Using Glyph Transformations

In High-Logic® FontCreator Professional

The transform feature is useful for making new type styles from existing fonts. It is not available in the Home Edition, but only in the Standard or Professional Editions or in the thirty-day Free Trial. Please see this » Comparison Chart « and the » Registration Page « for details.

General Advice for Working With Transformations

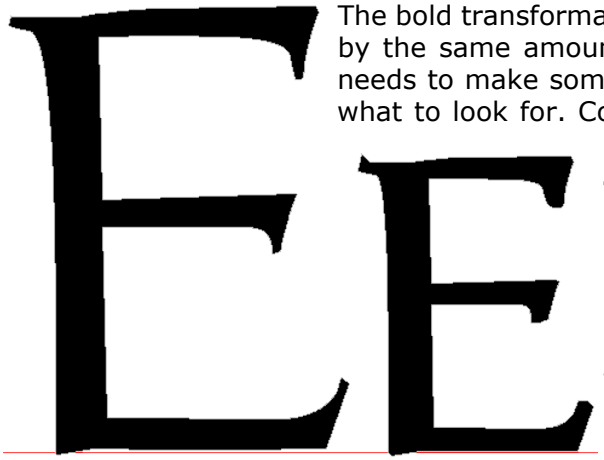
Because transformations cannot be undone, copy the glyphs that are going to be transformed, and leave them selected. After running the transformation, paste the clipboard contents back into the same glyphs, and undo the paste operation. The paste operation can then be undone to try different values.

From the Tools menu, select Glyph Transformer and from the Transform Wizard dialogue click the folder icon to open one of several predefined scripts. User-defined scripts can be created by adding any of the available features from the left panel, and saved for later reuse.

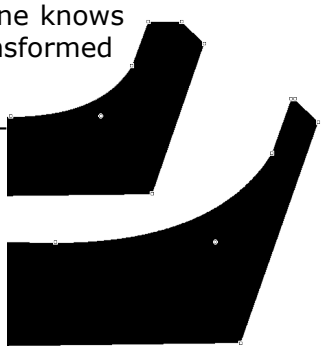
1. Small Capitals

Before running this transformation, copy the Capital Letters of the font to the lowercase positions, and select them. The Small Capitals transformation scales the uppercase by about 75%. To compensate for the scaling the stroke weight needs to be increased with a bold transformation. To add Small Capitals for OpenType Features use the Small Capitals Private Use script or the Petite Capitals script.

Script: Small Capitals.xml
Scale (75.00, 78.00) (0, 0)
Bold (20,12) preserve bearings
Move (0,12)

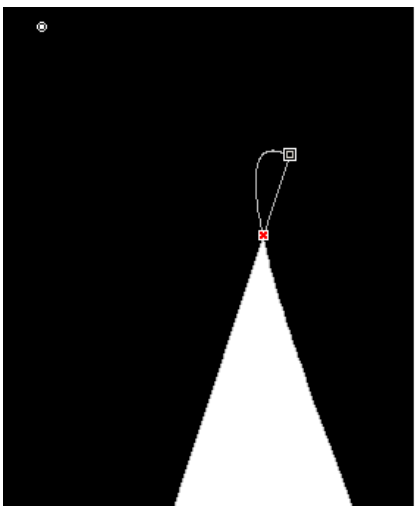


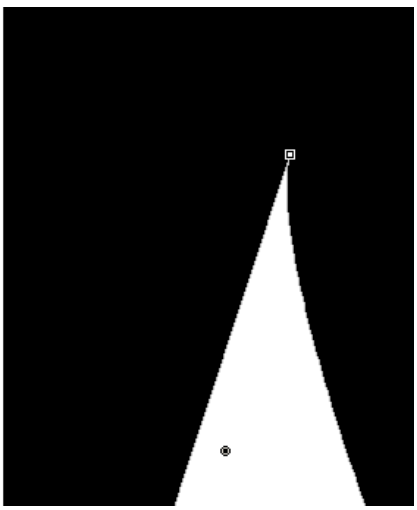
The bold transformation is not perfect. Thin strokes and serifs are made thicker by the same amount as thick strokes. After applying the transformation one needs to make some corrections. This is easy when one knows what to look for. Compare the original E and the transformed result.



The Serifs Are Too Thick — Zoom in close to see what happened. All one needs to do is select a few nodes with the lasso selection tool and move them to the right using the cursor keys. The serifs have been made 40 funits thicker when they only need to increase by about 10 funits.

Intersecting Co-ordinates — Sometimes a node crosses the contour giving a result like that shown below (Capital A). Just move or delete the offending node to correct this problem.

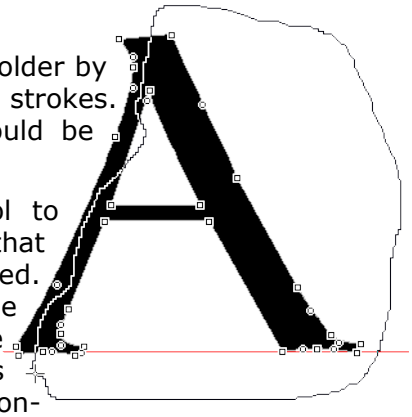




Contents

Thin Strokes

Thin strokes are made bolder by the same amount as thick strokes. Ideally, their weight should be proportional.



Use the lasso tool to select the nodes that need to be moved. Move them to the left until the stroke weights are proportional. Reduce the advance width to match.

2. OldStyle Figures

Old style figures or non-lining figures are found by default in some fonts like Georgia or Constantia, but most fonts have lining figures, which are designed for use with Capitals. Non-lining figures are designed for use with body text, with the digits being designed on the x-height.

0123456789

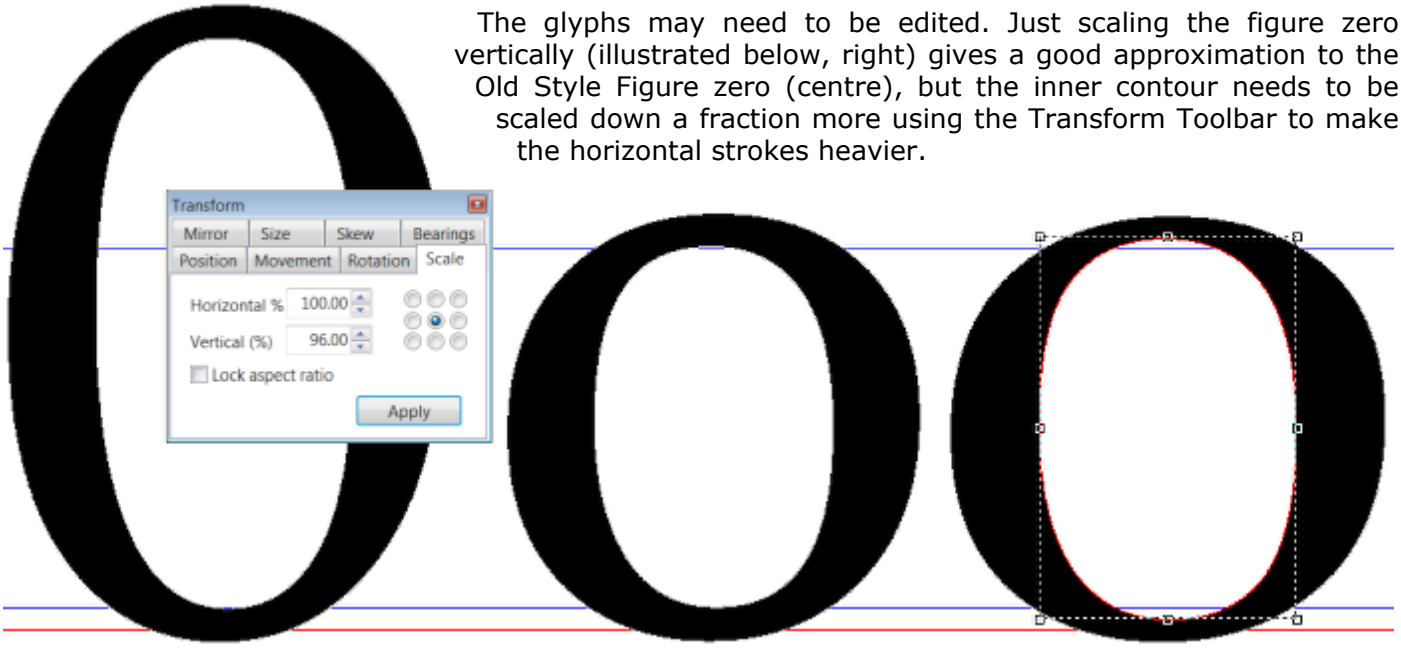
0123456789

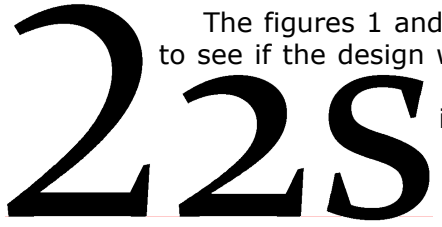
Fonts that have Old Style Figures by default may need lining figures for use with capitals. A similar script could be used to insert lining figures in the Private Use Area, scaling up the small figures, and moving the figures with descenders up to the baseline.

Insert Characters 58048-58050 (Apply subsequent features)
Complete Composites
Decompose
Scale (100.00,75.00) Bottom Centre
Insert Characters 58051-58053, 58055, 58057 (Apply subsequent features)
Position (-,925) Left Top
Insert Characters 58054, 58056 (Apply subsequent features)
Complete Composites
Decompose

This script inserts Old Style Figures in the Private Use Area. It first inserts the figures 0, 1, and 2, and scales them by 75% vertically to reduce them (approximately) to the x-height. Adjust the vertical scale value to suit the x-height of the font. Then it inserts the figures 3, 4, 5, 7, and 9, and moves them vertically so that the top left corner of each glyph is just above the x-height, to allow for the overshoot. The design of the 4 and 7 glyphs may or may not require any overshoot. Finally, it inserts the glyphs for 6 and 8, which are usually the same design as for lining figures.

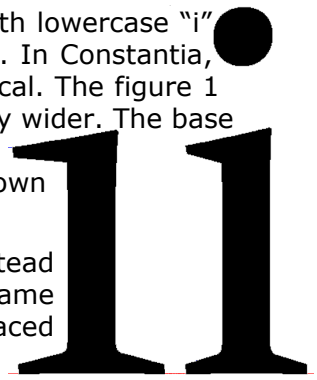
The glyphs may need to be edited. Just scaling the figure zero vertically (illustrated below, right) gives a good approximation to the Old Style Figure zero (centre), but the inner contour needs to be scaled down a fraction more using the Transform Toolbar to make the horizontal strokes heavier.





The figures 1 and 2 will need similar adjustments. Compare 1 with lowercase "i" to see if the design will be right when numbers are mixed with text. In Constantia, the figure zero and the lowercase "o" are identical. The figure 1 is similar to the dotless "i" but the base is slightly wider. The base of the figure two is the same as that of the regular figure 2, but the upper curve is scaled down vertically to fit the x-height.

The figure 4 has no foot since it is hanging below the baseline instead of standing on it. The other figures — 3, 5, 7, 8, and 9 — have the same design as regular figures, but are just positioned and/or spaced differently.



3. Petite Capitals Private Use

Small Capitals may replace the lowercase letters in a font to create a dedicated Small Caps font, or they may use the OpenType Small Capitals feature with glyph substitution to replace lowercase letters. The small capitals are similar in proportion to the uppercase, and between 70% and 80% of the height. The stroke weights are increased to compensate for the scaling.

Petite Capitals are designed to be substituted for lowercase letters when the Small Capitals attribute is applied in applications. They may be proportionally wider than the uppercase and should match the x-height of the lowercase. In OpenType fonts they are accessed by using glyph substitution However, since most applications don't currently support OpenType features, one should assign them to code-points in the Private Use Area. This is essential to generate them in FontCreator.

Small Capitals are not part of the Unicode standard. There are some small capitals in the Unicode charts, but these are IPA phonetic glyphs, not typographical variants. CompositeData.xml uses decimal code-points, so the following convenient ranges are used for Petite/Small Capital Assignments:

58033-58126	Basic Latin
58160-58255	Latin-1 Supplement
58256-58383	Latin Extended-A
58384-58591	Latin Extended-B
58912-58974	Basic Greek
59072-59218	Cyrillic
59680-59829	Latin Extended Additional
60531-60542	Number Forms, Nut Fractions or Stacking Fractions
61124-61392	Discretionary Ligatures

Lowercase a, code-point 97, corresponds to Small Capital A, code-point 58097. Uppercase A, (65), corresponds to Titling Capital A (58065). Zero (48) corresponds to Old Style Figure zero (58048), etc.

Not every glyph with a Unicode code-point is defined. Users with special requirements will need to edit CompositeData.xml and the Transform scripts to add data to compose other characters. The newly created Petite Capitals will need some editing to remove intersecting co-ordinates, some diacritics will need to be aligned, and "Get Union of Contours" will need to be used for glyphs like Æ, Ð, and Þ.

Script: Petite Capitals.xml
Insert Characters 58033, 58036-58038, 58040, 58041, 58063, 58091, 58093, 58097-58123, 58125, 58161, 58191, 58223, 58331, 58339 (Apply subsequent features)
Complete Composites
Decompose
Scale (61.00, 58.00) (0, 0)
Bold (12, 6) preserve bearings
Move (0, 6)
Left Side Bearing Point at x=0
Insert Characters 58224-58246, 58248-58255, 58257, 58259, 58261, 58263, 58265, 58267, 58269, 58271, 58273, 58275, 58277, 58279, 58281, 58283, 58285, 58287, 58289, 58291, 58293, 58295, 58297, 58299, 58301, 58303, 58307, 58309, 58311, 58314, 58316, 58318, 58320, 58322, 58324, 58326, 58328, 58333, 58335, 58337, 58341, 58343, 58345, 58347, 58349, 58351, 58353, 58355, 58357, 58359, 58361, 58363, 58365, 58367, 58369, 58371, 58373, 58375, 58378, 58380, 58382, 58501, 58507, 58509, 58511, 58537, 58539, 59693, 59717, 59735, 59737, 59745, 59747, 59749, 59751, 59771, 59773, 59779, 59789, 59809, 59811, 59813, 59923, 59929
(Apply subsequent features)
Complete Composites

4. Titling Capitals

Titling capitals are designed for use at larger point sizes, where regular capitals are too heavy. The easiest way to design these glyphs is to retain the same vertical size, and the same advance width, as regular capitals, but just to reduce the stem weight by about 20 funits.

Script: Titling Capitals.xml
Insert Characters 58065-58090, 58198, 58222, 58260, 58272, 58280, 58294, 58302, 58321, 58330, 58338, 58350, 58354, 58358, 58370 (Apply subsequent features)
Complete Composites
Decompose
Thin 10,0
Bearings LS increase 10, RS increase 10
Insert Characters 58192-58214, 58217- 58222, 58256, 58258, 58262, 58264, 58266, 58268, 58270, 58274, 58276, 58278, 58282, 58284, 58286, 58288, 58290, 58292, 58296, 58298, 58300, 58304, 58306, 58308, 58310, 58313, 58315, 58317, 58319, 58323, 58325, 58327, 58332, 58334, 58336, 58340, 58342, 58344, 58346, 58348, 58352, 58356, 58360, 58362, 58364, 58366, 58368, 58372, 58374, 58376, 58377, 58379, 58381, 58500, 58506, 58508, 58510, 58536, 58538, 59692, 59716, 59734, 59736, 59744, 59746, 59748, 59750, 59770, 59772, 59778, 59788, 59808, 59810, 59812, 59922, 59928 (Apply subsequent features)
Complete Composites

This script inserts Titling Capitals in the Private Use Area, reduces the stroke weight by 10 funits either side, then increases the side-bearings by 10 funits to compensate, thus retaining the same advance width. It then inserts more characters for the Latin Extended glyphs, and generates composites. Serifs, or any other thin vertical strokes, will need adjusting afterwards if they are too thin. Since no adjustment is made to horizontal strokes, the CapsHeight is unchanged, and accents should be in the correct vertical position. Some accents may need horizontal alignment.


This script could be changed to add Small Capitals if Petite Capitals are also needed.

5. Italic and Oblique

These transformations work well, though they can only make an oblique version of a font, not a true italic, which requires cursive lowercase letters. Uppercase letters and figures usually look fine, though a few require different designs. Maths symbols can be slanted or upright in italic fonts. There is no hard and fast rule. Other glyphs such as the notdef glyph, vertical line © ¤ ± × ® ™ and geometric shapes will also look better if not slanted, though they may need moving to the right.

Before running this transformation, if the font is a hand-written script, set the Family Kind in Format, Settings, Classification. Otherwise, it will be assumed to be Latin Text. Select all of the glyphs to transform, and copy them to the clipboard. After running the script, paste them back in, and undo/redo to see the before and after effects. If the angle is not quite right, change it and run the script again.

When satisfied with the results, choose Save As… from the file menu, and give the italic font a new TTF filename. The font's style is only changed to italic by the script if the option "Set font subfamily and font design to italic" is checked. Otherwise, though the glyphs are slanted, it remains a regular type style. This is what is usually required with a hand-written script, and that is what the script **Oblique.xml** does.

Italic.xml

Italic 11.00 deg, update
Left Side Bearing Point at x=0

After skewing, the glyphs will have a lot of off-curve extremes. These can be removed automatically for the whole font by running the Font Validation Wizard from the font menu, or by checking individual glyphs in the Glyph Edit Window with the Validation Toolbar (F7).

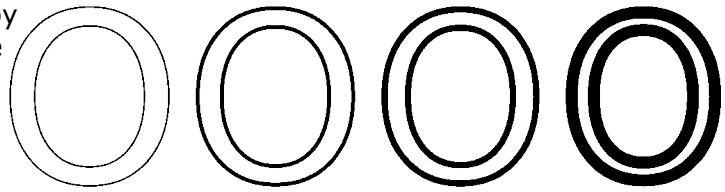
6. Eastern Europe, Extended Greek, and Vietnamese

These scripts insert a wide range of extended characters. Vietnamese requires characters in the Latin Extended Additional range and stacking diacritical marks. Some work will be needed to edit the accents and align them correctly in composites. The » Gentium font « was used to design the Greek script.

Eastern Europe.xml, Basic Greek.xml, Extended Greek.xml, and Vietnamese.xml

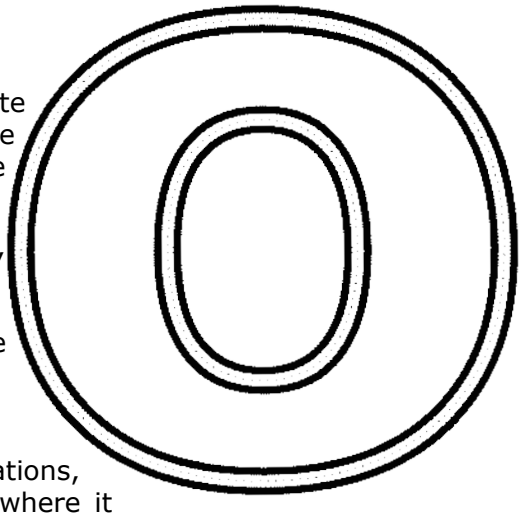
These all simply use the Insert Characers, Complete Composites, and Decompose commands.

7. Outline or Hollow



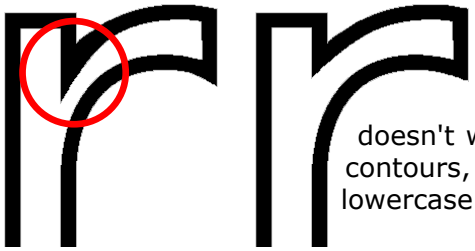
Select all of the glyphs to transform, and copy them to the clipboard. Open one of the Outline scripts: Light, Medium, Bold, or Heavy.

Script: Outline Medium.xml
Scale (97.00, 98.00) (0, 0) preserve bearings
Hollow (30, 30) preserve bearings
Move (0, 15)
Left Side Bearing Point at x=0



Before proceeding, and while the glyphs are still selected, paste the glyphs from the clipboard, and undo the paste operation. If the result is unsatisfactory, redo the paste operation and adjust the transformation.

The hollow transform enlarges the glyph in each direction, and a reversed (white) contour is generated inside it. The dotted lines shows the original contours of the glyph. The outline scripts scale the glyph down to compensate for the size increase from the hollow transformation, while maintaining the advance width.



Problem Glyphs

As with many transformations, there may be some glyphs where it doesn't work perfectly. Wherever there is a sharp change of direction in the contours, some manual adjustment may be needed afterwards as in this lowercase r (illustrated) — before (left) and after manual adjustment.

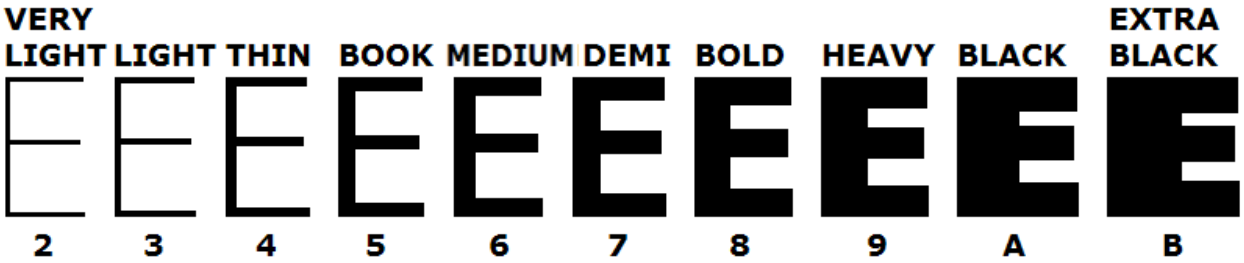
8. Maths Symbol and Symbols

These two are very simple. All they do is set the width and centre the glyph. In most fonts the maths symbols like < = > + ± should be the same width as the figures. Just modify the width value to suit the current font and save the transformation for reuse.

The Symbols transformation is intended for use with Miscellaneous Symbols, Dingbats, or Arrows. Many of these should be the same width, e.g. all of the Chess pieces.

9. Condensed Thin

This transformation scales the glyph horizontally, reduces the stroke weight, and scales the glyph vertically to compensate for the resulting reduction in the glyph height. It was designed to maintain the glyph height, and to reduce the vertical stroke weight from 202 funits to 82. This results in a Panose Classification of "Thin." See Dave Crosby's » Panose Tutorial « for an explanation of the Weight Ratio.



Different fonts need different values for scaling and for reducing the stroke weight. FontCreator now allows scaling to an accuracy of two decimal places.

Script: Condensed Thin.xml
Scale (90.00, 104.93) (0, 0)
Thin (40, 37) preserve bearings
Move (0, -37)

As with the previous transformations, copy the selected glyphs so that paste then undo/redo can be used to try different values until the ideal result is achieved.

Weight	Ratio
V Light	≥ 35
Light	≤ 18 < 35
Thin	≤ 10 < 18
Book	≤ 7.5 < 10
Medium	≤ 5.5 < 7.5
Demi	≤ 4.5 < 5.5
Bold	≤ 3.5 < 4.5
Heavy	≤ 2.5 < 3.5
Black	≤ 2.0 < 2.5
X Black	< 2

10. Bold Transformations

The previous illustration shows the full range of Panose weights from Very Light to Extra Black. When making a bold style from a regular style one needs to consider just how much bolder to make it. To calculate the Panose weight, measure the vertical stem of the uppercase E, and its vertical height. Divide the latter by the former to get the Weight Ratio. Verajja Regular has a capital height of 1493 funits and a vertical stem width of 202 funits, its Panose weight ratio is 7.39, so it is of medium weight, but at the thin end of the range.

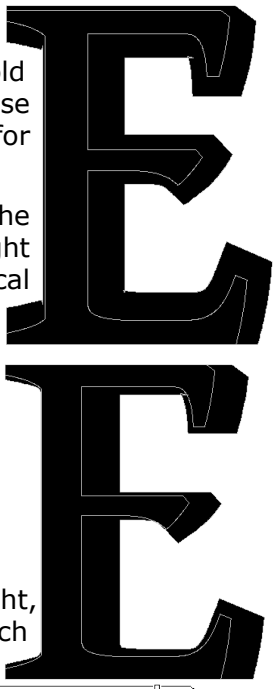
To make a bold style from the regular, open the Medium to Bold transform script and run it on the capital E to see if the result is what is wanted. This gives a stem weight of 350 and a capital height of 1494, so the weight ratio is 4.28, which is at the end of the bold range next to Demi Bold.

Script: Medium to Bold.xml
Scale (92.40, 92.40) (0, 0) preserve bearings
Bold (81, 57) preserve bearings
Move (0, 57)
Left Side Bearing Point at x=0

Verajja Bold has a capital height of 1493 funits and a vertical stem width of 385 funits, its Panose weight ratio is 3.88, so it is of bold weight, towards the heavy end of the range. To make a heavy style from the bold, run the Bold to Heavy transform script. This results in a stem weight of 509 and a capital height of 1493, so the weight ratio is 2.93, which is near the middle of the Heavy range.

The scripts were tested on a font with even contrast and straight lines. When used on fonts with complex curves and sharply contrasting strokes the results may be far from ideal. Do not give up at once. By modifying the script the results can be improved. If that is still not good enough, the thick and thin strokes may have to be separated, using the techniques described in the Thin With High Contrast tutorial.

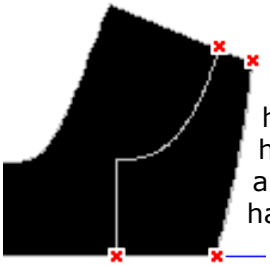
Correcting Bold Transformations



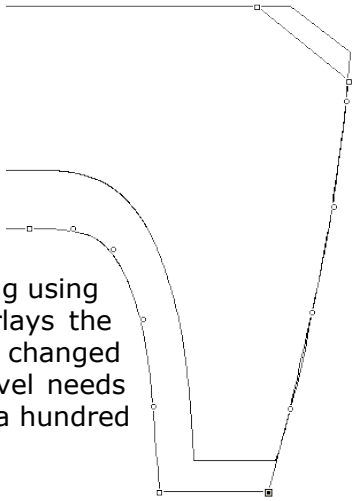
The Medium to Bold script was tested on Gentium, which at the time had no bold type style. The contrast between thick and thin strokes is low, so that should not cause too many problems. However, the vertical /horizontal contrast is greater than for Verajja, so the script needed to be modified.

The initial results are disappointing — the vertical stroke is correct, but the horizontal strokes are too heavy, and the serifs are much too heavy. The capital height has also increased, while it should be the same as the regular type style. The vertical bold effect needs reducing. A vertical bold value of 17 will add 34 funits to the vertical height, so after scaling the height should be 1260 – 34 = 1226 and so the scale factor should be 1226/1260 * 100 = 97.30%

Script: Medium to Bold High Contrast.xml
Scale (97.30, 97.30) (0, 0) preserve bearings
Bold (81, 17) preserve bearings
Move (0, 17)
Left Side Bearing Point at x=0



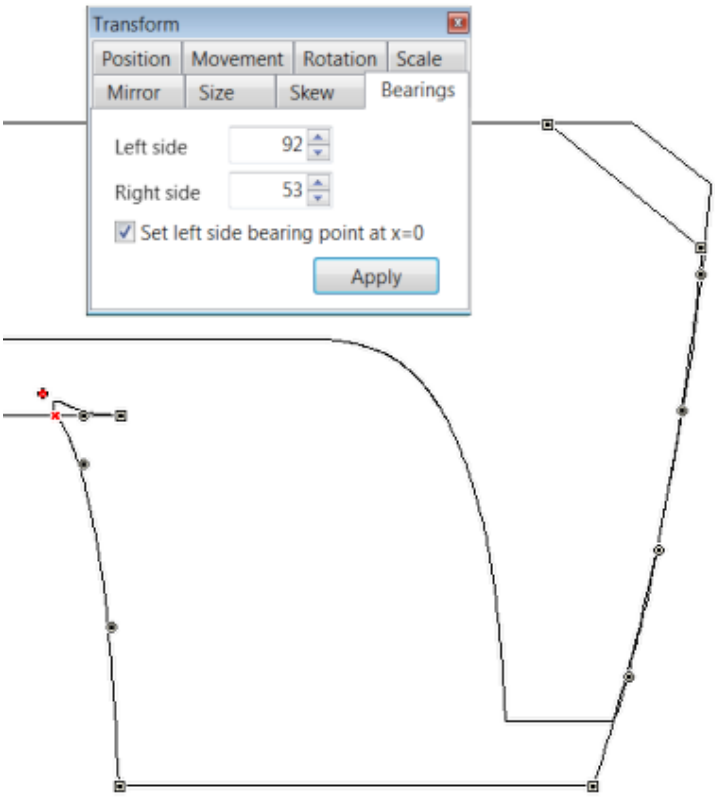
The horizontal strokes are now right, the capital height is right, and the vertical strokes are right. The weight ratio is 3.97, which is in the middle of the bold range. Adjusting the horizontal bold effect is easy if the font is too bold or too heavy. The only problem is the serifs. Unless the serifs are cut off and transformed separately, the nodes will have to be adjusted. Save the script with a new name, and reopen the original font.



Copy the A-Z glyphs, insert 26 new glyphs and paste them in. Run the modified script on these new glyphs, paste the glyphs back again, and undo the paste operation. Undo/redo in the glyph edit window to see the before and after results. Display the original glyph of the copy currently being editing using the Comparison toolbar, and adjust the right side bearing so that it overlays the current glyph. Turn off fill outlines and to see exactly how the serif has been changed by the script. The vertical height of the serif is right, but the top right bevel needs reducing and the bottom left curve needs moving to the right by more than a hundred funits.

Use the lasso tool to select a whole curve section to move it while retaining the shape. Nudge the nodes with the cursor keys plus control or shift as necessary. The result after adjusting the nodes is shown below.

When satisfied with the result, copy the right side-bearing from the original glyph and paste it into the Transform toolbar to reset the advance width.



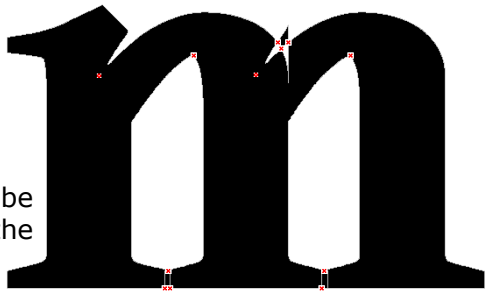
Smooth fonts with fewer nodes are easier to edit. See » this tutorial « for a quick way to reduce the number of nodes.

Having carefully adjusted the serifs on one glyph, cut and paste them to similar glyphs. Use the knife tool to cut the serifs off, copy them to the clipboard, and undo before pasting them into other glyphs. Then go back and weld them on with "Get Union of Contours." The illustration on the left shows the edited serif cut from the bottom of the "E" and pasted into the unedited "L" glyph. They match perfectly, so one just needs to cut off the thick serif and extend the lower arm of the L before joining the edited serif.

Bold Transformations of Lowercase

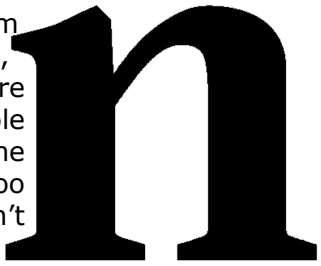
In most fonts the stems of the lowercase are thinner than those of the uppercase. If the same transformation is applied on the whole font, the lowercase, numerals, punctuation, and currency symbols will be too bold. Adjust the script to suit the lowercase. The lowercase "l" is a good choice on which to base the script, but check that the same stem width matches other lowercase glyphs. For Gentium it is 150 funits, compared to 160 funits for uppercase "E." The horizontal strokes are the same, so the script just needs slightly less bold width. Since the ascenders are taller than the capitals, the scale factor also needs adjusting slightly.

Script: Medium to Bold Lowercase.xml
Scale (96.10, 96.10) (0, 0) preserve bearings
Bold (76, 17) preserve bearings
Move (0, 17)
Left Side Bearing Point at x=0



While some glyphs don't need much work, others seem to be beyond salvage. Fix these awkward glyphs by comparing them with the original, then use the lasso tool to select nodes. Select all the nodes on a curve, not individual nodes. Move and undo, then deselect nodes until just those that need to be moved are selected. Bold and regular styles should have a similar x-height. When all editing is finished, copy the right side bearing from the original to adjust the advance width.

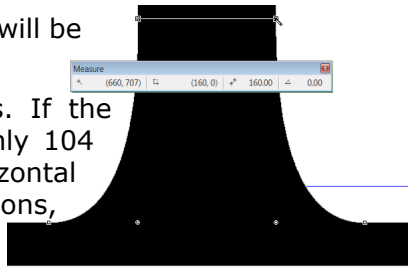
There is no need to duplicate work. The "n" on the right was just copied from the "m" that had already been edited. To delete all of the nodes in the right leg, and adjust the right side bearing, took about two minutes instead of twenty. More important, the glyphs are the same shape and proportion. Before fixing the whole font it is best to preview the results. Copy the new glyphs **temporarily** over the originals and see how they look in the Preview Toolbar (F8). Since the bold is too heavy, it would be better to use the Book to DemiBold script, which doesn't increase the weight quite so much.



abcdefghijklmnopqrstuvwxyz

Making a bold style from a regular font is more difficult than it looks. That is why Victor Gaultney hasn't released » a bold version of Gentium « after several years.

11. Superscripts



If superscripts are just scaled from the existing figures in a font, they will be too light compared to the rest of the font.

First, measure the weight of digit 1. In this font it is 160 funits. If the superscript transformation scales this by 65% horizontally it will be only 104 funits. To compensate, one needs to use the bold transform with a horizontal value of 28 funits. This will thicken the stroke by 28 funits in both directions, resulting in a vertical stroke of 160 funits — 104 + 28 + 28.

The horizontal strokes iare 140 funits. If the vertical scale factor is 60% the vertical transformation would again need to be 28 (140 - 84)/2

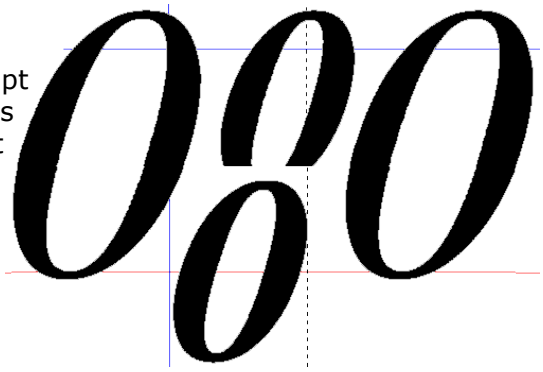
The illustrations show the digit 2, the digit 2 as a superscript (centre), and a true superscript ² (right). With scaling alone, the superscript is too light. The weight of the superscript is correct after the bold transformation has been applied. However, the thin strokes and serifs are now too heavy. They need to be adjusted by moving a few nodes.

Each font will need a different transformation. This script just uses average values. And some designers may prefer to raise superscripts higher. This script aligns the tops of superscripts with the tops of numerals (1¹). If a font already contains superscripts ¹²³ the script will only add the missing ones.

Script: Superscript.xml
Insert Characters 8304, 185, 178, 179, 8308-8316 (Apply subsequent features)
Complete Composites, Decompose
Scale (65.00, 60.00) LT
Bold (26, 20) preserve bearings
Move (0, -20)
Width (fixed 899) both sides (Remove this line if the figures are not of uniform width)
Left Side Bearing Point at x=0
Insert Characters 8305,8317,8318,8319
Complete Composites, Decompose
Scale (65.00, 60.00) LT
Bold (26, 20) preserve bearings
Move (0, -20)
Left Side Bearing Point at x=0

Superscript width (899) = figure width (1303) x scale (65/100) + Bold (2 x 26)

12. Subscripts



After adding the superscripts, use the Subscript Transformation to insert the Subscripts. This merely creates composites of the superscripts, and moves them down to bisect the baseline. This is the most useful position for subscripts as recommended by » Microsoft Typography « To raise or lower them adjust the vertical position using guidelines or the nudge keys (up, down with or without shift/control).

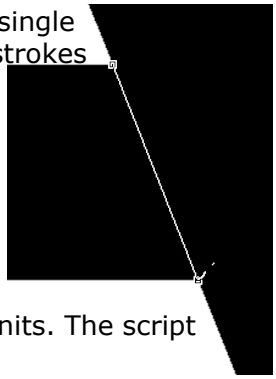
Script: Subscript.xml
Insert Characters 8320-8334 (Apply subsequent features)
Complete Composites

Superscripts and Subscripts in Italic type styles need to be offset horizontally to compensate for the italic angle. The recommended position is centred between figure zeros. A convenient way to do this is by using the Glyph Comparison Toolbar (F11) and the left/right keys (with Control and Shift modifiers).

The default transform scripts do not insert all possible superscripts and their subscripts, only the numbers and maths symbols.

Most applications won't use super/subscripts even if they exist in a font. They will just scale the numerals, which doesn't give the best results. Again, not all applications use the super/subscript data on the Format, Settings, General Tab as this data is often missing or wrong. The situation may improve if more font designers take the trouble to design their fonts properly.

13. Thin With High Contrast



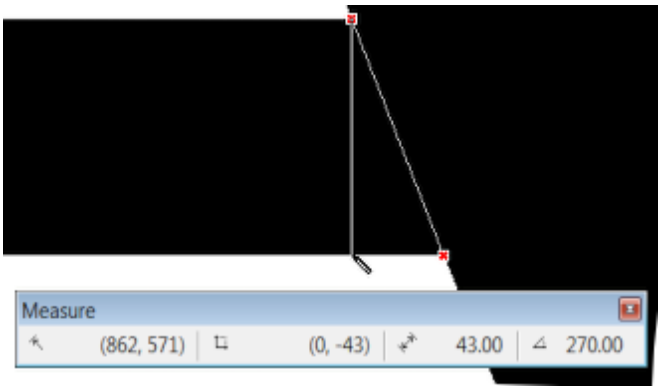
If the glyphs have even strokes one can get a satisfactory result with a single transformation. However, if the glyphs have a high contrast between thick and thin strokes one must apply the transformation several times to the thick strokes, but only once to the thin strokes. Bold and heavy transforms on fonts with high contrast can use a similar method.

Cutting the Glyph

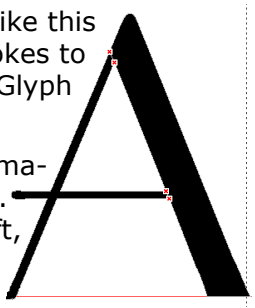
Zoom in close using Point Mode and cut the glyph with the knife tool to separate the thick and thin strokes. At the same time, measure the thin and thick strokes.

For this glyph, the thin strokes were 43 funits and the thick strokes were 184 funits. The script was designed to reduce the stroke weights to 33 funits and 144 funits respectively.

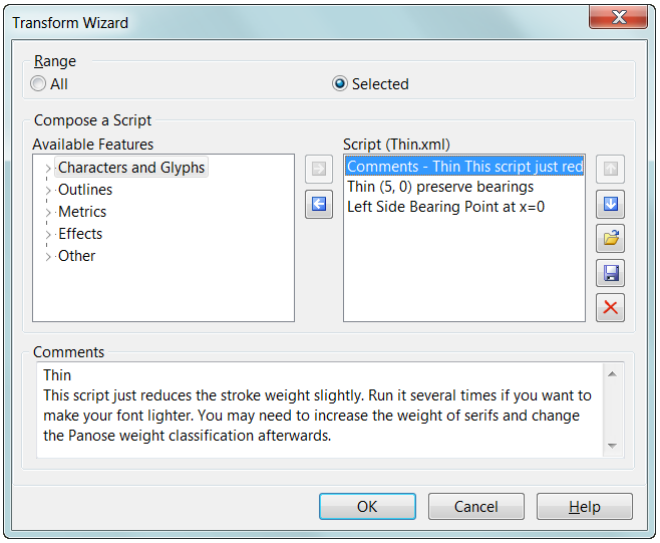
Applying the Transformation



After cutting, the glyph looks like this in Contour Mode. Cut the thin strokes to the clipboard and open the Glyph Transform Wizard dialogue.



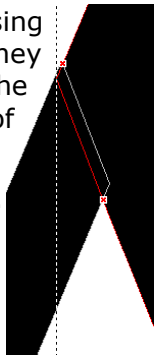
Clear any existing transformations by clicking on the delete icon. From the Effects group on the left, choose the Thin Transformation and click the right arrow to add it to the script. Enter the value for the horizontal transformation. Check the box to preserve side bearings. From the Metrics group on the left choose "Left Side Bearing Point at x=0" and add it to the script.




```
Thin (5, 0) preserve bearings
Left Side Bearing Point at x=0
```

Click OK to apply the transformation once to the remaining thick stroke. Repeat twice more, then paste the thin strokes back into the glyph, and apply the transformation again. Total, four times for the thick stroke, and once for the thin strokes. To make the horizontal strokes thinner too, do this at the final stage with:

```
Thin (5, 10) preserve bearings
Move (0, -10)
Left Side Bearing Point at x=0
```



Zoom in and align the strokes using Control + Left/Right Cursor keys so that they join neatly at the top. Select all and join the contours together with "Get Union of Contours" from the Glyph Toolbar. Delete any unwanted nodes to complete the glyph.

To reduce the work-load save the transform script after testing that it works as desired. Then:

- Copy and paste the glyphs to be transformed into empty glyphs
- Cut each of them with the knife tool into thick and thin strokes.
- In the Overview Window select the cut glyphs, copy them to the clipboard, and from the Insert menu, insert one glyph before each selected glyph.
- Paste the clipboard contents into these new glyphs to end up with two copies of each cut glyph.
- Delete the thin strokes from one copy and the thick strokes from the other copy
- Run the script on the thick strokes with suitable large values
- Run the script on the thin strokes with suitable small values
- Copy the thin strokes into the glyphs with thick strokes and join them
- Copy the modified and recombined glyphs back into their original code-points with paste special to copy the glyph outlines and glyph metrics, but not the glyph mappings.
- Delete the unwanted working copies.

14. Discretionary Ligatures

In the days of hot-metal type, many letter pairs were created as ligatures. Only a few of these have survived in most modern fonts — ff, fi, fl, ffi, ffl, and occasionally one finds ligatures with long s (ſt), st and ct. The latter doesn't have a code-point, but the former pairs are encoded in the Alphabetic Presentation Forms character set. A few fonts include historical ligatures found in the Latin Extended D character set, and others contain discretionary ligatures for purely decorative purposes.

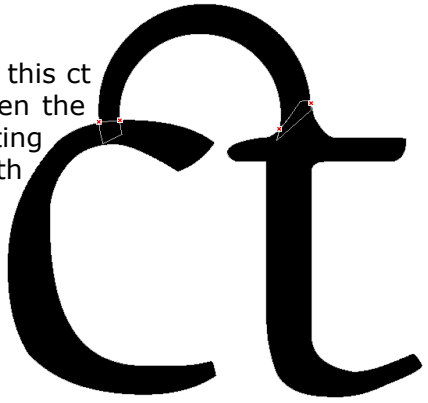
ÆæÆœAJauAvAyŒœVYwy ſäſhſiſlſöſþſſſi
ſſlſtiſtrſüſbſk ckctfjfrftfyfftffyftyſptrttttytytztttr
ckykyffriſpiſttw QuTh FFHFLHELAMBMDMEMPMR
NKNTŒŒGTTTWTYUBUDULUPUR ffiflffiffliſtst

This script inserts the above characters. If a font doesn't yet include the standard ff, fi, fl ligatures, this script will also insert them. After running the script, connecting contours need to be added to make the pairs into ligatures. There is scope to be creative with discretionary ligatures, but standard ligatures in the Alphabetic Presentation Forms (ff - ffl) should closely match the standard letter forms because standard ligatures are usually enabled by default.

Script: Discretionary Ligatures.xml
Insert Characters 42802-42809,42812,42813,42830,42831,42848,42849,61124,61125,61129-61131,61133-61136,61143-61148,61150,61151,61152,61363-61374,61378-61390,61392,64256-64262 (Apply subsequent features)
Complete Composites

Joining Ligatures

Some ligature pairs need to be joined with new contours. To create this ct ligature a ring from the samples toolbar (F12), was resized to suit, then the centre of the ring was offset to make it thinner at bottom left. After cutting the ring with the knife tool, the three contours are ready to be joined with "Get Union of Contours."



The same connecting contour can be used for the st ligature. Reusing contours helps to maintain a consistent design. The copy can be modified to fit the new pair, but the weight of the stroke especially should be the same for all related glyphs.

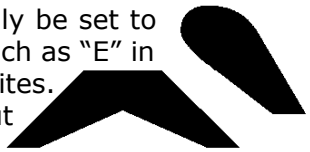
15. Low Profile Diacritics

Some fonts like Bitstream Vera (illustrated) use smaller accents on uppercase letters than they do on lowercase. FontCreator's Complete Composite feature will use them for uppercase if they exist. This script will insert lowercase diacritics in the Private Use Area and compose them from regular accents. That is just the first step in the design process. Their height needs to be reduced to suit the design of the font, while maintaining their weight so that they match other accents.

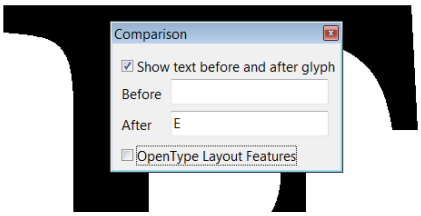
Although they are only used for uppercase composites, low profile diacritics should align vertically with regular accents because Complete Composites will move them up for uppercase composites, just the same as it does with regular accents. Acute, grave, breve, circumflex, caron, tilde, ring and hook have low profile versions, while diaeresis and dot above do not need them. Low profile accents should align with the middle or bottom of diaeresis, rather than with the middle or bottom of the regular caron accent. However they are designed, FontCreator will simply move them up for uppercase by the difference between x-height and CapsHeight.

16. Stacking Diacritics

If creating a font that offers full support for the Latin Extended Additional character set, which is essential for Vietnamese users, then one needs to add stacking diacritics. This script will insert them into the Private Use Area, and make a "best guess" at composing them from regular diacritics or low profile diacritics (if they already exist). Design them to fit below WinAscent and above the uppercase vowels.



After running this script, the advance width for stacking diacritics will initially be set to zero. This helps in the design process as one can display an uppercase vowel such as "E" in the Comparison Toolbar to see how they will look when used in composites. Complete Composites will centre them horizontally above each vowel, without moving them vertically. Some accents may need horizontal adjustment. The circumflex with grave accent (illustrated) is sometimes designed with the grave on the left (Gentium), or above (Times New Roman), but the composite data follows the recommendation of a Vietnamese user.



If the vertical space between CapHeight and WinAscent is really not enough, WinAscent must be raised before designing these accents. If any glyphs cross the WinAscent (or WinDescent) line, they will be clipped in most applications. Raising the WinAscent (or lowering the WinDescent) will increase the default line-spacing for the font when applications use single line-spacing. This may make text too widely spaced when not used for Vietnamese. The Gentium font breaks this rule, so it is unsuitable for use in all applications when typesetting Vietnamese. If fonts are designed so that the distance between WinAscent and WinDescent is exactly 2458 funits (2048 x 1.2), 10 point body text will be set at 12 point line-spacing when single spacing is used.

17. Spaces

This script inserts all of the fixed width spaces in the General Punctuation character set. Open Office and some other applications allow the user to use these special spaces for typesetting. After running the script there is nothing to be done, unless one wants to change the recommended advance width for thin space (1/5 em) or hair space (1/10 em). Punctuation space and figure space use the advance width for the period (46) and figure zero (48) respectively.

18. Letter-like Symbols

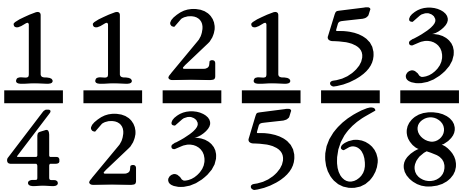
This transform script will add several letter-like symbols. It is just the initial step in the design process. Most of the glyphs will need editing to adjust weight and spacing. Some glyphs need to be rotated. Double-struck capitals should look like these glyphs from Lucida Sans Unicode, not like the simple outline letters produced by the hollow transform. Cut the outer contour to the clipboard to work on the inner contour, before pasting the outer contour back.

Script: Letterlike Symbols.xml
Insert Characters 8448, 8449, 8451-8454, 8457, 8462, 8468, 8470, 8471, 8478, 8480, 8481, 8482, 8486, 8487, 8490, 8491, 8498, 8505, 8506, 8507
CompleteComposites
Decompose
Insert Characters 8450, 8461, 8469, 8473, 8474, 8477, 8484 (Apply subsequent features)
CompleteComposites
Decompose
Hollow (50, 50) preserve bearings
Left Side Bearing Point at x=0

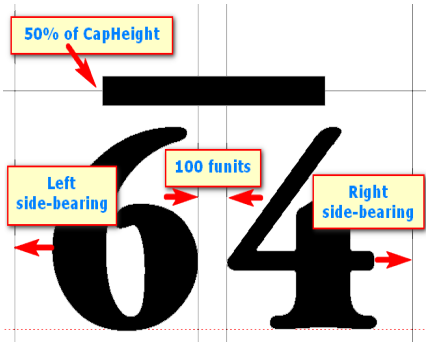
19. Stacking Fractions

Stacking fractions or nut fractions are designed to save space when typesetting fractional measurements. Regular fractions like ¼ and ½ use the fraction slash separator, while stacking fractions use a horizontal divisor.

Script: Stacking Fractions.xml
Insert Characters 60600-60609 (Apply subsequent features)
Complete Composites
Decompose
Bold (7, 5)



Move (0, 5)
Center Glyph
Set Left Side Bearing Point at x=0
Insert Characters 60611,60613,60615-60617,60619,60621,60623,60625,60627,60629,60631, 60632,60633,60635,60637,60639,60641,60643,60645,60647,60649,60651,60653,60655,60657, 60659,60661,60663,60664 (Apply subsequent features)
Complete Composites
Insert Characters 58188-58190,60531-60542,60544-60599 (Apply subsequent features)
Complete Composites



This script inserts characters in the Private Use Area, scaling the nominators and denominators from figures. It applies a bold transform and centres the glyphs. The advance width for the divisor, and thus for all single figure numerators and denominators, adds side-bearings of 100 funits to either side of the underline glyph after scaling it down by 50% to create the divisor. It is aligned vertically at 50% of the CapHeight.

Nominators and denominators from 11 to 63 all derive their width from the 1/64 denominator, the side-bearings are those of the six and four denominators. 100 funits separate the two glyphs.

20. Ordinals

The Ordinals Feature in OpenType fonts uses superscripted letters automatically after numerals e.g. 1st, 2nd, 3rd, 4th, 1èm 2º 3ú etc. This script inserts superscripts for both lowercase and uppercase in the Private Use Area.

Insert Characters 60065-60090, 60096-60122, 60180, 60200, 60218, 60232, 60250
(Apply subsequent features)
Complete Composites
Decompose
Scale (70, 68) Fixed point (0,0)
Bold (25, 8)
Move (0, 508)
Left side-bearing point at x=0

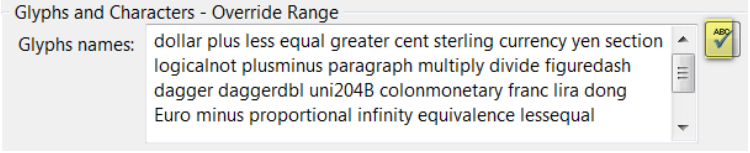
This script inserts glyphs for A-Z, a-z , grave, acute, È, è,Ú, and ú in the Private Use Area. It scales them by 70% about the origin and makes the glyphs bolder to compensate. It then moves the glyphs up so that the tops of ascenders in the superscripts align with the tops of ascenders. This value will need adjusting to suit each font.

21. Override Range

The override range features can be used to select specific glyphs prior to performing other glyph transformations on them. Glyphs can be selected using "Override Range by Codepoint(s)," or "Override Range by Glyph Name(s)" (which is necessary if glyphs are unmapped). If a font has Tabular Figures in the Private Use Area that are unmapped, this new feature could be used to select them by glyph name, e.g. zero.tnum, one.tnum … nine.tnum, or whatever glyph names are used.

Override Range by Glyph Names (dollar plus less equal greater cent sterling currency yen)
Width (Fixed 1200) Both Sides
Center Glyph
Left Side Bearing Point at x=0

This script selects some maths and currency symbols, applies a fixed width, centres the glyphs, and sets the left side-bearing at zero. Add or remove glyph names to the range and adjust the width to suit the figure width of your font before running it.



Click the Validate button to count the number of glyphs and check for errors

22. Inverse

This simple script has no parameters. It adds a black rectangle to each selected glyph, and reverses the direction of the contours to produce a white glyph on a black background.

Override Range by Codepoint (65-90)
Inverse

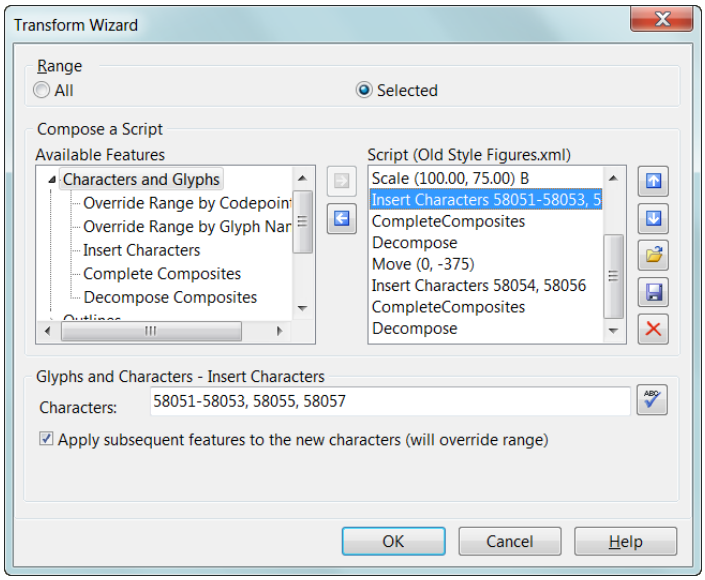
This script would select the letters A-Z, add a black rectangle the width of the glyph and the full height between WinDescent and WinAscent, reversing any contours to create a white glyph on a black background.

23. Optimize

This script has no parameters. It reduces the number of points of all contours in the selected glyphs. Running this script may have a subtle effect on curves so check that the results are satisfactory before saving the changes. Use the Background Image toolbar with Fill Outlines disabled to easily compare the results before and after this transformation.

24. Editing and Saving Scripts

Transform scripts are plain text, XML files. You can edit them in the Transform Wizard dialogue, or use a plain text editor once you know the correct syntax. Be careful, as scripts won't run at all if you get the syntax wrong. Click on the Open folder icon to load existing scripts, and click on the Save icon to save customised or new scripts. Click on the delete icon to clear the current script (it will not delete anything on the hard drive). To replace a default script with and edited version, use the save icon and overwrite the existing script.



When inserting characters, the Validate code-points icon will check that the code-points are valid, and count the number of characters that will be inserted. There is a limit of 1,000 characters to the insert characters command, but this feature can be used several times in one transform script to overcome that limitation if working with very large Unicode or CJK fonts.

Additional features can be found on the left of the Glyph Transform Wizard dialogue. Most of them have several options.

25. Appendix

Summary of Code-points in the Private Use Area Used by Scripts

57334-57364	Stacking Diacritics for Vietnamese
57365-57378	Low Profile Diacritics for Uppercase or Small/Petite Capitals
58033-58126	Basic Latin for Petite Capitals and Titling Capitals
58048-58057	Old Style Figures, Proportional Figures, or Tabular Figures
58160-58255	Latin-1 Supplement for Petite Capitals and Titling Capitals
58188-58190	Stacking fractions for ¼, ½, and ¾
58256-58383	Latin Extended-A for Petite Capitals and Titling Capitals
58384-58591	Latin Extended-B for Petite Capitals and Titling Capitals
58912-58974	Basic Greek for Petite Capitals
59072-59218	Cyrillic for Petite Capitals
59680-59829	Latin Extended Additional for Petite Capitals and Titling Capitals
60531-60599	Number Forms, Nut Fractions or Stacking Fractions
60065-60255	Basic Latin and Latin-1 Supplement Superscripts or Ordinals
60320-60334	Historical Ligatures ſä to ſk (Medieval Unicode Font Initiative)
60600-60664	Divisor, Numerators, and Denominators for Stacking Fractions
61122-61398	Discretionary Ligatures: bb to ffk

About the Private Use Area

There are 6,400 code-points in the Private Use Area BMP (Basic Multilingual Plane) from 57344 to 63743. As the name suggests, font designers can use this area however they wish. They are intentionally left undefined so that third parties may define their own characters without conflicting with Unicode Consortium assignments.

FontCreator uses Unicode mappings for its Complete Composites feature. For example, if an empty glyph is mapped to decimal code-point 224 à the feature can compose à grave from lowercase a (code-point 97) and grave accent (code-point 96). The font designer only needs to ensure that aeiou and grave accent exist in the font to compose àèìòù.

This feature was extended to the Private Use Area to automate the creation of glyphs for OpenType features, and the composition of accented characters with those additional glyphs. If the font designer has already created a Petite Capital A at code-point 58097, then Complete Composites will use that glyph with grave accent to compose Petite Capital À at code-point 58224. Decimal code-points in the Private Use Area were chosen to correspond with those used by the same glyphs in the Latin character sets.

If the font designer has taken the extra step of creating a low-profile grave accent at codepoint 57365, then the Complete Composites feature will use that smaller accent instead of the standard grave accent. The standard grave accent will be used as a fallback glyph if no low profile version exists. If no grave accent exists either, the composite glyph will be incomplete, *i.e.* it will show only the base glyph without any accent, and if no Petite Capital A has been created yet either, it will be empty, although the blue caption colour in the Glyph Overview will indicate that it is a composite glyph.

Designers who are developing fonts for Hebrew, Arabic, or Asian scripts may want to use the Private Use Area in an entirely different way. The CompositeData.xml is a plain text file and can be edited in any suitable text editor to customise the Complete Composites feature in any way that the user wishes. Please see the Complete Composites Tutorial for some suggestions.