

An introduction to OpenType

What's in a name

OpenType = Open + Type?

What's in a name

OpenType = Open + Type? No!

What's in a name

Extension to the TrueType font format and unification with the PostScript Type1 font format.

What's in a name

Extension to the TrueType font format and unification with the PostScript Type1 font format.

- An additional description language for the outlines ("unification" with Type 1 fonts);

What's in a name

Extension to the TrueType font format and unification with the PostScript Type1 font format.

- An additional description language for the outlines ("unification" with Type 1 fonts);
- Enhanced font capabilities ("extension" of the format).

Outlines description

The forms of the letters in a font are described using *quadratic splines* (Bézier curves of degree 2); Type 1 fonts use *cubic* splines.

Outlines description

The forms of the letters in a font are described using *quadratic splines* (Bézier curves of degree 2); Type 1 fonts use *cubic* splines.

⇒ OpenType enhances the TrueType format to enable use of either of these description (mutually exclusive).

Font capabilities

A "modern" digital font (TrueType, PostScript Type 1, etc.) does not only contains the form of the letters (*glyphs*); it has information on interaction between them, in particular :

Font capabilities

A "modern" digital font (TrueType, PostScript Type 1, etc.) does not only contains the form of the letters (*glyphs*); it has information on interaction between them, in particular :

- Ligatures ($f + i \rightarrow fi \neq fi$);

Font capabilities

A "modern" digital font (TrueType, PostScript Type 1, etc.) does not only contains the form of the letters (*glyphs*); it has information on interaction between them, in particular :

- Ligatures ($f + i \rightarrow fi \neq fi$);
- Kerning ($A + V \rightarrow AV \neq AV$).

Font capabilities

A "modern" digital font (TrueType, PostScript Type 1, etc.) does not only contains the form of the letters (*glyphs*); it has information on interaction between them, in particular :

- Ligatures ($f + i \rightarrow fi \neq fi$);
- Kerning ($A + V \rightarrow AV \neq AV$).

This information is enough to set text in "Western" languages and can (ab)used to typeset complex scripts (e.g., Arabic, Indic scripts).

Font capabilities

A "modern" digital font (TrueType, PostScript Type 1, etc.) does not only contains the form of the letters (*glyphs*); it has information on interaction between them, in particular :

- Ligatures ($f + i \rightarrow fi \neq fi$);
- Kerning ($A + V \rightarrow AV \neq AV$).

This information is enough to set text in "Western" languages and can (ab)used to typeset complex scripts (e.g., Arabic, Indic scripts).

... but it's by far not everything about fonts!

Features

For a better handling of the Western scripts, and for complex scripts, OpenType introduces the concept of *features*.

Features

For a better handling of the Western scripts, and for complex scripts, OpenType introduces the concept of *features*.

A feature is a set of rules for moving or replacing glyphs or groups or glyphs by others.

Features

For a better handling of the Western scripts, and for complex scripts, OpenType introduces the concept of *features*.

A feature is a set of rules for moving or replacing glyphs or groups or glyphs by others.

These rules are defined in two new tables: GPOS for moves (Glyph POSitioning), and GSUB for substitutions (Glyph SUBstitution).

Features

For a better handling of the Western scripts, and for complex scripts, OpenType introduces the concept of *features*.

A feature is a set of rules for moving or replacing glyphs or groups or glyphs by others.

These rules are defined in two new tables: GPOS for moves (Glyph POSitioning), and GSUB for substitutions (Glyph SUBstitution).

GPOS and GSUB abilities include very general modifications like contextual substitution and positioning (e.g., move the second dot in "S.A.V.").

Features

For a better handling of the Western scripts, and for complex scripts, OpenType introduces the concept of *features*.

A feature is a set of rules for moving or replacing glyphs or groups or glyphs by others.

These rules are defined in two new tables: GPOS for moves (Glyph POSitioning), and GSUB for substitutions (Glyph SUBstitution).

GPOS and GSUB abilities include very general modifications like contextual substitution and positioning (e.g., move the second dot in "S.A.V.").

Every feature is identified by a four-letter tag.

Features

For a better handling of the Western scripts, and for complex scripts, OpenType introduces the concept of *features*.

A feature is a set of rules for moving or replacing glyphs or groups or glyphs by others.

These rules are defined in two new tables: `GPOS` for moves (Glyph POSitioning), and `GSUB` for substitutions (Glyph SUBstitution).

`GPOS` and `GSUB` abilities include very general modifications like contextual substitution and positioning (e.g., move the second dot in "S.A.V.").

Every feature is identified by a four-letter tag.

The "traditional" functionalities of fonts are reimplemented in term of lookup: ligatures are associated to the feature `liga`, and kerning to the feature `kern`.

Features

For a better handling of the Western scripts, and for complex scripts, OpenType introduces the concept of *features*.

A feature is a set of rules for moving or replacing glyphs or groups or glyphs by others.

These rules are defined in two new tables: `GPOS` for moves (Glyph POSitioning), and `GSUB` for substitutions (Glyph SUBstitution).

`GPOS` and `GSUB` abilities include very general modifications like contextual substitution and positioning (e.g., move the second dot in "S.A.V.").

Every feature is identified by a four-letter tag.

The "traditional" functionalities of fonts are reimplemented in term of lookup: ligatures are associated to the feature `liga`, and kerning to the feature `kern`.

Both these features are activated by default by the rendering engine; other features can be deactivated by default.

A feature sampler

Discretionary ligatures (dlig):

A feature sampler

Discretionary ligatures (dlig):

. *Ein Teil von jener Kraft,
die stets das Böse will,*

A feature sampler

Discretionary ligatures (dlig):

. *Ein Teil von jener Kraft,
die stets das Böse will, und stets das Gute schafft.*

A feature sampler

Discretionary ligatures (dlig):

. *Ein Teil von jener Kraft,
die stets das Böse will, und stets das Gute schafft.*

Small capitals (smcp):

I like UNICODE.

A feature sampler

Discretionary ligatures (dlig):

. *Ein Teil von jener Kraft,
die stets das Böse will, und stets das Gute schafft.*

Small capitals (smcp):

I like UNICODE.

Old numerals (onum):

1729.

A feature sampler

Discretionary ligatures (dlig):

. *Ein Teil von jener Kraft,
die stets das Böse will, und stets das Gute schafft.*

Small capitals (smcp):

I like UNICODE.

Old numerals (onum):

1729.

Superscripts (sup):

They can be used to write m^2 in text mode.

A feature sampler

Discretionary ligatures (dlig):

. *Ein Teil von jener Kraft,
die stets das Böse will, und stets das Gute schafft.*

Small capitals (smcp):

I like UNICODE.

Old numerals (onum):

1729.

Superscripts (sup):

They can be used to write m^2 in text mode.

Here is what happens if we activate it to early:

A feature sampler

Discretionary ligatures (dlig):

. *Ein Teil von jener Kraft,
die stets das Böse will, und stets das Gute schafft.*

Small capitals (smcp):

I like UNICODE.

Old numerals (onum):

1729.

Superscripts (sup):

They can be used to write m^2 in text mode.

Here is what happens if we activate it to early:

The^y can be u^{sed} to w^{rite} m² in te^xt mode.

Scripts and languages

Each feature is defined for a specific *script* and *language*: different languages can have a different handling of the same feature.

Scripts and languages

Each feature is defined for a specific *script* and *language*: different languages can have a different handling of the same feature.

There is a default language for each script, but no default script for the font.

Scripts and languages

Each feature is defined for a specific *script* and *language*: different languages can have a different handling of the same feature.

There is a default language for each script, but no default script for the font.

Scripts are tagged with four lowercase letters, and languages with three uppercase letters (e.g., 'latn', 'cyr', 'deva'; 'ENG', 'FRE', 'TRK').

Scripts and languages

Each feature is defined for a specific *script* and *language*: different languages can have a different handling of the same feature.

There is a default language for each script, but no default script for the font.

Scripts are tagged with four lowercase letters, and languages with three uppercase letters (e.g., 'latn', 'cyr', 'deva'; 'ENG', 'FRE', 'TRK').

Example: in Turkish, there are two different characters 'ı' and 'i' without dot, so the uppercase for 'ı' is 'I', and the small capitals feature yields a different result:

Scripts and languages

Each feature is defined for a specific *script* and *language*: different languages can have a different handling of the same feature.

There is a default language for each script, but no default script for the font.

Scripts are tagged with four lowercase letters, and languages with three uppercase letters (e.g., 'latn', 'cyr', 'deva'; 'ENG', 'FRE', 'TRK').

Example: in Turkish, there are two different characters 'ı' and 'i' without dot, so the uppercase for 'ı' is 'I', and the small capitals feature yields a different result:

I like UNICODE.

Scripts and languages

Each feature is defined for a specific *script* and *language*: different languages can have a different handling of the same feature.

There is a default language for each script, but no default script for the font.

Scripts are tagged with four lowercase letters, and languages with three uppercase letters (e.g., 'latn', 'cyr', 'deva'; 'ENG', 'FRE', 'TRK').

Example: in Turkish, there are two different characters 'ı' and 'i' without dot, so the uppercase for 'ı' is 'I', and the small capitals feature yields a different result:

I like UNICODE.

The fonts only knows how to place the glyphs individually, it does not know about the properties of the script itself; this is the responsibility of the rendering engine.

Scripts and languages

Each feature is defined for a specific *script* and *language*: different languages can have a different handling of the same feature.

There is a default language for each script, but no default script for the font.

Scripts are tagged with four lowercase letters, and languages with three uppercase letters (e.g., 'latn', 'cyr', 'deva'; 'ENG', 'FRE', 'TRK').

Example: in Turkish, there are two different characters 'ı' and 'i' without dot, so the uppercase for 'ı' is 'I', and the small capitals feature yields a different result:

I like UNICODE.

The fonts only knows how to place the glyphs individually, it does not know about the properties of the script itself; this is the responsibility of the rendering engine.

⇒ In an Arabic font is written nowhere that the text has to be set right to left!

Other TrueType extensions

Apple uses its own extension to the TrueType format, called AAT with similar principles, but the implementation is done using finite automata (*potentially* more powerful).

Other TrueType extensions

Apple uses its own extension to the TrueType format, called AAT with similar principles, but the implementation is done using finite automata (*potentially* more powerful).

SIL International has developed another extension called Graphite.

Other TrueType extensions

Apple uses its own extension to the TrueType format, called AAT with similar principles, but the implementation is done using finite automata (*potentially* more powerful).

SIL International has developed another extension called Graphite.

Both these extensions only allow quadratic splines to be used in the glyphs description (TrueType glyph description).