Inter-Office Memorandum

To	Sil Users	Date	August 1, 1979
From	Roger Bates	Location	Palo Alto
Subject	Sil file definition	Organization	CSL

XEROX

Filed on: [MAXC]<Sil>SilFileDescription.press

This is a description of the SIL file structure. The definition of the Sil file format was made solely for the convenience of the Sil program, and is simply a copy of the internal data structure that Sil maintains in memory during operation.

The first word in the file is a check code, that can be used to see that the file is actually a Sil format file. This word has two possible values, octal 34562 and 34563. The second value has been adopted as part of the design automation process to indicate the a Sil schematic file has been "build" into a wire-list file via the Build process.

Following this word are a secession of "item blocks" that describe the contents of the Sil drawing. An Item block has the following form:

word 0: macro-code

If this word equals a character code, then this item is to be added to the list of items for the macro Font 4 character. All items which are part of macro definitions will appear first in the file.

If this code is -1, then this is a normal item (string, line etc) and is to be displayed once as indicated by the following words.

word 1: state and X-min

bits 0,,3 are used internally to indicate the current state of an item - normal, selected, or deleted.

bits 4,,15 are the minimum X coordinate bit position of the bounding box that the item occupies.

word 2: Y-min

bits 0,,15 are the minimum Y coordinate bit position of the bounding box that the item occupies.

word 3: color and X-max

bits 0,,3 are used to indicate the color associated with this item according to the following table:

00: Neutral
01: Red
02: Yellow
03: Green
04: Cyan
05: Violet
06: Magenta

08: Brown
09: Orange
10: Lime
11: Turquise
12: Aqua
13: UltraViolet
14: Pink

07: White	15: Smoke

bits 4,,15 are the maximum X coordinate bit position of the bounding box that the item occupies.

word 4: font, face, and Y-max

bits 0,,3 are used to indicate the "font" associated with this item. Font values are interpreted according to the following table:

00: Font 0
01: Font 0 with "boldness" opposite to default value in user.cm
02: Font 1
03: Font 1 with "boldness" opposite to default value in user.cm
04: Font 2
05: Font 2 with "boldness" opposite to default value in user.cm
06: Font 3
07: Font 3 with "boldness" opposite to default value in user.cm
08: Font 4 macro
09: Font 5 macro
10: Font 6 macro
11: Font 7 macro
12: Font 8 macro
13: Font 9 macro
14: A line with lingth and width indicated by preceeding coordinates
15: A background with area indicated by preceeding coordinates

bit 4 is used to indicate the "italics" opposite to default value in user.cm. This bit only has significance if the font value is 0 through 7.

bits 5,,15 are the maximum Y coordinate bit position of the bounding box that the item occupies.

word 5 etc: optional string

If the preceeding word indicated an item of fonts 0 through 13, then additional words are required to define the characters in the item string. This is done with a BCPL string definition, that is, the left byte of word 5 is the number of characters in the string, the right byte is the first character. Additional words are appended as needed to contain the specified number of character bytes.

All coordinate value are expressed in "Alto Screen" units and are refernece to the upper left hand corner of the screen. When making a press file, Sil scales all coordinate values by 17780/500.

Notice that descriptions of library fonts 5 through 9 characters are not defined within this Sil file. A table is filled in by Sil during input, and later the library files indicated in user.cm are opened, and the front of the file is scanned for items defining the required character codes for each of the library fonts characters.

Editorial comment

Notice that in the preceeding definition, there are several instances of references to the contents of the user.cm to get font family name, default faces, and library file names. This means that a signifacant part of a Sil picture is not defined by the Sil file at all. Users of Sil may come back to an old file that they carefully saved away only to find that they really "lost" part of the drawing because they change some of the parameters in their user.cm file for new applications.

This has even more impact for people interested in processing Sil files by other programs, since they must either work around the parameters in user.cm, or deal with user.cm

> When one runs Sil/I, Sil reads user.cm for font information, and file pointers, and saves this information in two files called Sil.fonts, and Sil.fps (file pointers). The proceedures for reading these two scratch files could easily be lifted from the Sil sources [MAXC]<SIL>SilSources.dm.

Programs should in fact only use user.cm to fill in default values when first needed. All information should be explicidly saved as part of the output file, so that no future refereneces should be dependant of the contents of user.cm