Inter-Office Memorandum

To	Whom it May Concern	Date	July 13, 1981
From	Lyle Ramshaw	Location	Palo Alto
Subject	A Font Sampler	Organization	CSL

XEROX

Filed in many little pieces on:
[Ivy]<Fonts>CloverCharacters*.press

This document describes the fonts that are available on the PARC/CSL Dover printer named Clover, and on some other printers in PARC and SDD as well. It is a companion to a much shorter document entitled CloverFonts.Press''. In CloverFonts.Press'', Clover's dictionary is described by a table that lists all of the fonts by family name, size, and face: consult it when you need to know what large sizes of TimesRoman are available, for example. This document covers only a sample of Clover's fonts, one size and face of most families, but it shows you what each of the characters in those fonts actually looks like.

A font associates graphic symbols with identifying numbers called *character codes*. If you stick to the vanilla characters, you can often get by without knowing about character codes, since the graphic symbol that you want can be obtained simply by pressing the corresponding key on the keyboard. However, there are many more character codes than there are keys on the keyboard. The extra codes are used for more exotic graphic symbols, and for various control functions. To use an exotic graphic symbol, you have to convince your editor or illustrator to employ one of these exotic codes. There are many programs that use fonts, and almost as many different conventions for access to the exotic codes. The first few pages of this document contain a few hints on how to type exotic codes to some of our standard programs: this is the issue of character code *input*. But most of this document deals with the issue of character code *output:* what graphic symbol, if any, will result from printing a particular code in a particular font.

The output section consists of a collection of tables, one per font, ordered alphabetically by the font's family name. The cells in each table are labelled by character codes, and in each cell is a single instance of the corresponding character, along with some reference tick marks. The two horizontal tick marks show the baseline of the character, while the two vertical tick marks indicate the width of the character. Adjacent characters in a string are printed with their baselines aligned, and so that the right width tick of the first character is coincident with the left width tick of the second. Note that the left and right vertical tick marks of a zero-width character are coincident. (Wizards: The left ticks, if extended, would intersect at the character origin while the right ticks would meet at the end of the width vector. The data line indicated is the centerline of the tick.)

In this document, character codes will be written as three digit octal numbers preceded by a pound sign (#). Most fonts only associate graphic symbols with some subset of all possible codes; and most programs only let their users employ some subset of all possible codes. In particular, many fonts and many programs only handle the *seven bit codes:* #000 through #177. The remaining codes, #200 through #377, are known as the *eight bit codes*.

The title page of this document shows the names normally used to refer to the seven bit character codes at PARC. The codes from #000 through #037 are known as *control codes*, and the prefixed up-arrow in the names of these codes is read control''.

CloverCharacters.Press

Character code input:

Here are some comments on typing exotic character codes to some of the standard programs that deal with fonts. Following this text are two pages that show in tabular form how to type each seven bit character code to the keyboard handler in the Alto OS and to Bravo 7 respectively.

Alto Operating System: Only seven bit codes. Control characters can be typed by using the key labeled ctrl'' as a shift key. In addition, <ctrl><space> gives #000 and <shift><lf> gives #140.

Bravo 7: Only seven bit codes. You can get the alphabet and the vanilla punctuation marks by typing their keys in the standard way: this accounts for all codes above #040 except for #140 and DEL=#177. Of these, Bravo lets you get at one out of two: the code #140 can be typed as <shift><hyphen>, while DEL is not available. To input a control code, first type the non-control version in either upper or lower case, and then type a <ctrl><S>. That is, ^S=#023 acts as a postfix operator to change the preceding character into the corresponding control character. This ^S feature allows you to insert any control character into your document. But four of these characters have special interpretations when present in a Bravo source file, and hence cannot be used to represent graphic symbols. At the end of each paragraph in a Bravo source file is a special sequence of characters beginning with a ^Z=#032, and only these ^Z's are permitted in a Bravo source file. If you insert any other ^Z's, Bravo will not be able to Get the resulting file. The other three stolen copntrol codes have special interpretations that relate to carriage control: ^I=#011 is tab, ^L=#014 forces a page break, and ^M=#015 is carriage return.

Using control characters from Bravo is more pleasant if Bravo can be convinced to paint the appropriate graphic symbols on the display screen as well. There are two principal ingredients of success in this quest: updated Alto fonts and Graphic mode. You first need to retrieve Alto (.AL) fonts that contain the appropriate graphic symbols in the control slots: the directory [Ivy]<Altofonts> is a good place to look. After retrieving these new Alto fonts, of course, you must reinitialize Bravo. Now you are ready to learn the mysteries of Graphic mode. Every character in a Bravo document is either Look Graphic or Look unGraphic, and unGraphic is usually the default. If a control character has the unGraphic attribute, then Bravo does not look in the control slot of the Alto font for a symbol to paint, but instead paints the corresponding lower case letter with an overbar. On the other hand, control characters with the Graphic attribute are displayed with the appropriate symbol from the control slot in the Alto font, if any, or else with a black blob. If you use control characters, all of your text should probably be Look Graphic. You can make this a little easier by adding the line LOOKS: g'' to your user.cm file, and reinitializing Bravo; this will make Grahic mode the default, instead of unGraphic mode.

Draw: Only seven bit codes. The control codes are not accessible at all.

SIL: Only seven bit codes. Control codes are handled by a postfix ^S as in Bravo.

TEX: The TEX on Maxc only deals with seven bit codes, although a hack is provided for accessing the upper half of a font as if it were a separate font. When TEX is taking input from a file, control codes in the file get through to TEX without any trouble. If you are typing directly to TEX on Maxc, then some control characters such as ^C don't get through, since they are given a special interpretation by TENEX. But a ^V can be used as a prefix quoting character, to force the next character of the typein to be taken literally. More or less the same thing is true when typing directly to the Mesa implementation of TEX: some control characters have an associated editing function (^A, ^H, ^Q, ^R, ^W, and ^X), but ^V can be used to quote the following character.

Remember that, when worst comes to worst, TEX allows you to talk about character codes explicitly: the \char'' control sequence allows you to specify a character by giving its octal code.