SIL ViewGlyph — Font Viewing Program

Author: Bob Hallissy, NRSI
Version: 1.81
Date: 2009-10-01

What is SIL ViewGlyph?

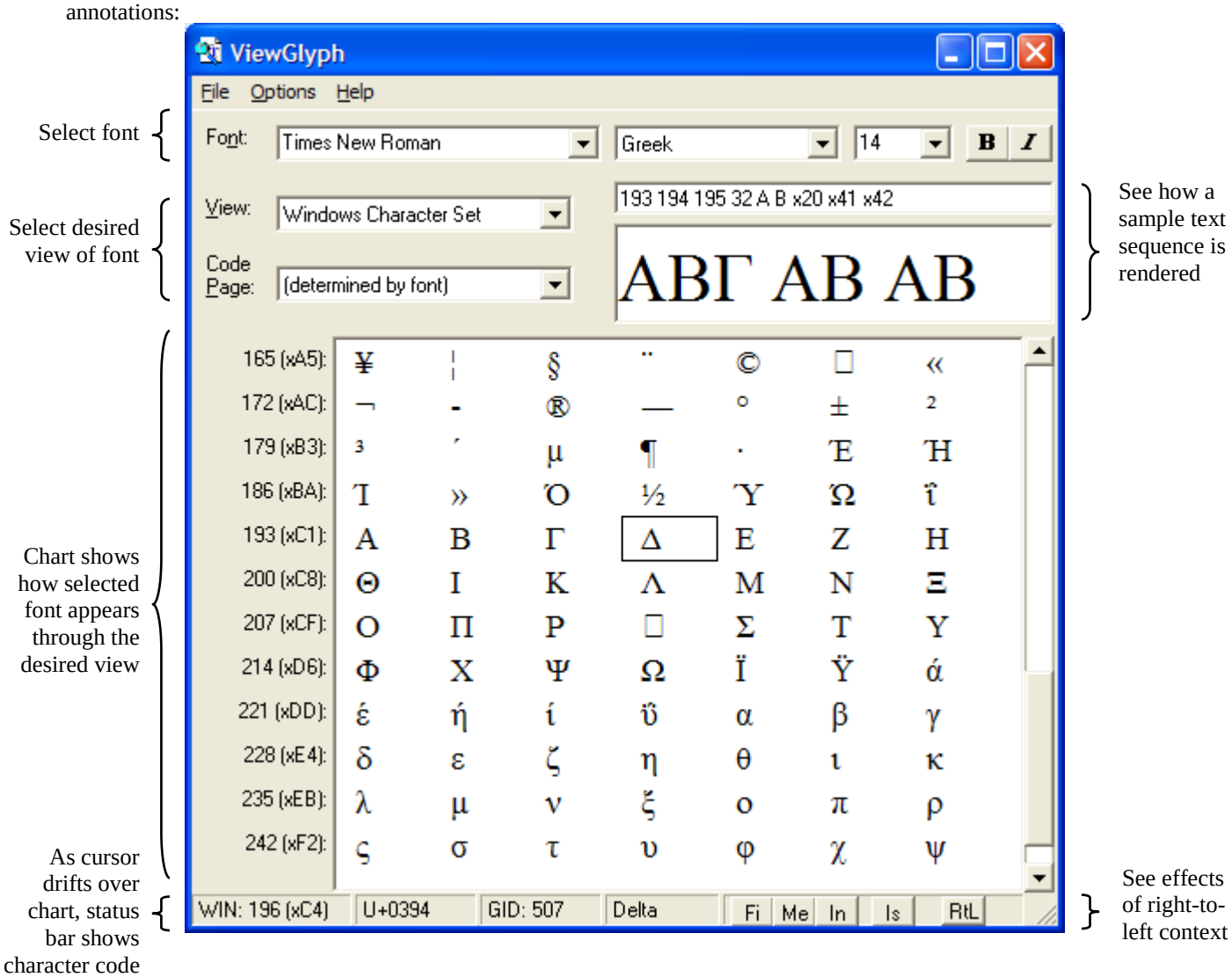
SIL ViewGlyph[1] is a Windows® 32-bit utility program I wrote in order to experiment with Windows text API calls, Unicode, and to be able to look inside TrueType® fonts. There are lots of enhancements I'd like to make, but the tool serves some use as it is, and since others have found it useful, I offer it to you as is.

What does ViewGlyph allow me to do?

Among its most useful features, ViewGlyph allows you to *see* a font's contents through *different eyes*, so to speak. Want to know what the font looks like when used by a Unicode application? How about when the font is moved to a Macintosh? Or, suppose I have the multilingual extensions installed and want to know what a font looks like when viewed through a particular codepage? ViewGlyph can show you the raw glyph palette, which is useful if you are writing "smarts" (i.e. OpenType, Graphite, or AAT tables) for your font. Ever wonder what Windows does when you ask for a font that doesn't exist? Now you can find out what font Windows substitutes. You can view certain TrueType font tables ('name' and 'cmap') and see various font metrics. Finally, ViewGlyph is useful for investigating how Windows maps 8-bit characters into Unicode through various codepages.

Where is the Help file?

Sorry, there is no help file yet. Someday. In the mean time, here is a snapshot of the main window with some annotations:



[1] SIL ViewGlyph is not in any way related to Roman Pivovaroff's ViewGlyph program

System requirements

ViewGlyph works best on Windows 2000 or later, but it will work (with some reduced functionality) on anything from Windows 95 forward. The primary areas of reduced functionality on Win 9x/ME are where Uniscribe is needed for the correct display of text. Specifically, on Win9x/ME you not see complex scripts being properly shaped in the Sample Text window, and you will be able to see supplemental plane characters (Unicode values U+10000 and above) only in the UCS-4 view rather than the Unicode view.

Using ViewGlyph's main window

Select desired font

Using the controls across the top of the main window, select the font name, script name, point size, and style (bold and/or italic) that you want ViewGlyph to use. [GeekNote: The font and script lists are generated using the EnumFontFamiliesEx() API]

The special script value *(Default)* indicates to Windows that you don't care what script gets used. [GeekNote: this causes ViewGlyph to use DEFAULT_CHARSET as the charset value.]

For TrueType fonts, the point size selection affects the Chart display, but not the Sample Text display. The size of the text in the sample text display is calculated to be as large as will fit (based on the metrics of the selected font).

The font name control is a combo box, so you may type any name you want. If the name you type does not represent an installed font, Windows will automatically select some other font for you — this mechanism is called the Windows font mapper, and it happens in any application. See the *Font Statistics* window to determine what font Windows chose.

Select desired view

This is where you decide how you want to view the font. Do you want to see it as 8-bit programs see it? Or as Unicode? Or perhaps you want to see the raw glyph palette. The available selections are dependent on the font, but might include:

Selection:	**What is shown:**	**Restrictions:**
Glyph Ids	Raw TrueType glyph palette	TrueType only
Windows Character Set	As seen by Windows 8-bit applications	
Windows Unicode	As seen by Windows Unicode applications	TrueType only
Windows Symbol	As seen by Windows Unicode applications	TrueType only
Windows UCS-4	As seen by Windows Unicode applications	TrueType only
Apple Unicode	As the font would look to Apple Unicode applications	TrueType with appropriate cmap table
Macintosh Character Set	As the font would look to Macintosh applications	TrueType with appropriate cmap table
Encore SILID	Arranged by Encore SILID	SIL Encore library font

GeekNote: selecting any except the first four entries *(Glyph Ids, Windows Character Set, Windows Unicode,* or *Windows Symbol)* causes ViewGlyph to do its own character-to-glyphID mapping by looking in the appropriate cmap table of the TrueType font.

Select CodePage

If you have selected *Windows Character Set* view, then you must select what codepage you want to use. The list you see will be dependent on what codepages are installed in your system, but will include at least the following values:

Selection:	**What codepage is used:**	**API used:**
(determined by font)	Whatever codepage identified by script selection	8-bit
(ANSI = nnnn*)*	The Default ANSI codepage, as defined by the system.	16-bit
(OEM = nnn*)*	The default OEM codepage as defined by the system.	16-bit
nnnn Name	nnnn	16-bit

In the case of *(ANSI* = nnnn*)* and *(OEM* = nnn*)*, the numeric value is the system-defined ANSI or OEM codepage number (GeekNote: as determined by GetACP() and GetOEMCP())

GeekNote: selecting *(determined by font)* causes ViewGlyph to use render 8-bit text using the 8-bit (or "ASCII") APIs. For any other codepage choice ViewGlyph uses MultiByteToWideChar() to convert 8-bit characters into Unicode, and then to render the 16-bit Unicode using the 16-bit ("wide") APIs.

GeekNote: The list of available codepages is determined by looking in the Registry rather than using EnumSystemCodePages() because the API doesn't work correctly on Windows 95 (according to some sources).

Look at the chart

The chart now shows how the selected font looks in the view you've requested. PageUp and PageDown keys will scroll the chart, as will manipulating the scrollbar. As the cursor drifts over the chart, information about the character under the cursor is displayed in the status bar at the bottom of the window. What is displayed will depend on the View that has been selected. The following abbreviations are used:

Code:	Meaning:	Numeric value:
WIN:	Windows 8-bit character code	decimal
MAC:	Macintosh 8-bit character code	decimal
U+	Unicode value (hex)	hex
GID:	Glyph ID, i.e., glyph palette index	decimal
SILID:	Encore SILID value	decimal

Additionally, the fourth panel in the status bar shows the PostScript name (if there is one) for the glyph.

Clicking on the chart will add the character to the Sample Text window. Right-clicking brings up a context menu which, if you are viewing *Windows Character Set* or any Unicode view, will include copying the character to the Windows clipboard. Copying as RTF causes the font name and script information to accompany the character. (Note: Word 97 running on Windows 9x doesn't support Unicode clipboard data, so using RTF is the only way, on Windows 9x, to copy Unicode text to Word 97),

Enter sample text

The edit box on the right side of the main window allows you to define a sequence of characters to be rendered together. The data must be entered as a space-separated sequence of character codes. Codes may be single character (meaning ANSI), decimal, or hex values. The screen shot of the main window above shows all three formats.

In addition to typing the codes directly into the text window, you can click on the chart and the code will be added to the sample text.

Remember: the codes in the sample text are interpreted according to the view you have selected. So if you choose a Unicode view, the codes are interpreted as Unicode values. If you select the Glyph ID view, the numbers are GIDs. This means that has you change views, the results in the sample text display will change.

The sample text display also supports a few right-click options, including copying text to the clipboard.

Supplementary Plane characters

For fonts that support SP (supplemental plane) characters (i.e., Unicodes U+10000 and above) there are two ways ViewGlyph can display them. If you choose the *Windows UCS-4* view then ViewGlyph does its own cmap lookups in both the chart and sample text displays, and this works on any version of Windows. On Windows 2000 or later you can see how Windows renders SP characters in the sample text window by selecting the *Windows Unicode* view. (NB: The chart will show only those characters below U+10000, but the sample text window can show anything). In this view the sample text window is displayed using Windows Unicode text APIs, and in order for you to see SP characters you must have Uniscribe enabled.

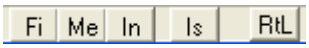
If you are having trouble seeing SP characters in *Windows Unicode* view, make sure you have not disabled language processing under *Options / Sample Window API.* Next, you may need to inform Windows that you need Uniscribe processing. This will be done automatically if you have configured Windows to process any complex languages (e.g., Arabic, Hebrew). But if not, you should add the following setting to your registry.

```
[HKEY_LOCAL_MACHINE\SOFTWARE\Microsoft\Windows NT\CurrentVersion\LanguagePack]
   SURROGATE=(REG_DWORD)0x00000002
```

This will enable Windows 2000 and later to use the internal Uniscribe module to display supplementary characters.

Right-to-left contexts

New in version 1.90 are 5 buttons on the right part of the status bar:



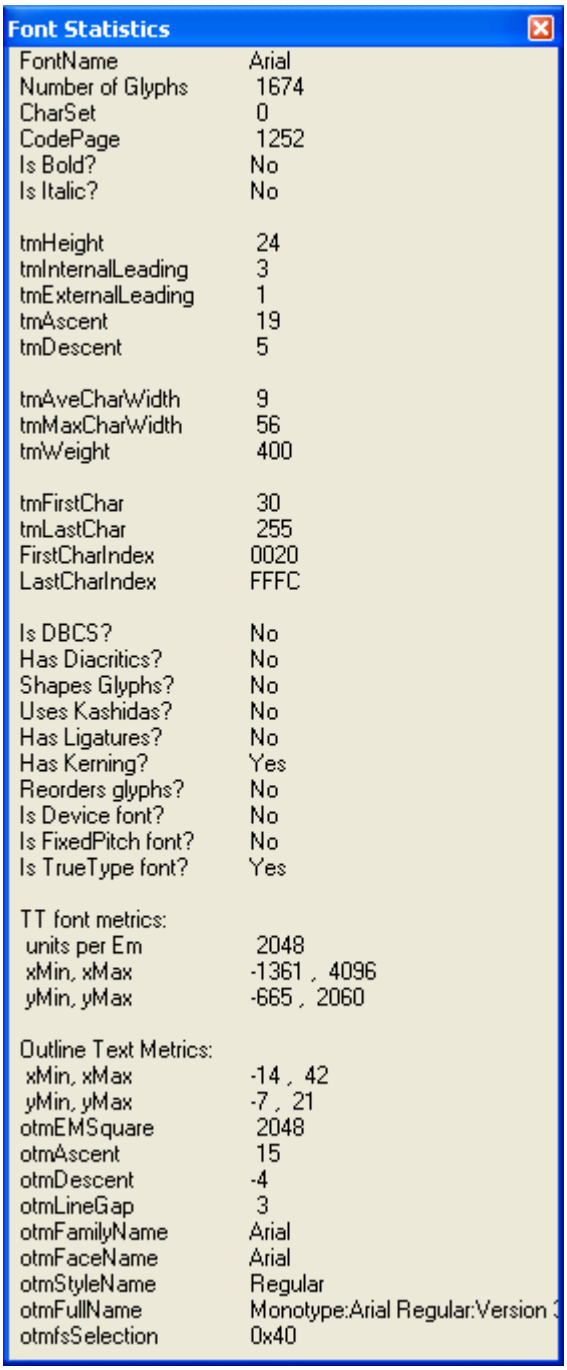
(Note that these disabled in some views.)

The first four turn on contextual shaping of characters in the Chart area. The buttons are, in order, Final, Medial, Initial, and Isolate, and they work by placing the Unicode character U+200D (ZERO WIDTH JOINER) before or after the character in the chart.

The final button turns on RTLREADING in both the Sample and Chart areas so that, for example, Unicode bidi mirroring occurs for characters like parenthesis.

Optional Windows

The *Options* menu allows you to open up some other windows. The Statistics, Cmap, and Names windows show additional information about the selected font. Note that the information in these other windows is independent of how you have chosen to *view* that font in the main window (i.e., the View and Codepage settings) — it is dependent only on the font you have selected.



Font Statistics Window

The Font Statistics window provides miscellaneous information gathered from a number of sources, including various APIs and the font tables themselves. Note that data collected directly from TrueType font tables will not be present if the font you have selected is not a TrueType font.

How to use the Font Statistics Window

Make sure the window is visible. If not, click *Options/View Statistics*. Position the window wherever convenient.

Suppose, in the main ViewGlyph window, you type in the name of a font that doesn't exist in your system? You can use the Font Statistics window to find out the name of the font that Windows substituted for your request.

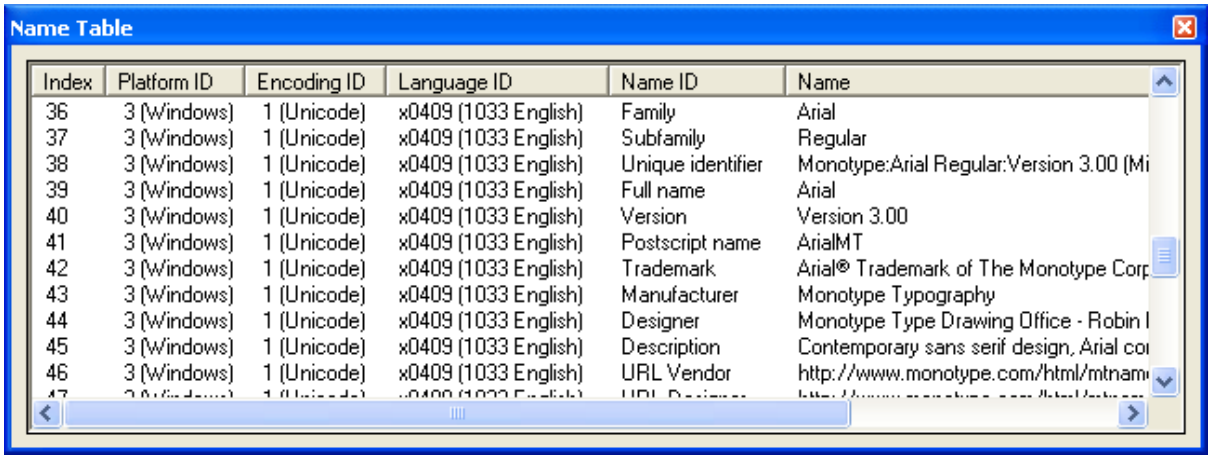
Want to know whether Windows is selecting a real Bold or Italic font (vs. algorithmically bolding or slanting the text)? Select a font in the main window and then click the Bold or Italic buttons. Watch the otmFaceName value in the Statistics window to see what font Windows was really choosing.

GeekNote: The data in the Font Statistics window are gathered from various APIs as follows:

Data:	**API:**
FontName	GetFaceName()
tm* and CharSet	GetTextMetrics()
Outline Text Metrics	GetOutlineTextMetrics()
Is DBCS, Has Diacritics, etc.,	GetFontLanguageInfo()
Number of Glyphs, TT font metrics	GetFontData() to read various TT font tables directly

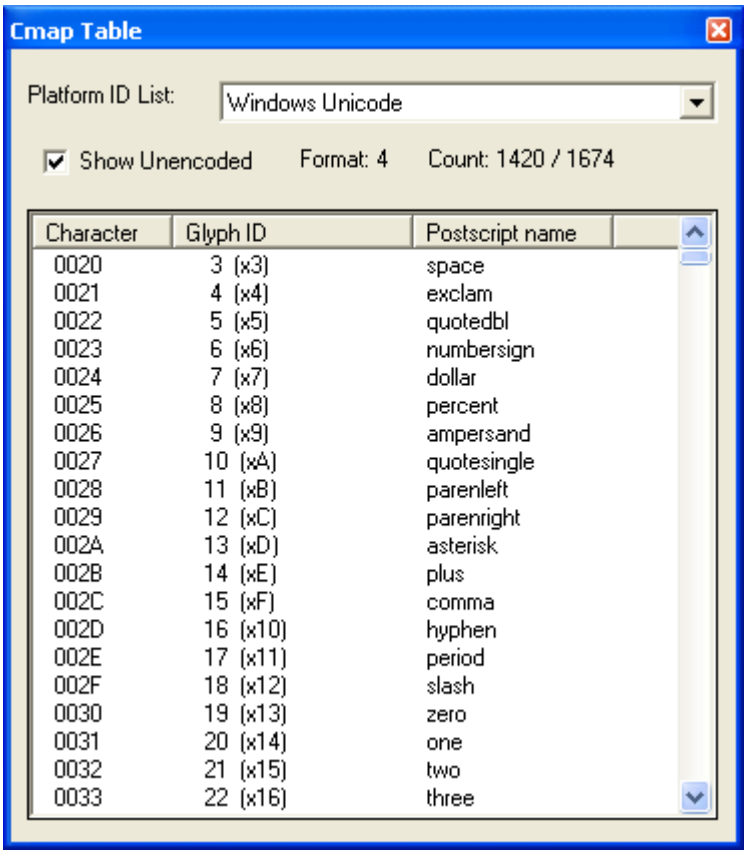
Names Window

The Names window, displayed when you click *Options/View Names*, shows the content of the TrueType 'name' table in the selected font. Here is a sample from Times New Roman font:



Some name strings can be very long. If you rest the cursor over one of the names, the "tooltip" will show more of the name. You can also select one or more rows and use ctl-c to copy them to the clipboard. (Note: As this is a VB6 application, the list control cannot display characters that are not in CP1252, so you may see question marks in some name entries.)

Cmap Window



A TrueType font's *cmap* table is what the operating system uses to map from a character code (on Windows this will be Unicode) to the glyph palette in the font. Most fonts have different cmaps for different operating systems. The Cmap window allows you to see any of the cmaps in a font. Once you click *Options/View Cmap*, then you need to select the platform whose cmap you want to see. The Postscript names shown in the Cmap window are not actually part of the cmap table, but are extracted from the post table and shown for convenience.

The cmap list allows you to select one or more lines and copy them to the clipboard using ctl-c. Checking the *Show Unencoded* checkbox causes the list to include all glyphs whether or not they are mentioned in the cmap. As with other list view windows, you can sort the data by clicking on a column header.

NB: Very large fonts, e.g., Arial Unicode which attempts to cover all of Unicode, may require as much as minute or two to load the cmap list, and up to 10 or 15 seconds to clear the cmap list. Just be patient.

Other Options

When ViewGlyph is used on Windows 2000, you may find that the chart view gives odd results for certain non-Roman scripts, e.g., Arabic. This is because the Uniscribe shaping engine is kicking in and attempting to reorder and shape the characters as if they were running text. To avoid this, select *Options / Chart window API / Extended (language processing disabled).* (GeekNote: The *Chart window API* and *Sample window API* options determine whether TextOut() or ExtTextOut() APIs are used in the chart and sample text displays. The *language processing disabled* setting adds ETO_IGNORELANGUAGE flag to ExtTextOut().)

Viewing uninstalled fonts

Ordinarily you will launch ViewGlyph and then select a font from the dropdown list. But what if you want to view a font that is not yet installed in your system? You can use File/Open to browse for a TTF, or you can drag a font from an Explorer window onto ViewGlyph's main window, or you can launch ViewGlyph with a font file

name as a parameter. This means you can associate .TTF files with ViewGlyph in order to be able to double-click a font and see it with ViewGlyph.

It should be noted that when you ask ViewGlyph to view an uninstalled font, it temporarily installs the font in order to view it. If you had some other font by the same name installed, that font is temporarily uninstalled. In any case, as soon as you view a different font, ViewGlyph returns everything to its original state.

Known bugs and limitations

Some Macintosh fonts, e.g., Mac Times, cannot be installed in Windows because they don't have the required tables. ViewGlyph cannot help you.

Revision History

1.81.0000	1-Oct-2009	Fix	Fix overflow error on startup (Window positions in INI > 32K)
		New	Made cmap window auto-load Unicode cmap if present.
		New	Additional otm* metrics in the font statistics.
1.80.0000	23-Aug-2007	New	Statusbar buttons to show Initial, Medial, Final and Isolate forms, and also to turn on right-to-left reading.
1.79.0000	10-Mar-2007	Fix	Cmap overflow for certain large fonts (notably from Microsoft Vista)
1.77.0000	6-May 2004	New	Home and End keys scroll chart
1.76.0000	12-May-2003	Fix	Crashed on fonts with more than 32768 post names
1.75.0000	17-Feb-2003	New	Support for supplemental plane chars ("surrogates")
		New	Copy-to-clipboard from sample text window
		New	Copy-to-clipboard permitted in any Unicode view (e.g., Mac Unicode)
		Fix	Dragging certain objects over main window caused runtime error
		Fix	Malformed cmap could put program in infinite loop
		Fix	Did not handle fonts with multiple MacRoman cmaps
1.74.0000	09-Sep-2002	New	Hex and decimal display of character and glyph codes
1.73.0000	09-Jul-2002	New	Main window now fully sizable.
		New	Options to control Chart and Text display APIs, including ability to suppress Uniscribe processing (via ETO_IGNORELANGUAGE)
		New	Names window supports multiple-select and ctl-c copies selected items to the clipboard.
		Fix	Corrected postscript names for Mac-std characters
		Fix	Was not remembering font size when closed while viewing a font file.
1.72.0000	10-Dec-2001	Fix	Print of Windows Character Set, for symbol fonts, gave error or incorrect results.
1.71.0000	16-Mar-2001	New	cmap window can show unencoded glyphs, supports multiple-select, and ctl-c copies selected items to clipboard.
1.70.0000	11-Oct-2000	New	Can now open font files (via File/Open, command line, or drag&drop)
		New	Added menu to expose options.
		New	Displays PostScript names of glyphs
		New	'cmap' and 'name' table views
		New	Font Statistics window can be separately positioned or even closed
		Fix	Print of Windows Character Set doesn't show correct chars
		Fix	PageUp sometimes generated an Overflow message
		Fix	Captions on Printed charts should now be readable
1.60.0008	10-Nov-1999	Fix	IsFixedPitch display was backwards
		New	Added font statistics such as "Reorders", etc. to display
1.60.0007	07-Nov-1999	New	Option to display chart one char at a time (needed for Win2000 because W2K does shaping even for simple TextOut). Turn option on and off via chart context menu
		New	Added progress indicator when retrieving LARGE cmaps.
1.60.0006	15-Sep-1999	Fix	Certain Glyph IDs weren't calculated correctly in printed chart
		Fix	Change heading to Arial (for PM)
		Fix	Fonts with more than 32000 glyphs broke the program in several places. Printing of such fonts not fully tested yet.
1.60.0005	11-Nov-1998	Fix	Certain Glyph IDs weren't calculated correctly in printed chart
		Fix	Change heading to Arial (for PM)
		Fix	Heading on first page was sometimes in wrong position on printout
1.60.0004	11-Nov-1998	New	Ability to print a chart (press Ctl-P)
		Fix	Unicode values for Symbol fonts now correct.
1.60.0003	02-Oct-1998	New	Added ability to copy unformatted Unicode to clipboard. Word97 on Win95 doesn't recognize it, but it does on WinNT
1.60.0003	02-Oct-1998	New	Added ability to copy unformatted Unicode to clipboard. Word97 on Win95 doesn't recognize it, but it does on WinNT
1.60.0002	30-Sep-1998	Fix	Split "copy to clipboard" into "copy formatted text to clipboard" and "copy unformatted text to clipboard" to clarify. As a side effect, it always copies unformatted text to the clipboard so non-RTF apps still get something useful.
		Fix	Context menu in Chart now restores mouse position so selected character doesn't change.

		New	Added Cancel to chart context menu
1.60.0001	30-Sep-1998	Fix	Form load generated error when loading 1.6.0 View settings.
		Fix	Kept returning to “Apple Unicode” for fonts that had such a cmap
		Fix	Spelling correction in Font Statistics window
		New	Copy to clipboard from right-click on chart
1.60.0000	24-Sep-1998	Fix	Character mapping displayed in the status panel was based on the font selected rather than on the codepage being viewed.
		Fix	In status panel, sometimes “GID:” didn’t prefix the glyph id.
		Fix	“(Default)” codepage changed to “(determined by font)” since this more accurately reflects what is going on inside. Changed “(Default ANSI)” and “(Default OEM)” to show the numeric value based on GetACP() and GetOEMCP() API.
1.50.0010	03-Sep-1998	Fix	TT Fonts such as SaintFrancis that do not have Windows cmap (have only Mac cmap) caused error.
1.50.0009	03-Sep-1998	Fix	Wouldn’t see certain fonts such as “GF Zemen Unicode” because it isn’t enumerated except in the “EX” API. Therefore:
		New	Now uses advanced API to allow selection of font by name and script. The Script dropdown list selection sets the Charset value used when picking a font. A special “(Default)” entry allows for “DEFAULT_CHARSET”.
		New	Changed Bold and Italic selections to buttons with Ctl-B and Ctl-I accelerators
		New	“(Default)” codepage entry now shows numeric value based on GetACP() API.
1.50.0005	18-Aug-1998	Fix	Switching from non-TT to TT didn’t re-enable CodePage list
		Fix	Switching from non-TT to TT didn’t restore View list selection
1.50.0004	18-Aug-1998	Fix	Would abend if user entered empty string for font name.
1.50.0001	??-Aug-1998	Fix	Would lock up on NT -- bad calling conventions on RegEnumValue()
1.50.0000	10-Aug-1998	New	First public release of VB 5 version of program

