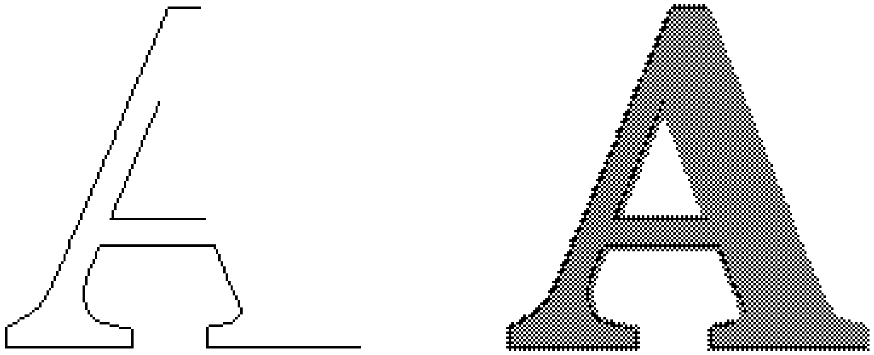
6. Refresh



The display area may be viewed as a background overlaid with a transparency on which curves are drawn. The background picture is a "one bit per point bitmap" where dark areas are represented as gray halftone.

6.1 Refresh:

Because of the particular way in which curves are drawn and deleted, the display area may get dirtied in regions where curves cross or overlap each other, and where knots coincide. Therefore a command is provided for refreshing the display area. This is a reasonably fast operation which regenerates the background and produces a clean display of spline curves without knots. The current selected section, if any, disappears. The refresh command comes in two flavors: with a clear background or with the current background.

6.2 Shift:

This is a refresh combined with a translation of all the curves. The translation is specified as for a move command: source point and destination point. The background, if displayed, is not translated.

6.3 New background:

In order to obtain a new background, a character dot matrix may be read from a file in CU format. This character matrix will be expanded so as to fill a maximum area of the display, and the character will be displayed in gray halftone. The expansion factor is the same for all the characters in the same CU file, as it is determined by the constant height of the matrix and the width of the widest character. The interaction scenario is as follows: type the name of the CU file which will cause the file to be scanned for its content (be patient); alternatively, if the same CU file is used as before, only type escape, since the file does not need to be scanned again; then the list of the characters the CU file contains is displayed; now type the desired character (or type escape followed by the octal code).

7. File input/ouput and plotting

7.1 Read and Write:

Two commands permit reading and writing the displayed splines, without concern for whether these splines form a well-defined character outline. Arbitrary sets of splines may thus be stored and retrieved. This is the same file format as used by the illustrator program DRAW[1] (the recommended file name extension is DRAW). When reading pictures generated by DRAW, text and curve brushes are ignored.

7.2 Plot:

Plotting of the picture is done using the PRESS file format. The command plot outputs the picture as a bitmap in PRESS format.

The file may be printed on EARS through MAXC; for this you may use the command file PRINT.CM which FRED generates. The file may also be used by programs accepting PRESS files: for instance, MARKUP[2] may be used for inserting the bitmap picture into a document.

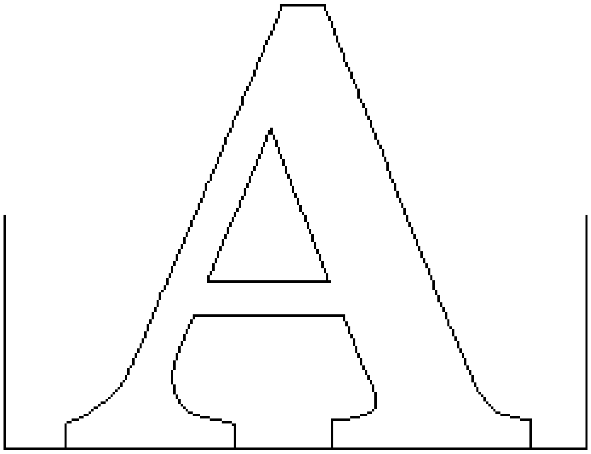
----

[1] Documentation on <GR-DOCS>DRAW.EARS

[2] Documentation on <ALTODOCS>MARKUP.EARS

8. Making a font

The main intended use of FRED is for making fonts, or more precisely creating spline outlines of characters.[1] Spline characters are generated using the curve editing features of FRED (described in section 3 and 4). Additional commands are provided for storing in a file and retrieving from a file such a character description, as well as for specifying the additional information necessary for fully defining the font. These commands are available from a submenu which scrolls in when the command font is invoked.



Section 8.1 first describes the various elements composing a spline font description. Then section 8.2 explains the various commands and methods for creating and modifying these components. Section 8.3 presents the file input/output commands.

Generation of the appropriate fonts for various devices, using spline fonts, is done with the program PREPRESS.[2]

8.1 Description of a spline font:

A special LISP-compatible text format is used for spline fonts (given in appendix, for the very curious). The recommended extension for such a file is SF. A spline font description contains the following components for each character:

a) character outline: it is composed of a number of closed curves made of a number of end-to-end connected splines.

----

[1] R.F. Sproull, "Fonts project", September 9, 1974.

[2] Documentation on <GR-DOCS>PREPRESS.BRAVO

b) base line and width: or more precisely, the position relative to the outline of the character of the horizontal base line, the left side of the character slug and the right side of the character slug.

c) fiducials: the spline outlines generated by FRED are intended to be used by the program PREPRESS which "scan-converts" the character, i.e. generates the actual dot matrix used on a printing or displaying device. The actual resolution of the dot matrix will be a function of the resolution of the device (for instance 500 lines/inch) and the desired point size of the displayed or printed character (say 12 points).[1] In order to guarantee that the scan-converting process will produce an appropriately scaled dot matrix font from a given spline font, there must be some means to relate the particular coordinate system used for the spline outline to the size of the final dot matrix. For that purpose, each character definition contains a set of two numbers called fiducials. These two numbers are respectively equal to the height and width in the coordinate system of the spline outline of a square whose side is equal to the point size of the character. These numbers are used to determine the scale factor to apply both verticaly and horizontaly to the spline coordinates for producing a dot matrix for a particular point size.

d) character identification:

family: e.g. Baskerville;

character: e.g. "A", or octal ASCII code 101;

face (or style), which has three components: bold or medium or light, regular or italic, condensed or regular or expanded (defaulted to medium, regular, regular).

e) bookkeeping information: version number, creation date, and name of file used for background.

8.2 How to create a spline font:

FRED can define all the components of a spline font with a number of special purpose commands.

a) character outline: in general practice, this outline is generated by creating and editing splines to follow the contours of a halftone character displayed as a background (section 5.2). There are two typical cases. The background character could be obtained from an existing font (in dot matrix format) for a device such as Alto, VTS or SLOT, which one wants to convert to the more general spline font format. Alternatively (and the most likely in the future), one could create an original font in spline outline format. For this purpose one would first create a digitized picture

----

[1] The point is a unit of type measurement equal to 1/72 inch (vive le systeme metrique...).

of the type font to use as the background. In either case, CU file format is the standard, since it is the format used by the video font digitizing system. The recommended resolution for digitized type font pictures is 256 by 256; this creates rather large files but provides a background with minimally jagged contours which are easier to fit with spline curves.

b) base line and width: current base line and width may be modified or redefined in only two ways: with the command base & width, or by reading a character definition from a spline font file. The command base & width actually activates a special mode for defining an arbitrary rectangle in the display area (which is also used for defining fiducials). A submenu scrolls in, with the following commands:

left and right: when that mode is activated, switch 1 is used for defining the left side of the rectangle, switch 2 the right side. Switch 3 is unused.

top and bottom: when that mode is activated, switch 1 is used for defining the top side of the rectangle, switch 2 the bottom side. Switch 3 is unused.

move: use any switch to reposition the bottom left corner of the rectangle, its dimension unchanged.

height & width: input at the keyboard the desired dimensions of the rectangle (in screen units), the bottom left corner remaining fixed.

ok: terminate, i.e. return to font command.

When the command base & width enters the rectangle defining mode, a rectangle is displayed corresponding to the current values of base line and width. You may then modify base line and width by redefining the bottom, left and right side of this rectangle, moving the rectangle around (which affects only the base line) or eventually typing in the value of the width.

As an additional option, width (but not base line) may be automatically obtained from the CU font character currently used as a background. This is useful when converting an already existing font. The option comes in the form of a question when entering the command base & width.

c) fiducials: current fiducials may be modified or redefined in only two ways: with the command fiducials, or by reading a character definition from a spline font file. The command fiducials activates the same mode as the command width (for defining an arbitrary rectangle in the display area). It is described above (8.2 c).

When the command fiducials enters the rectangle defining mode, a rectangle is displayed corresponding to the current values of the fiducials. However only the dimensions of this rectangle (or square) are important. Its position on the screen are irrelevant. You may then modify the values of the fiducials by redefining the top, bottom, left and right side of this rectangle, or eventually by typing in the values.

As an additional option, fiducials may be automatically computed from the CU font character currently used as a background. This is useful when converting an already existing font to spline format. The option comes in the form of a question when entering the command fiducials. You must prepare for that option when reading a new CU file: answer yes to the question "Do you want FIDUCIALS automatically computed?"; then enter the point size of the font to be converted, and the resolution of the printing device (500 lines/inch for EARS fonts).

However, when creating an original font, the recommended practice is to digitize a picture of the font type also containing some marks or graduation indicative of the point size of the font. These marks will appear on the screen as part as the background, and the fiducials will be defined by pointing at them.

d) character identification and bookkeeping information are defined or modified through a command labelled miscell which provides some self-explanatory keyboard interaction.

8.3 Reading and writing spline font files

One SF file may be opened at a time, for reading, writing or both. Opening a file, or creating a new file, is done with the command get. The filename must have extention SF. Getting a font file (say FOO.SF) may take some time if it contains many characters, as it implies scanning the file and duplicating it under the name FOO.XF. Beware that SF files grow fast: for efficiency, it is recommended not to store much more than a dozen characters into one single SF file. When quiting or when getting another SF file, the previously opened SF file is closed. Confirmation is expected before closing the file. Confirming with a V (for verify) allows selective deletion of unwanted characters from the file being closed. After file FOO.SF has been closed, FOO.XF will be a copy of the initial file FOO.SF.

Do not exit from FRED by any means other than quit. There are ways to recover from the effect of a crash or other similar disruption, but they require expertise.

Characters may be randomly read from, or written on the currently opened SF file. Specify a character by typing either a single key, escape followed by octal code, or return to abort. Overwriting a previously stored character requires confirmation. The readcharacter command displays a character directory of the opened file.

The command defineandwrite differs from writechar in that it automatically goes through the commands base, width, fiducial and miscell before proceeding to write out the font definition. When writing out the font outline, all splines not forming a closed curve will be ignored. This means that auxilliary curves created as templates or used as constructive elements, that is to say not actually part of a character outline, do not have to be deleted at the time of writing.

9. Keyboard commands

Command input may be done on the keyboard (as well as from the menu) for most operations at the top level. This allows faster interaction for the experienced user.

The key corresponding to a command is simply the first letter of that command: e.g. key command U is equivalent to menu command undo. There are only a few exceptions:

- repeat is escape;

- keys M and C are used to set the meaning of keys T, V and H to be either a move or a copy operation (translate, vertical symetry, horizontal symetry);

- mainly for safety reasons, wipe is done with <control>W;

- background and refresh operations also use control keys:

<control>B	refresh with background
<control>C	refresh with clear background
<control>N	new background

- in addition, delete and backspace are used to suppress an unwanted selection.

10. Getting started

Obtain the file <GRAPHICS>FRED.DM and LOAD it. It contains the following files:

- the program files: FRED, FREDOV1.BB to FREDOV5.BB;

- the menu picture files: MENU1.FRED to MENU4.FRED;

- a utility program SFMUNCH for processing spline font files (described in Appendix A).

Acknowlegments

This document greatly benefited from help and suggestions by Bill Bowman and Bob Sproull.

Appendix A

SFMUNCH

This is a utility program for processing spline fonts: concatenation of SF files, setting fiducials, and character transformations (shearing-for italics-, condensing and expanding). The syntax of the command is as follows:

```
SFMUNCH   <output SF file>   <operations>   <list of input SF files>
```

The available operations are:

i/I incline characters by the specified slope percentage i;

e/E expand characters by the specified percentage e;

c/C condense characters by the specified percentage c;

xf/X set x fiducials to the given value xf;

yf/Y set y fiducials to the given value yf.

If no operation is specified, a simple concatenation of the SF files is done. Transformation specifications may be mixed with the list of input files. They take effect only for the input files following them.

In addition, when /V is used, confirmation is expected before processing and writing out each character.

Examples:

```
        SFMUNCH METEOR.SF METEOR*.SF
```

concatenates all METEOR characters into a single file;

```
        SFMUNCH/V METEOR.SF METEOR*.SF

or      SFMUNCH METEOR.SF/V METEOR*.SF
```

selectively concatenates METEOR characters into a single file;

```
        SFMUNCH METEORI.SF 10/I METEOR.SF
```

generates a font file of pseudo-italics (10 per cent incline);

```
        SFMUNCH NUMSYM.SF SYMBOLS.SF 15/E NUMERALS.SF
```

generates a font file of symbols and expanded numerals.

Appendix B

Font file format

The following description uses the notation:

```
<...>   is a list,
{...} is a string,
[...]   is a number.
```

A spline font file has the form:

```
        <character description> ... <character description> STOP
```

where <character description> is either of the form:

```
        ((FAMILY  {family name})
         (CHARACTER  [code])
         (FACE   { B | M | R }   { R | I }   { C | R | E })
         (WIDTH  [width in x]  [width in y])
         (FIDUCIAL  [dimension in x]  [dimension in y])
         (VERSION  [number]  {date})
         (MADE-FROM  {file name}
                [x character origin]  [y character origin]
                [x fiducial origin]   [y fiducial origin])
         (SPLINES  <closed curve> ... <closed curve>))
```

or of the form:

```
        ((FAMILY  {family name})
         (CHARACTER  [code])
         (USE {family name}  [code]
                { B | M | R }   { R | I }   { C | R | E }))
```

where <closed-curve> is:

```
        (<spline> ... <spline>)
```

where <spline> is:

```
        ([n] <knot list> <weight list> <derivative list> {solution method})
```

where [n] is the number of knots,
and <knot list> is:

(([X_1] [Y_1]) ([X_2] [Y_2]) ... ([X_n] [Y_n]))

and <weight list> is:

([W_1] [W_2] ... [W_n])

and <derivative list> is:

(([X_1'] [Y_1'] [X_1''] [Y_1''] [X_1'''] [Y_1''']) ...
... ([X_{n-1}'] [Y_{n-1}'] [X_{n-1}''] [Y_{n-1}''] [X_{n-1}'''] [Y_{n-1}''']))

and {solution method} is:

```
        { NATURAL | CYCLIC | PSEUDO-CYCLIC }
```

Comments of the form:

```
        (COMMENT {any string})
```

may be inserted in a <character description>.

FACE information stands for:

```
        BOLD | MEDIUM | LIGHT
        REGULAR | ITALIC
        CONDENSED | REGULAR | EXPANDED
```