# 6. Smart Fonts

## Graphite: An Extensible Rendering System

SIL (Summer Institute of Linguistics) International has been working with minority languages for over fifty years in the areas of linguistics, literacy and translation. As computers became integral to the work, computing solutions were sought that could handle complex scripts for minority groups. During the 1990s, a team was established to investigate problems of non-Roman computing. After some investigation, it was concluded that an extensible rendering system was needed to handle existing problems with non-Roman computing. The Graphite system has been released in a beta version 0.8 (see http://graphite.sil.org).

- Full Unicode compliance, including support for the Private Use Area
- Left-to-right and right-to-left rendering and bidirectionality
- Glyph reordering, as needed for Indic scripts
- Extensive contextualization, including ligatures, startand end-of-line forms and contextualization across line breaks
- Stacked diacritics
- Kerning
- Full editing support, including independent editing of ligature components

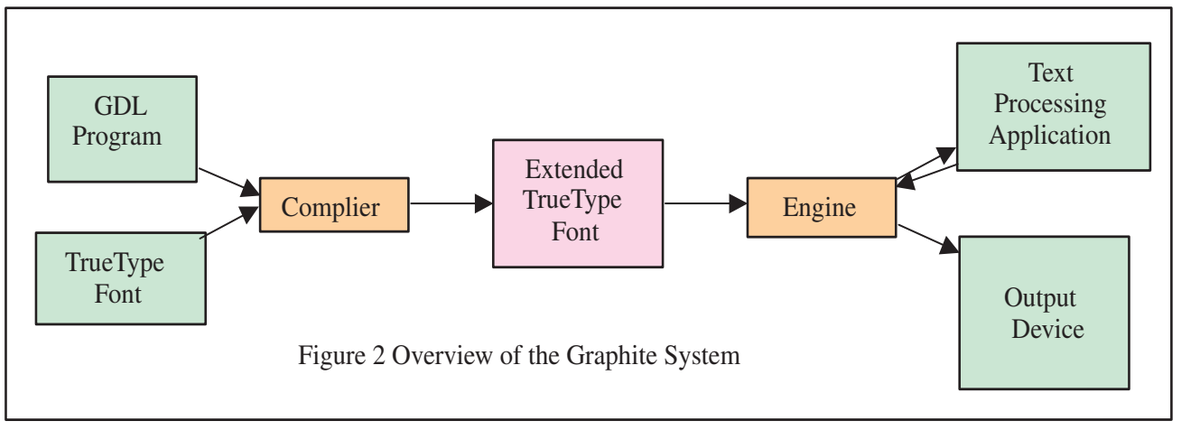


Figure 2 Overview of the Graphite System

Graphite includes a compiler for a high-level rule-based programming language called the Graphite Description Language (GDL) and an engine that uses the resulting font files for rendering. A GDL program is compiled against a TrueType font, creating custom tables stored in the font. These are used by the rendering engine to find line-breaks within the text, draw the text on the screen, perform cursor tracking, use arrow keys, and highlight selections. An application makes calls to the Graphite engine to render text on the display or printer. The following diagram shows these relationships:

The Graphite specification proposed all of the following to be features of the Graphite system:

- Split cursor support

Graphite is designed to render up to one line of text in a single writing system. The application performs paragraph layout and rendering of a mixture of writing systems through repeated calls to the Graphite renderer. The application is also responsible for managing the details of editing the text. The concept of WorldPad originated from the need for a text editor with extensible rendering capabilities.

Two features that are not yet available in Graphite are justification and vertical text layout. These and other features are under consideration for future development.

Graphite is a package that can be used to create "smart fonts" capable of displaying writing systems with various complex behaviors, such as:

- Contextual shaping
- Ligatures
- Reordering
- Split glyphs
- Bidirectionality
- Stacking diacritics
- Complex positioning

## What is a "smart font"?

A "smart font" is a font with accompanying data that contains instructions describing how the glyphs are to be selected and laid out on the screen or display device. This is in contrast to a "dumb font," where there is a direct correspondence between the data characters and the displayed glyphs, and the glyphs are laid out simply in a straight line, side by side.

Typically the smart font data is in the form of tables inside the font file itself. Some sort of engine or driver software is required to read the tables from the font and use them appropriately in the rendering process.

In other words, a smart font can be thought of as a font with a small computer program built into it. The engine is the interpreter that runs the program.

## Different smart font formats:

Three popular smart font technologies are: Apple Advanced Typography, developed by Apple; OpenType, developed by Microsoft and Adobe; and Graphite, developed by SIL International. These technologies are based on TrueType fonts with the rules stored into special TrueType data tables. Apple Advanced Typography support is included in MacOS X. OpenType is supported in Windows 2000 and Windows XP and in Linux by the GTK. SIL International provides an OpenSource library which supports Graphite. The ICU LayoutEngine and Layout Extensions Library provide an OpenSource implementation of OpenType which one can use on platforms where it isn't already supported.

The individual font formats used by all these systems are based on the TrueType font format. Essentially each system places an extended set of tables in a TrueType font file containing rules and lookups the rendering system can apply in order to display the underlying text correctly using that font. Since the TrueType font format has extensible tables, and a font developer can add any number of them to a font, it is theoretically possible (but a great deal of work) to create a single font with the different tables necessary to for that font to work with all three systems.

## Why was Graphite developed?

SIL International is a non-profit organization that performs linguistic research, literacy development, and translation work among ethnic minorities around the world. Their members are currently working with approximately a thousand language groups on six continents, and increasingly, many of the areas in which SIL work use scripts that require complex rendering.

Traditionally, much of SILs work has been with preliterate groups, or those that are literate only in a national or trade language. In cases where there is no written form for the target language, orthography development is a significant part of our field workers' literacy promotion efforts. There are a variety of linguistic, sociological, cultural, and political factors that play into the orthography issue, and often it is necessary to adapt the script of the national language for use with minority languages.

The minority languages are usually quite distinct from the national language, much less closely related in a linguistic sense than what one might think of as a "dialect." Therefore it is common to find linguistic phenomena in the minority language that are not present in the national language and which considerably complicate the issue of orthography

development. For example, it may be the case that a minority language is tonal, while the national language is not, and the orthographic solution involves using the standard writing system with some extra diacritics to indicate tone. Or the minority language might have a set of sounds characterized by a certain linguistic feature, such as aspiration or breathiness, that are not present in the national language, and the desire is to add to the standard orthography a set of variant characters to represent these variant sounds.

So computer implementations of writing systems for many minority languages require processing that is not supported in implementations developed for the better-known languages. Some languages use scripts that have not been implemented at all, and must be represented entirely using the Unicode Private Use Area characters. Others require a handful of special characters with complex behaviors, or have special rules about how diacritics are positioned, or how existing characters combine.

To handle the needs of the wide variety of linguistic communities where SIL works, then, SIL need an extensible mechanism adequate to handle any orthographic phenomenon we might encounter in any writing system based on any modern script. The only other modern extensible smart font technology adequate to handle all the world's writing systems is Apple's AAT, and the Macintosh is not appropriate for use in many of the communities in which SIL works. So Graphite was developed specifically to meet this need on Windows, the platform used by most of our field workers.

## How is Graphite different from OpenType and other complex script technologies?

Open Type is a widely accepted and powerful smart font technology developed jointly by Adobe and Microsoft. The fundamental difference between Graphite and OpenType is that OpenType was not developed as a full writing system implementation system. Rather the assumption is that script-specific knowledge—such as the fact that certain Devanagari vowels are written before the preceding consonant, or that certain Arabic letters have four distinct forms—is incorporated at the application level or in an operating system module such as Uniscribe. For this reason, OpenType does not include the capability to handle some complex behaviors such as Indic reordering.

Uniscribe is a module for handling complex-scripts that is built into recent versions of the Windows operating system. It is a layer built on top of OpenType font technology, and contains knowledge of script behaviors of most scripts that are part of the Unicode standard. However, its behavior cannot be extended, as is often needed for minority language groups, and it provides no support for characters in Unicode's Private Use Area ranges.

Apple has for years provided truly extensible smart font support for potentially all known writing systems, through their earlier GX and more recent AAT technologies. However, their solutions are proprietary and available only on the Macintosh. Graphite has been implemented on Windows and is open-sourced. Graphite also includes a rule-based programming language (GDL) to facilitate font development, while AAT development requires more direct construction of the font tables.

Pango is a open-source framework to handle internationalized text and complex scripts, developed on the Linux platform. Unlike Graphite, it is not a smart font technology, but rather a system in which script behaviors are coded directly in a high-level programming language. Currently work is underway to integrate Pango and Graphite in such a way that Pango will call the Graphite engine if the font being used is Graphite-enabled.

## What is a "complex script"?

Asian scripts derived from Arabic and Brahmi are called "complex scripts" since:

- the logical and visual order of characters may be different (requiring reordering)

- Arabic text is read from right to left, but numbers are read from left to right (requires bi-directional rendering)
- the form and position in which underlying nominal characters should be rendered is variant and dependant on the context of the characters (requiring context dependant glyph substitution and positioning)
- sequences of characters often form complex ligatures (requiring many to one glyph substitution and composition)
- some characters may have multiple glyph parts which are split e.g. the Tamil short-O vowel (requiring one to many decomposition)

## On what platforms is Graphite available?

Graphite currently exists on the Windows platform (Windows 98 and later). A port to Linux is underway.

## What applications support Graphite?

WorldPad is a basic text editor developed by SIL that can render text using Graphite. There is project underway called SILA with the purpose incorporating Graphite support into the Mozilla web browser and related applications. A beta version is available. SIL is developing a suite of linguistic and anthropological research tools called FieldWorks. Graphite support will be included in version 2.0 of these tools.

Efforts are underway to incorporate Graphite support into the OpenOffice word processor.

A Graphite-enabled *ActiveX control* is also under development.

## What Graphite fonts exist?

Currently there are no Graphite-enabled fonts that have been officially released. A Roman/Cyrillic font with support for *IPA* and stacking diacritics is available in beta, and there is an implementation of *Burmese* that is nearly complete. Some work has been done on *Thai*, *Arabic* and *Khmer*. A font to display *Neo-Sumerian Cuneiform* is under development by Karljürgen Feuerherm. Graphite Fonts can be downloaded from http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&item_id=GraphiteFonts

## How do make own Graphite font?

The following is the process through to create a Graphite-enabled font:

1. Download the Graphite compiler and the WorldPad application which will be used to test the font. The WorldPad package includes the Graphite rendering engine.
2. Locate the font containing the glyphs for your writing system. One will need a font whose license allows one to modify it. It may be helpful to make some initial modification to the font, although this is rarely absolutely necessary. Such modifications might include adding attachment points to the glyph curves, or assigning postscript names to the glyphs.
3. Write a program using the Graphite Description Language (GDL), describing the smart behavior required in the font.
4. Compile the GDL program and the font together, using the Graphite compiler. The output is the Graphite-enabled version of the font.
5. Test the font using WorldPad. One may also need to find or develop an appropriate keyboard to use to input data. Fix any bugs in your GDL program and recompile.

## What kind of license do we need to make a Graphite font?

All the software to create and test a Graphite-enabled version of a font is freely available on this web site. However, to put the Graphite tables into the font, user will need to have a license permitting him to modify the font. To share the font with others, one will need a license permitting one to distribute the font.

## The Future of Graphite

Presently Graphite is only useful to programs which can interface with its application programming interface or API. One aspect of the vision for Graphite is the desire for this technology to become widely available throughout the world. The initial development is complete and has been made available as open source. It is hoped that further refinement be undertaken by others interested in the problems of non-Roman and complex Roman computer support. It would be particularly helpful if this technology were incorporated in other open source projects. The ultimate hope is for multilingual and multiscript support to be available for all computing systems. Educators and language specialists need to encourage the provision of fonts and rendering technologies for all languages. Such support will further the goals of literacy and language learning for all.

Courtesy/Source:

http://www.techknowlogia.org/TKL_active_pages2/CurrentArticles/main.asp?FileType= HTML & ArticleID=349

http://scripts.sil.org/cms/scripts/page.php?site_id=nrsi&item_id=GraphiteFAQ

http://iris.lib.virginia.edu/tibet/xml/showEssay.php?xml=/tools/smartfonts.xml &m=print