

1 Welcome to TrueType Open

TrueType Open is an extension to the TrueType font standard. TrueType Open fonts contain additional information that extends the capabilities of the fonts to support high-quality international typography:

- TrueType Open fonts allow a rich mapping between characters and glyphs, which supports ligatures, positional forms, alternates, and other substitutions.
- TrueType Open fonts include information to support features for two-dimensional positioning and glyph attachment.
- TrueType Open fonts contain explicit script and language information, so a text-processing application can adjust its behavior accordingly.
- TrueType Open fonts have an open format that allows font developers to define their own typographical features.

This overview introduces the power and flexibility of the TrueType Open font model. The rest of the document describes the TrueType Open components in technical detail that will be useful to font vendors and software developers creating TrueType fonts and the text-processing applications that use them.

TrueType Open at a Glance

TrueType Open addresses complex typographical issues that especially affect people using text-processing applications in multi-lingual and non-Latin environments. TrueType Open fonts may contain alternative forms of characters and mechanisms for accessing them. For example, in Arabic, the shape of a character often varies with the character's position in a word. As shown here, the Ha character will take any of four shapes, depending on whether it stands alone or whether it falls at the beginning, middle, or end of a word. TrueType Open helps a text-processing application determine which variant to substitute when composing text.



Figure 1a. Isolated, initial, medial, and final forms of the Arabic character Ha

Similarly, TrueType Open helps an application use the correct forms of characters when text is positioned vertically instead of horizontally, such as with Kanji. For example, Kanji uses alternative forms of parentheses when positioned vertically.

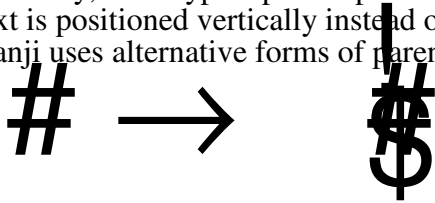


Figure 1b. Alternative forms of parentheses used when positioning Kanji vertically

The TrueType Open font format also supports the composition and decomposition of ligatures. For example, English, French, and other languages based on Latin can substitute a single ligature, such as “fi”, for its component glyphs—in this case, “f” and “i.” Conversely, the individual “f” and “i” glyphs could replace the ligature, possibly to give a text-processing application more flexibility when spacing glyphs to fill a line of justified text.

5 + 6 = 7

Figure 1c. Two Latin glyphs and their associated ligature

w + e + r = 8

Figure 1d. Three Arabic glyphs and their associated ligature

Glyph substitution is just one way TrueType Open extends font capabilities. Using precise X and Y coordinates for positioning glyphs, TrueType Open fonts also can identify points for attaching one glyph to another to create cursive text and glyphs that need diacritical or other special marks.

TrueType Open fonts also may contain baseline information that specifies how to position glyphs horizontally or vertically. Because baselines may vary from one script (set of characters) to another, this information is especially useful for aligning text that mixes glyphs from scripts for different languages.

Boston zkkj gfdsap iuyt

Figure 1e. A line of text, baselines adjusted, mixing Latin and Arabic scripts

TrueType Versus TrueType Open

A TrueType font is a collection of several tables that contain different types of data: glyph outlines, metrics, bitmaps, mapping information, and much more. TrueType Open fonts contain all this basic information, plus additional tables containing information for advanced typography.

Text-processing applications—referred to as *clients* of TrueType Open—can retrieve and parse the information in TrueType Open tables. So, for example, a text-processing client can choose the correct character shapes and space them properly.

As much as possible, the tables of TrueType Open define only the information that is specific to the font layout. The tables do not try to encode information that remains constant within the conventions of a particular language or the typography of a particular script. Such information that would be replicated across all fonts in a given language belongs in the text-processing application for that language, not in the fonts.

TrueType Open Terminology

The TrueType Open model is organized around glyphs, scripts, language systems, and features.

Characters Versus Glyphs

Users don't view or print characters: a user views or prints *glyphs*. A glyph is a representation of a character. The character “capital letter A” is represented by the glyph “A” in Times New Roman Bold and “A” in Arial Bold. A TrueType font is a collection of glyphs. To retrieve glyphs, the client uses information in the cmap table of the font, which maps the client's character codes to glyph indices in the table.

Glyphs can also represent combinations of characters and alternative forms of characters: glyphs and characters do not strictly correspond one-to-one. For example, a user might type two characters, which might be better represented with a single ligature glyph. Conversely, the same character might take different forms at the beginning, middle, or end of a word, so a font would need several different glyphs to represent a single character. TrueType Open fonts contain a table that provides a client with information about possible glyph substitutions.

The image shows five distinct typographic variations of the ampersand character (&). Each variation is a solid black glyph on a white background. They are numbered 1 through 5 from left to right. Variation 1 is a simple, clean ampersand. Variation 2 has a slightly more decorative, rounded top. Variation 3 is a classic, slightly stylized ampersand. Variation 4 is a more ornate, calligraphic-style ampersand. Variation 5 is a very decorative, almost floral ampersand.

Figure 1f. Multiple glyphs for the ampersand character

Scripts

A script is composed of a group of related characters, which may be used by one or more languages. Latin, Arabic, and Thai are examples of scripts. A font may use a single script, or it may use many scripts. Within a TrueType Open font, scripts are identified by unique 4-byte *tags*.

The image shows three distinct glyphs side-by-side. From left to right: a Latin lowercase 'w', a Kanji hash symbol (a stylized '3' shape), and an Arabic numeral '3'. All three glyphs are rendered in a bold, black, serif font style.

Figure 1g. Glyphs in the Latin, Kanji, and Arabic scripts.

Language Systems

Scripts, in turn, may be divided into language systems. For example, the Latin script is used to write English, French, or German, but each language has its own special requirements for text processing. A font developer can choose to provide information that is tailored to the script, to the language system, or to both.

Language systems, unlike scripts, are not necessarily evident when a text-processing client examines the characters being used. To avoid ambiguity, the user or the operating system needs to identify the language system. Otherwise, the client will use the default language-system information provided with each script.

A charming mess
Le cahier français
Das Wasser war heiß

Figure 1h. Differences in the English, French, and German language systems.

Features

Features define the basic functionality of the font. A font that contains tables to handle diacritical marks will have a “mark” feature. A font that supports substitution of vertical glyphs will have a “vert” feature.

The TrueType Open feature model provides great flexibility to font developers because features do not have to be predefined by Microsoft Corporation. Instead, font developers can work with application developers to determine useful features for fonts, add such features to TrueType Open fonts, and enable client applications to support such features. The TrueType Open Tag Registry at the end of this document contains a current list of Microsoft registered features.

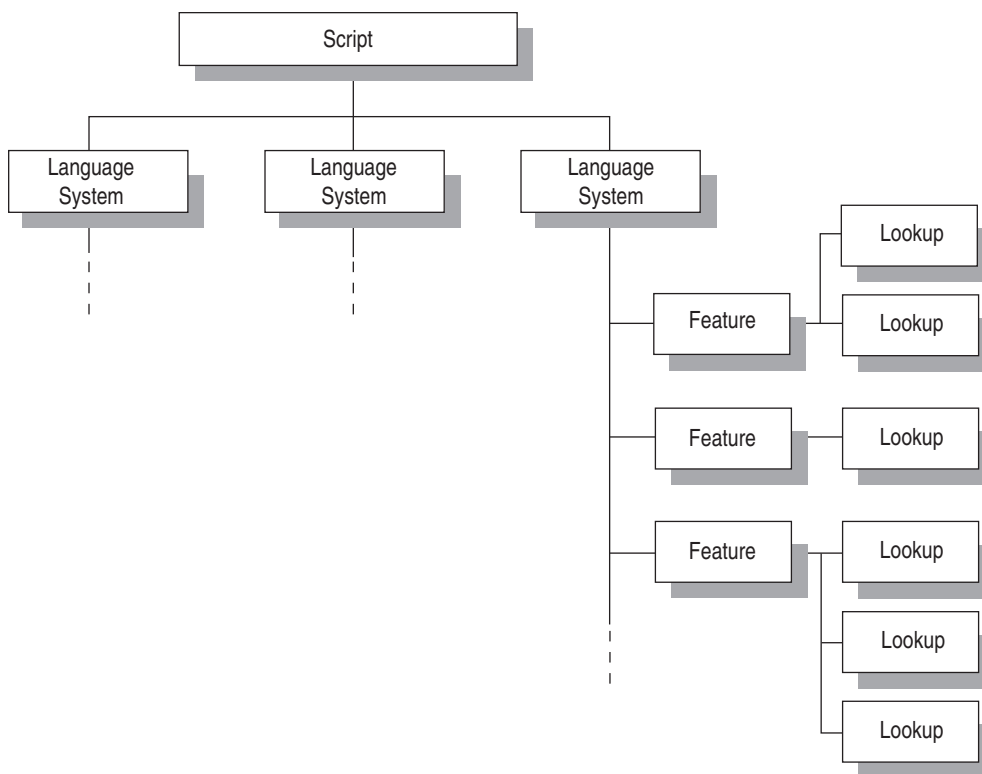


Figure 1i. The relationship of scripts, language systems, features, and lookups for substitution and positioning tables

TrueType Open Tables

TrueType Open comprises five new tables: GSUB, GPOS, BASE, JSTF, and GDEF. These tables and their formats are discussed in detail in the chapters that follow this overview.

TrueType Open Tables

Name	Description
GSUB	Contains information about glyph substitutions to handle single glyph substitution, one-to-many substitution (ligature decomposition), aesthetic alternatives, multiple glyph substitution (ligatures), and contextual glyph substitution.
GPOS	Contains information about X and Y positioning of glyphs to handle single glyph adjustment, adjustment of paired glyphs, cursive attachment, mark attachment, and contextual glyph positioning.
BASE	Contains information about baseline offsets on a script-by-script basis.
JSTF	Contains justification information, including whitespace and Kashida adjustments.
GDEF	Contains information about all individual glyphs in the font: type (simple glyph, ligature, or combining mark), attachment points (if any), and ligature caret (if a ligature glyph).

Text Processing with TrueType Open Fonts

The design of TrueType Open is based on the following text-processing model:

1. Using the cmap table in the font, the client converts the character codes into a string of glyph indices.
2. Using information in the GSUB table, the client modifies the resulting string, substituting positional or vertical glyphs, ligatures, or other alternatives as appropriate.
3. Using positioning information in the GPOS table and baseline offset information in the BASE table, the client then positions the glyphs.
4. Using *design coordinates*, the client determines device-independent line breaks. Design coordinates are high-resolution and device-independent.
5. Using information in the JSTF table, the client justifies the lines, if the user has specified such alignment.
6. The operating system rasterizes the line of glyphs and renders the glyphs in *device coordinates* that correspond to the resolution of the output device.

Throughout this process the text-processing client keeps track of the association between the character codes for the original text and the glyph indices of the final, rendered text. In addition, the client may save language and script information within the text stream to clearly associate character codes with typographical behavior.

TrueType Open Fonts in Windows 95

The core system fonts in the Middle East and Far East versions of Windows 95 are TrueType Open fonts. These fonts demonstrate aspects of TrueType Open's versatility.

Middle East Windows 95

Middle East Windows 95 uses several Arabic TrueType Open fonts: fixed regular weight, proportional regular weight, fixed bold, and proportional bold. These fonts take advantage of many glyph substitution features available in TrueType Open, namely simple substitution (one-to-one contextual), ligature substitution (many-to-one), and mark set substitutions. In Middle East Windows 95, the operating system itself handles glyph substitution, using data in the GSUB table of each font.

Far East Windows 95

Far East Windows 95 also uses several Japanese TrueType Open fonts: fixed serif, proportional serif, fixed sans serif, and proportional sans serif. The Japanese fonts take advantage of a subset of TrueType Open features, including vertical glyph substitution and baseline positioning. As with Middle East Windows 95, the operating system in Far East Windows 95 will handle glyph substitution, using data in the GSUB table in each font. However, the text-processing client will need to handle baseline positioning, using data in the BASE table in each font.

Frequently Asked Questions

Q. Is TrueType Open a new font format?

A. No. TrueType Open fonts are TrueType fonts with extensions. TrueType Open introduces new table structures that contain additional typographical data.

Q. To ensure compatibility, do I need to revise all my fonts to be TrueType Open fonts?

A. No. TrueType fonts that do not contain TrueType Open information are still valid fonts. A client can determine which, if any, TrueType Open features a font contains and make decisions accordingly.

Q. Can I use TrueType Open fonts on Windows 3.1x?

A. Yes. Because the basic TrueType font format has not changed, TrueType Open fonts can be used by operating systems and applications that support TrueType but do not implement TrueType Open functionality.

Q. Which TrueType Open features are supported by Windows 95?

A. A text-processing client can access all TrueType Open tables by using the GetFontData API and requesting a TrueType Open table by name. The client is responsible for placing text appropriately according to the information retrieved.

Q. Is Microsoft developing tools for creating TrueType Open fonts?

A. Microsoft is working with Independent Software Vendors (ISVs) to enhance their font development tools to handle TrueType Open information. Internally, Microsoft uses a proprietary tool for assembling TrueType Open binary tables from TrueType Open data in text format.

Q. What is the difference between Apple's TrueType GX and Microsoft's TrueType Open?

A. TrueType Open and TrueType GX address some of the same issues, especially those arising from one-to-one relationships between characters and glyphs. However, TrueType Open has a richer two-dimensional positioning model than TrueType GX. TrueType Open also contains explicit script and language support, so a text-processing application can adjust its behavior accordingly.

Another major difference between the two font formats is that TrueType GX fonts contain *state machines*, which choose and position glyphs. This format requires that clients follow a particular model of text processing, and TrueType GX fonts are difficult for a client to decode. In comparison, TrueType Open fonts are simple to read and support clients in making their own decisions regarding text processing.