

Version 7.0.1

Released 21st June 2013

FontCreator 7.0 is a major new version, so owners of earlier versions will need to pay for an upgrade. The latest version adds several long-awaited features that will make this a must-have upgrade for keen font designers. Web Designers may be tempted to create their own fonts now that FontCreator supports the Web Open Font Format (WOFF). [Test Page](#) (View in Firefox)

Professional users who need the high-end features (highlighted by **), and Home users too, will appreciate the improvements to their fonts that are now possible with automatic hinting. All users will benefit from the improved overview, extra features, bug fixes, and simplified interface.

Free help is available on the active support forum, even for the unregistered 30-day trial version. The glyph overview, the menus and several dialogues have been removed or reorganised so please take the time to study the interface changes before asking, "Where is the Format Menu?" Or "What happened to the Platform Manager?"

[The help file](#) is available online too, so you can get a good idea about what features are available before downloading the free trial version.

The most recent minor update adds a few more refinements:

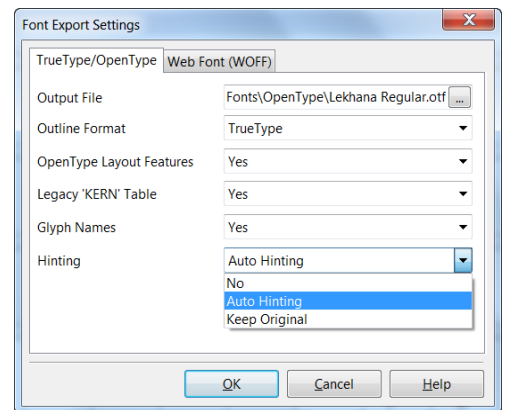
OpenType layout features are decompiled automatically on opening a font.

- New script function added to the layout feature editor.
- Added support for multiple substitution that allows ligatures to be separated into single glyphs.
- Added support for multiple backtrack, input and lookahead groups for chained context lookups.
- Added new keyword "ignore" for chained context lookups.
- Added support for multiple substitution tables.

Project Files

FontCreator now uses Project files instead of working only with TrueType or OpenType fonts directly. Each project can have its own set of ruler guides, notes, OpenType feature script, and export settings. You can open TrueType or OpenType fonts as before, then save them as a project file before starting work, leaving the original files untouched. Select the desired Export Options before exporting your project as a TTF, OTF, or WOFF font file.

Use Export from the File menu to export the project to a TrueType or OpenType font or to a Web Font, choosing whether to include hinting and/or OpenType layout features. The default feature script includes some common features, but custom scripts can be saved with the project and used to add more features. See the next section on adding OpenType features. Exporting a font only takes a few seconds.



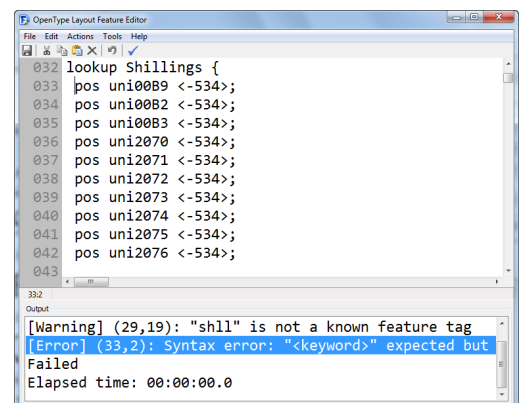
OpenType Feature Support**

Users of PagePlus and DrawPlus who read this review may already be familiar with using OpenType features, because they have been supported since PagePlus X5 was released in October 2010. Until now, adding OpenType features to fonts created with FontCreator required third-party tools like VOLT or the [OpenType Compiler](#). This was beyond the ability of many users.

FontCreator 7.0 Professional and Standard Editions have built-in support for scripts to add many (but not all) OpenType features. The script editor has line-numbering, and can be customised to use a font and colours of your choice. The font size can also be zoomed directly using the mouse scroll-wheel.

The Professional Edition (but not the Standard Edition) can extract scripts from other fonts. This first release does not support all [glyph substitutions](#) yet. The compiler will generate warnings or error messages, as shown above when compiling a script from [Linux Libertine](#). Double-click the error message to go to the line, to debug the error. A comment marker (#) before the problematic line lets the script compile successfully.

Although there is a lot to learn about OpenType features, once you have created a script to suit your fonts, it's a simple process to build an OpenType font from a TrueType version. If there are any syntax errors in compiling a script, the messages will help you to debug your script or fix the glyph names in your fonts to match the script. As with all coding, syntax has to be 100% correct — miss a semicolon, or use the wrong capitalisation, and the script will fail to run. Below are a few examples of features you



might like to add. My scripts use meaningful glyph names to make them easier to read and edit — they will need editing if you want to use the glyph names generated by FontCreator. <Omit the comments in angled brackets>.

Standard Ligatures (Not editable in the Home Edition**)

```
feature Ligatures liga { <Feature name and OpenType layout tag "liga".>
lookup ligaSub; <Lookup table to use for this feature.>
} lookup ligaSub { <Feature definition is {within braces}, lookup table name, { begins lookup}>
  sub f f i -> ffi; <If user types f, f, i, then substitute ligature ffi.>
  sub f f l -> ffl; <"f" and "l" are the glyph names for lowercase "f" and "l".>
  sub f f -> ff; <"ff" is the glyph name for the ligature "ff".>
  sub f i -> fi; <If user types f, i, then substitute ligature fi.>
  sub f l -> fl; <If user types f, l, then substitute ligature fl.>
} <End of lookup table. Lookup table is within {braces}.>
```

Scientific Inferiors**

```
feature ScientificInferiors sinf { <Feature definition, layout tag>
lookup ScientificInferiors;
}
lookup ScientificInferiors {
  sub zero -> zero.sinf; <If user types 0 substitute a scientific inferior zero.>
  sub one -> one.sinf; <"one.sinf" is the glyph name of the glyph to substitute.>
  sub two -> two.sinf; <"two" is the default glyph name for digit 2.>
  sub three -> three.sinf; <".sinf" is the name suffix I chose for scientific inferiors.>
  sub four -> four.sinf; <There is no rule that says one must use the feature tag as a suffix.>
  sub five -> five.sinf; <It is just a convention that makes the script easier to understand.>
  sub six -> six.sinf;
  sub seven -> seven.sinf;
  sub eight -> eight.sinf;
  sub nine -> nine.sinf;
  sub plus -> plus.sinf;
  sub minus -> minus.sinf;
  sub equal -> equals.sinf;
  sub parenleft -> parenleft.sinf;
  sub parenright -> parenright.sinf; <Substitute a subscripted right parenthesis.>
}
}
```

Ordinals**

```
feature Ordinals ordn { <Feature name and layout tag.>
lookup Ordinals;
}
group @Ordinals [zero one two three four five six seven eight nine A.sups B.sups C.sups D.sups
E.sups F.sups G.sups H.sups I.sups J.sups K.sups L.sups M.sups N.sups O.sups P.sups Q.sups R.sups
S.sups T.sups U.sups V.sups W.sups X.sups Y.sups Z.sups a.sups b.sups c.sups d.sups e.sups f.sups
g.sups h.sups i.sups j.sups k.sups l.sups m.sups n.sups o.sups p.sups q.sups r.sups s.sups t.sups u.sups
v.sups w.sups x.sups y.sups z.sups]; <Glyphs to be followed by ordinals.>
group @Alphas [A B C D E F G H I J K L M N O P Q R S T U V W X Y Z a b c d e f g h i j k l m n o p q
r s t u v w x y z]; <Glyphs to be used as ordinals.>
lookup Ordinals{
  context (@Ordinals) @Alphas; <If alpha glyph follows a glyph in ordinals, then ...>
  sub 0 Super; <In the above context, use this lookup table.>
} lookup Super {
  sub A -> A.sups; <If user types A after a digit or after an ordinal substitute superscript A.>
  ... <Repeat sub line for A through to Z and a to z, and more if required, e.g. è ú etc.>
  sub Z -> Z.sups;
  sub a -> a.sups;
  sub z -> z.sups;
}
}
```

This contextual lookup Ordinals feature first checks to see if a digit has been typed. If it has, then it checks to see if an alpha character is typed next. If not it does nothing, so 123 remains as usual. However if the user types 1s, the Ordinals feature superscripts the s. If the user continues to type "t" or any other alpha characters, they are all superscripted. However, as soon as the user types a space, or some punctuation, any following characters are normal again.

Ordinals in Adobe fonts (e.g. SourceSansPro) and Microsoft fonts (e.g. Calibri) are implemented differently. If the user types a digit or lowercase alpha character and the Ordinals feature is enabled, all are substituted with superscripts. You can decide how you want to implement this feature, but I recommend the former method because it means that the ordinals feature can be enabled as part of the paragraph style for body text, and then ordinals are automatic.

Stylistic Sets**

```
feature StylisticAlternates salt { <Feature name and tag.>
  lookup StylisticAlternates;
} lookup StylisticAlternates {
  sub asterisk -> [asterisk asteriskmath uni2051 uni2042 uni203B];
  sub plus -> [plus uni2795 uni271A uni271B uni271C];
  sub at -> [at uni260E uni260F uni2121 uni213B];
  sub copyright -> [copyright uni2117 uniF000 estimated uni2139];
  sub multiply -> [multiply uni2715 uni2716 uni2717 uni2718];
  sub dagger -> [dagger uni2622 uni2623 uni26A1 uni2620];
  sub daggerdbl -> [daggerdbl uni26A0 uni2691 uni26B0 uni26B1];
  sub bullet -> [bullet uni204D uni2023 uni2767 uni2712];
  sub circle -> [circle fullmoon waninggibbousmoon lastquartermoon waningcrescentmoon];
  sub blackcircle -> [blackcircle newmoon waxingcrescentmoon firstquartermoon waxinggibbous-
moon];
  sub uni2600 -> [uni2600 sunrise uni26C5 uni2602 uni2603];
}
```

This feature requires intervention by the user. After selecting an asterisk, the user can choose from any of the five symbols from asteriskmath to Unicode character 273B, which is a Dingbat. If the user selects a plus sign they can choose from a different set of five cross-like symbols.

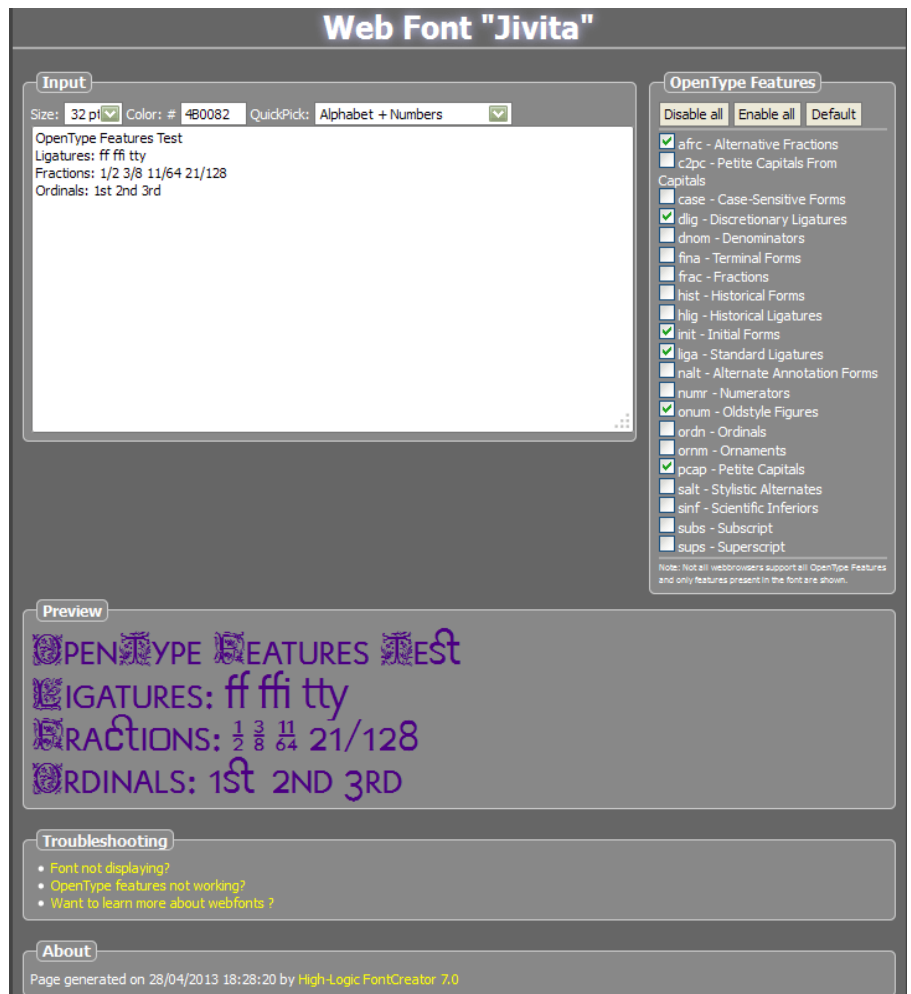
For each "sub" line the first glyph is repeated, which is Stylistic Set 0 (the original glyph), followed by five alternative glyphs for the user to choose from. You may add up to 99 Stylistic sets.

Web Font Support

Web Fonts (WOFF format) are a compressed TrueType font format designed for embedding on web pages. They allow web designers the freedom to use fonts that are not Websafe since the fonts don't need to be installed on the site visitors' computers.

FontCreator 7 supports the WOFF format, and shows a preview of the font in your default web browser. If your browser supports OpenType features in CSS (Internet Explorer E10, Chrome12 or later, FF4 or later, and the Latest version of Safari), then this local HTML preview page can be used to test OpenType features without actually installing the fonts. You can quickly check that the features are working as intended, before finalising the OpenType features script and installing the OpenType version of the font for testing in Windows applications.

This preview demonstrates some of the features in my own font Jivita Regular. By checking and unchecking the OpenType Features in the panel on the right, you can see how the features interact. For example, if fractions are enabled, Ordinals won't work as expected, or Discretionary Ligatures will override Petite Capitals. The input text can be modified by typing (or by editing wofftest.js at your own risk), and the size changed from the font size drop list.



The screenshot shows a web browser interface for testing the 'Web Font "Jivita"'. It features an 'Input' section with a font size of 32pt, color #4B0082, and a 'QuickPick' dropdown set to 'Alphabet + Numbers'. Below this is a preview of the font with OpenType features enabled, showing 'OPEN TYPE FEATURES TEST', 'LIGATURES: ff ffi tty', 'FRACTIONS: 1/2 3/8 11/64 21/128', and 'ORDINALS: 1ST 2ND 3RD'. To the right is an 'OpenType Features' panel with checkboxes for various features like 'Alternative Fractions', 'Discretionary Ligatures', 'Initial Forms', etc. A 'Troubleshooting' section at the bottom provides links for 'Font not displaying?', 'OpenType features not working?', and 'Want to learn more about webfonts?'. An 'About' section at the very bottom states 'Page generated on 28/04/2013 18:28:20 by High-Logic FontCreator 7.0'.

text can be changed, and some preset text samples can be selected from the Quick Pick drop list. Not all OpenType features are supported by CSS, but the preview is enough to test most common features.

FontCreator 7 can also open existing Web Fonts for editing. Use the Professional Edition to add OpenType features.

Automatic Hinting

Select "Auto hinting" or "Keep original" on exporting fonts. Automatic or none, will remove original hints. Now your fonts will look better on screen at small sizes, without the spindly appearance cause by "dropout" where thin strokes disappear. Except for complex or large fonts, it takes only a few seconds, but it does significantly increase the size of font files (in my tests by about 16-25%).

Optical Metrics**

New to FontCreator 7, is the option to generate side-bearings for the font automatically. This may take some time on slower computers as a lot of processing is involved, but it only needs to be done once after editing the glyph outlines. Getting character spacing just right is one of the most time-consuming and difficult tasks when designing fonts. Kerning should only be done after finalising the side-bearings.

When generating Optical Metrics, choose from Tabular Figures or Proportional Figures. Most fonts have Tabular figures (fixed width digits) for even alignment of figures in tables. A few fonts like Georgia have proportional figures. OpenType fonts may have both, and/or OldStyle figures. FontCreator only calculates Optical Metrics for the default digits. Copy metrics to additional glyphs using Paste Special (Ctrl Alt V).

Professional font designers who are importing vector images into FontCreator from Adobe Illustrator or CorelDraw will find it well worth the extra price over the Standard Edition just for the time that this saves. Since they will have faster computers than mine, it should take only about 15 minutes to process a font. The default setting of 27 seems to give rather too much spacing in my limited tests. Try a value of 24 or 23 and see if that is better. Optical metrics are added to the undo history, so you can try progressively smaller or larger values before deciding on the value that suits your font.

American Garamond Bitstream's Spacing • Optical Metrics (value = 24)



Other fonts may need different values. I haven't tested it much yet as it takes nearly 10 minutes to run on my PC.

Overview Panel

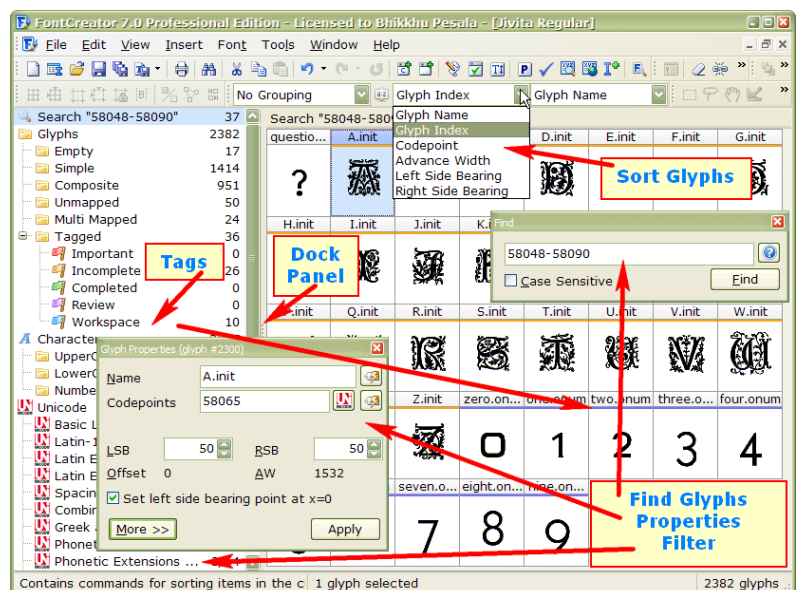
The glyph overview has been completely redesigned. A resizable panel on the left of the overview allows fonts to be filtered by glyph type or character set. Clicking the thumb button on the splitter will dock the panel when its not in use, and shortcut Ctrl L will show it again. Shortcuts Ctrl J and Ctrl K will display code-points in Decimal or Hex.

The Tab key will focus the panel even if it is closed, then the cursor keys can be used to display the previous or next character set. Glyphs can be grouped by Unicode blocks or by width, and sorted by glyph index, name, or width in ascending or descending order using the buttons on the Overview's toolbar.

The Find function will locate a single glyph using its mapping or glyph name, or display a range of glyphs from their mappings, e.g. Find "58048-58090" will display the glyphs as in the screen shot. These are the mappings that I use for OldStyle numerals, Case-sensitive Forms, and Initial Capitals.

One of five tags will underline the glyph caption in a different colour. Right-click on a glyph in the overview to access the tag submenu.

Zoom in or out using shortcut keys (Numpad +/-) or the font size drop list on the toolbar. Sort by name, index, advance width, or side-bearings, in ascending or descending order.



Notes Toolbar

A new Notes Toolbar (Shortcut F2) is a convenient place to record changes made, and list things to be done for each font project.

Glyph Properties Toolbar

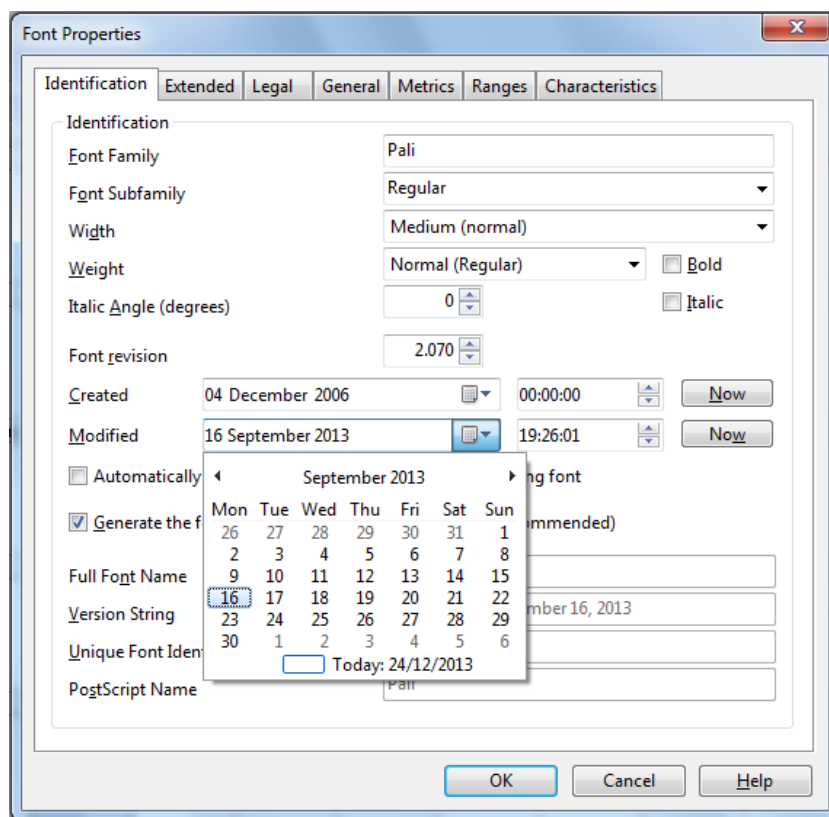
This is new, and replaces the Glyph Properties dialogue. Show/hide it with the F3 Shortcut. Use the buttons to set the glyph name from the glyph mapping code-point, or to set the code-point from the glyph name.

 For a new blank glyph, use the "Select Unicode Character" button to select a code-point from a list of Unicode Character Names for each Unicode Character set.

Glyph names are generated on opening fonts using a file named "glyphnames.dat" in the Application Data FontCreator folder. Edit this plain text file if you want FontCreator to generate different names. Each line in the text file takes the form "Hex code;glyphname" e.g. "FB04;f_f_l" which you can edit, e.g. to "FB04;ffl" or add your own lines, e.g. "1F312;waxingcrescentmoon" etc.

Interface Changes

Some obsolete items have been removed and items on the Format menu has been moved. The Font Properties dialogue replaces the Format Settings dialogue. The font information dialogue replaces the previous Font Properties dialogue.



Key Changes:

- The Platform Manager has been removed. "Convert to Symbol Font" or "Convert to Unicode Font" on the Tools menu will update the required platforms automatically.
- The PCL Dialogue has been removed.
- Font Naming has been moved to the Font Properties Dialogue, accessible from the Font menu.
- The Postscript Names Dialogue has been removed. Glyph names can be set automatically from the code-point by clicking the icon on the Glyph Properties toolbar.
- The Glyph Mappings (code-points) can be set automatically from the glyph name by clicking the icon on the Glyph Properties Toolbar. Find Unicode Names for each character set by clicking the "Select character" icon on the Glyph Properties toolbar.
- The Format Menu has been removed. Kerning and Smoothing are now on the Font Menu.
- Format, Settings, was moved to Properties on the Font Menu.
- Font Properties were moved to the Font Information Dialogue accessible from the Font Menu.
- Sort Glyphs has been moved from the Font Menu to a submenu of the Tools Menu.
- Font Naming Fields are now on the Identification page of the Font Properties dialogue.
- Advanced Naming Fields are on the Legal page of the Font Properties dialogue.
- Contents and Layout and Header Flags are on the General page of the Font Properties dialogue.
- PANOSE classification is on the Characteristics page of the Font Properties dialogue.