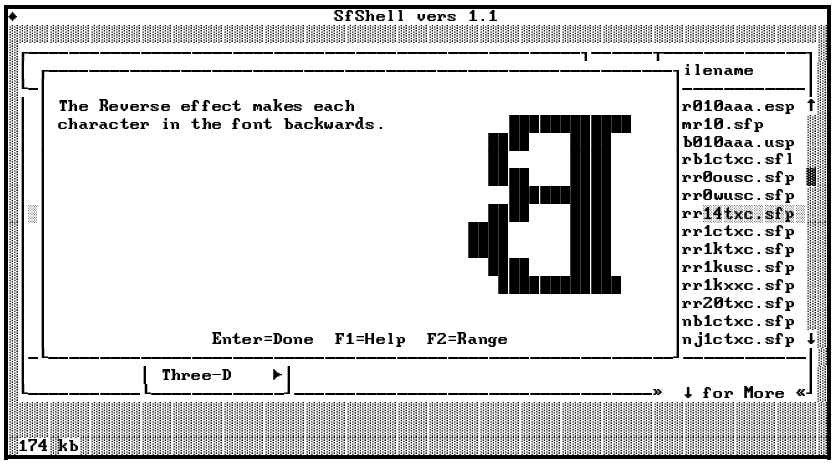


Figure 24.1. The Reverse panel

24.1. Options

There are no options for the reverse effect.

24.2. 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.

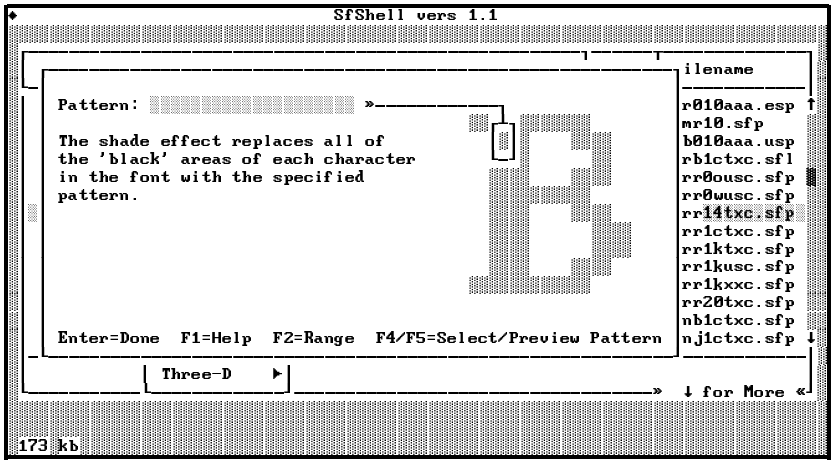


Figure 26.1. The Shade panel

26.1. Options

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.

26.2. 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.

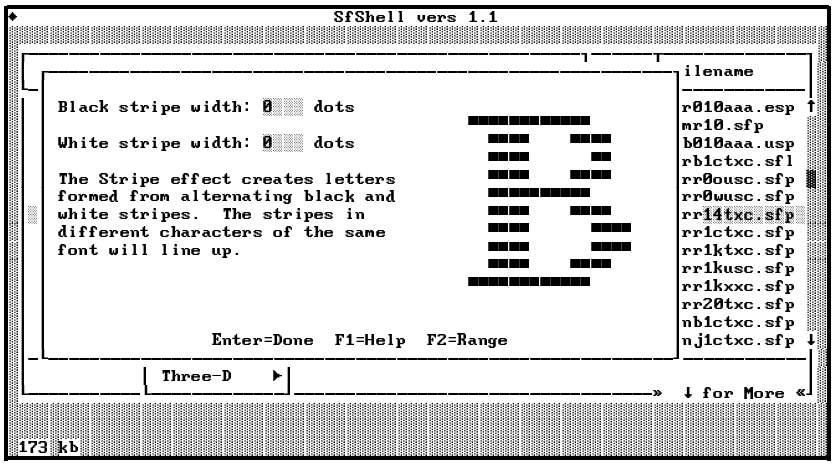


Figure 28.1. The Stripe panel

28.1. Options

Black stripe width

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

White stripe width

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

28.2. 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 Tilt Effect

Tilting creates characters that have a "tilted" baseline. Each character in the font can be rotated by an arbitrary angle. You cannot rotate a character by more than 90 degrees. The tilt effect in combination with rotating, mirroring, and reversing can produce a font that is effectively tilted by any angle.

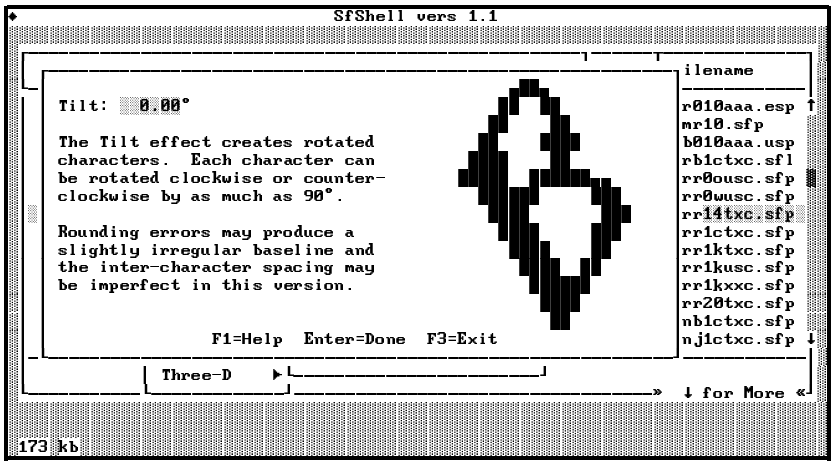


Figure 30.1. The Tilt panel

30.1. Options

Tilt

The tilt specifies the amount of tilt in degrees. A positive tilt value creates a font with a baseline that runs down and to the right. A negative value creates a font that runs up and to the right.

30.2. Technically Speaking

This effect is one of the most time consuming to run (on a 12pt font it may require evaluating three trigonometric functions, a square root, and several floating point and integer operations more than one and a half million times—and it's worse for bigger fonts). It creates a new character box large enough to hold the "tilted" original box and performs a trigonometric translation of every pixel into the new box. Although the effect makes some small changes to the top offset and left offset values for each character (in an attempt to correct for translation errors inherent in translating from one square grid system to another), the character dimensions are basically unchanged.

This effect creates a rotated font, but you still need a sufficiently flexible typesetting program to set the rotated text. Simply creating a rotated font will not, for example, cause your run-of-the-mill word processor to print it on an angle!

The horizontal spacing in a tilted font is sometimes imperfect. It is unclear why this is the case. Hopefully, a future version of Sfware will correct this problem.

32. The Filled Three-D Drop Shadow Effect

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

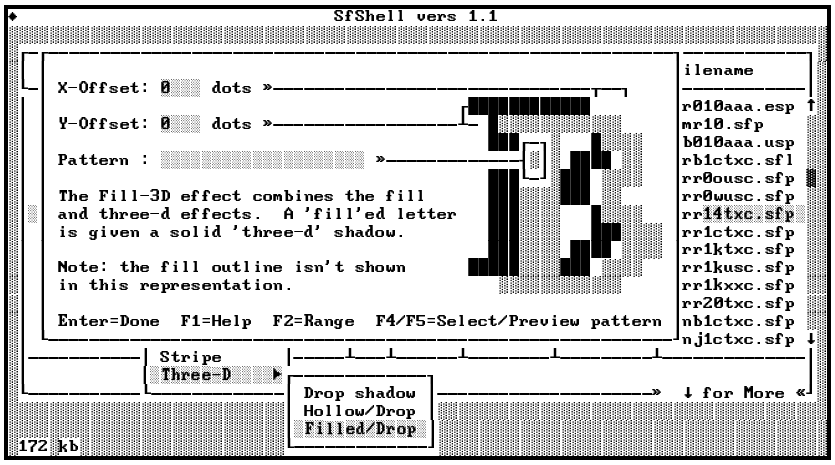


Figure 32.1. The Filled Three-D Drop Shadow panel

32.1. Options

X-Offset

The x-offset 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-offset, the y-offset controls the distance of the shadow above or below the character. Positive values create shadows below the character, negative values above.

Pattern

The pattern specified is applied to the original character.

32.2. Technically Speaking

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

34. Patterns

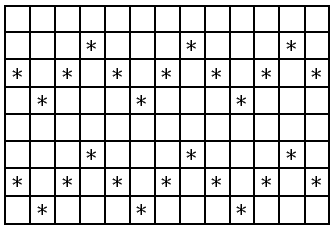
34.1. What are patterns?

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

34.2. How do I create one?

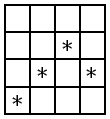
Creating a new pattern is not difficult. The best way to begin is with a piece of graph paper and a pencil. Experiment until you have something that you like and then follow the directions below.

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



Isolate a "generating region"

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



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

34.3. Saving the pattern

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

34.4. Previewing Patterns

It is possible to preview any pattern by pressing F5 when you are on a pattern field or when the list of patterns is displayed. The list of patterns will be displayed if you press F4 when you are on a pattern field.

34.5. Technically Speaking

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

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

`/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.

`/$`

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!

37. Contacting the Author

37.1. By Mail

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

Norman Walsh
#42I Southwood Apts
Brittany Manor Dr
Amherst, MA 01002

37.2. Electronically

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

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. If it is available, Sfware 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.

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

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

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

glyph A glyph is a more general term for a symbol that can appear in a font. Usually we refer to individual symbols in a font as characters (because they are things like "A" and "&"). However, since any arbitrary smear of ink can occur in a font, a more general term is sometimes used.

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

HMI HMI is an abbreviation for horizontal motion index. This is the width of the space that the LaserJet printer leaves when an undefined character is printed. Many font designers take advantage of this behavior by defining the HMI to be exactly the width of a space and not defining a space character in the font.

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

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

mask See *filemask*.

memory, expanded See *EMS*.