Sonata Font Design Specification

Adobe Developer Support

Technical Note #5045

31 March 1992

Adobe Systems Incorporated

Corporate Headquarters
1585 Charleston Road PO Box 7900
Mountain View, CA 94039-7900
(415) 961-4400 Main Number
(415) 961-4111 Developer Support
Fax: (415) 961-3769

Adobe Systems Europe B.V.
Europlaza
Hoogoorddreef 54a
1101 BE Amsterdam Z-O, Netherlands
+31-20-6511 200
Fax: +31-20-6511 300

Adobe Systems Eastern Region
24 New England
Executive Park
Burlington, MA 01803
(617) 273-2120
Fax: (617) 273-2336

Adobe Systems Japan
Swiss Bank House 7F
4-1-8 Toranomon, Minato-ku
Tokyo 105, Japan
81-3-3437-8950
Fax: 81-3-3437-8968

PN LPS5045

Contents

Sonata Font Design Specification

1 Introduction

The Sonata™ music notation font was designed to be easy for a PostScript™ language developer to work with. Many decisions on the sizes and orientations of glyphs were made specifically with software developers in mind. This document outlines some of the aspects of the font that influence its setting in various environments. To help software developers use the font to the fullest and in accord with its design, this documents supplements other documentation on the Sonata font and provides some of the philosophy that informs the design of the font.

2 Character Widths

Character widths in ordinary text fonts primarily govern the inter-letter spacing of the text. The width of a character includes the “white space” on each side of the character (the side bearings). In the PostScript language, the character width determines where the current point is located after printing a character.

When printing music, rarely are characters placed one after another on a line, as in Roman text. In the PostScript interpreter, the character widths are used solely to control the location of the current point after printing a character. For many symbols in of the Sonata character set, their widths are exactly the left-to-right width of the symbols themselves. In other words, most characters have zero side bearings. There are several deviations from this.

In situations where the subsequent location has some importance, the character width of a symbol is designed to meet the needs of setting that symbol. For instance, the flag and stem characters are normally used in conjunction with other symbols (such as notes or note heads), and the registration of the symbols is extremely important. The widths of these characters are zero in the Sonata font, so the current point will not move when the next symbol is set “on top” of it. See the discussion of composite symbols later in this document for a more thorough examination of this process.

3 Origins or Reference Points

Each glyph in a PostScript language font has an origin, or the reference point, from which the character is painted. In the typical sequence of moving to a particular location on a page and printing a font symbol, the point that was originally the current point becomes the origin of the character as it is painted.

The origins of the characters in the Sonata font are designed to minimize the effects of round-off when placing symbols. Characters normally set in close registration with one another are designed to require minimal repositioning to set them. For instance, any symbol (or note) that is normally set on the staff has its origin centered about its horizontal axis.

To set most symbols on a staff line, one need only move to the exact location of the particular line or space on the staff, and print the appropriate symbol. No complicated calculation is necessary to accurately locate these symbols on the staff. Most of the symbols have their origins at the left edge of the character (or in line with the left edge). This decision was made in part to accommodate screen display issues where font display required the origin of the character to be at the left of the character. The rare exceptions are the composite characters.

The staff character has its origin at the left edge of the shape, in the center of the bottom staff line. Although this bottom line may be of infinitesimal thickness at most point sizes, it is important to realize that the origin is in the center of the line, and compensation for line thickness is not necessary for placement.

4 Size Issues

Because Sonata is a font, it typically operates in an environment set up primarily for text. In most applications, this means that the notion of size is carried in the point size of the font. This is based on the printer’s point, or 1/72 of an inch. In music, this measure does not pertain to anything, except perhaps accompanying text.

The Sonata font is set up so that the point size carries some meaning. In particular, the point size of the font is exactly the distance between the topmost staff line and the bottom staff line of the staff character. This measurement is center to center, which means that no adjustment is necessary to allow for the thickness of the staff lines themselves. All other symbols in the font are designed proportionally to be the correct size. Note that the entire font can be scaled to any size, and all the symbols will be scaled proportionately.

The relationship between the staff height and the point size is important when setting music. This allows one to easily determine the distance between staff lines, without extensive computation. The center-to-center distance between staff lines is always one fourth the point size of the font. This is an exact measure, and placement of notes might depend on it.

If the font is scaled to 24 points, the distance between the first and second staff lines is exactly 6 points. This allows software to easily set up a coordinate system based on the staff character that can carry through all setting operations. If the staff is not drawn using the staff character, rather the PostScript language line-drawing primitives are used, it is important to set up the staff at an appropriate size for the symbols being used. The rule of thumb given above for sizing the staff should be followed.

5 Character Set

There are many symbols in the Sonata character set that might not be needed by an application. Many are designed specifically to help solve a difficult placement or setting problem. For instance, there are several “backward” note shapes, with the note head on the opposite side of the stem than expected. These are designed to work in building chords that must share the same stem, without alignment problems occurring from having to do **moveto** operations back and forth. They are designed to be able to maintain a fixed X location and only move upward or downward when placing notes for a chord.

Similarly, there are two different space characters, with different widths. One, stemspace, is exactly the width of a note head less the width of a stem. This can be used in special situations where the current point must be left in exactly the right spot for printing a subsequent stem or flag character. There are also two note heads, a quarternotehead and a noteheadextra, that appear to be identical. In fact, they are identical from the PostScript interpreter’s point of view, but they are differentiated so that in screen representations an application can customize the look of an upside-down composite note without affecting its placement. In other words, the printer-font characters are the same and will print identically, but the screen font characters might differ slightly for display reasons.

6 Encoding

The encoding for the Sonata font is designed to be as mnemonic as possible. The encoding refers to the indexing of symbols in the font by byte code, or ASCII code. This is indirectly tied to keyboard mappings, which generate byte codes, but is also important to a music-setting application, which does not even use the keyboard.

The encoding for Sonata is intended to be easy to use for a person typing. The encoding is based on the Apple® Macintosh® keyboard layout, that is, that related characters are grouped by keys on the Macintosh keyboard. The symbols are typically either visually related to the key to which they are associated, or they are related mnemonically through the actual letter on the keycap. Related characters are typically grouped on the same key and are accessed by using the *shift, option,* and *command* keys to get related characters.

For instance, the q key is associated with the quarternoteup glyph, Q (or shift-q) with the quarternotedown character, option-q with the quarternotehead character, and so on. The treble clef is located on the ampersand key (&), because resembles an ampersand.

In general, the shift key flips a character upside down, if that makes sense for a given character, and the option key selects the note head equivalent of a note. There are many instances where this is not possible or practical, but there is a philosophy in its design that will become evident and will allow a user to easily remember the location of most of the characters in the font.

7 Composite Symbols

The Sonata font contains composite characters that may be used to build up symbols from pieces, rather than imaging an entire symbol at once. There are standard notes up through 128th notes in the font, for instance, but there is also a separate note head, stem, and two kinds of flags for building these notes as composites. This is done to accommodate as many styles of setting music as possible. If the internal data representation you work with builds up symbols piece by piece, you might find the composite characters to be very helpful. Otherwise, you can just use the 128th note, or the 64th rest, or whatever you need from the standard selection of characters.

The composite symbols are special in many ways, because their only use is to be set in conjunction with other symbols. They all have character widths of zero, so the current point will not move while you build a symbol. When adding flags to a note, for instance, you might want to move vertically to build many flags, but you needn’t move horizontally.

If a composite symbol is being set, it is best to set the flags, stems, or other composite parts first (because the current point will not be changed as the characters are painted), followed by the note or note head or other standard symbol upon which the composites can be placed. There are four different stems, six flags, and a host of backward notes for chords.

Note that the eighth and sixteenth rests have widths that are narrower than the actual width of the symbols. This is designed to allow stacking rests without tedious placement. If you move vertically after placing one rest, the current point will be just right for stacking another rest on top of the existing one.

Appendix: Changes Since Earlier Versions

Changes since May 4, 1991

- Document was reformatted in the new document layout and minor editorial changes were made.

Changes since May 19, 1989 version

- A few layout styles were changed, trademark information was cleaned up, and a few other minor modifications were made.
- Updated addresses on cover page.

Index