

The Cabbagetown series of fonts are © 2014 by bobistheowl for Metaphase Brothel Graphix. They are free to download, and free for personal use.

Unauthorized commercial use of the Cabbagetown font series is prohibited. For commercial use and licensing, please contact us by e-mail at:

cabbagetown.license@teksavvy.com

The fee for commercial use and licensing is variable, depending on how the font(s) is/ are to be used. The fonts themselves and the designs will enter the public domain on January 1, 2027, at which date they will become free to use for any purpose.

Thanks to Claude Pelletier for the use of his capital E glyph from the font *Les Etoiles* as the basis for the ampersand in Cabbagetown.

Thanks to Toto for technical assistance, and his tutorial on changing vertical metrics with Studio5.

Thanks to Lisa for design improvement suggestions and encouragement.

Thanks to Sandy for technical assistance, and for the creation of the ‘Open Type’ supplemental glyphs, (M Dash, N Dash, ellipsis, etc.).

Thanks to Luc Devroye for his ongoing support, and for hosting my home page.

Thanks to Pablo Impallari for locating the Light Shade patent document.

The Cabbagetown series was made with MS Paint, MS Word, ScanFont 3 and FontLab Studio5. WinSnap 3.5.5 was used for screen captures of vector images.

MS Paint was used to create the monochrome bitmap source graphics, and to edit screen captures. MS Word was used in the early editing stage, where large point size versions of glyphs being edited were copy/pasted to MS Paint to create bitmap versions of the edited vectors. The fonts were created and edited with ScanFont 3. FontLab Studio5 was used to change the UPM size and vertical metrics for the Std, Micro, and Nano versions.

Cabbagetown was rendered in four sizes, and four styles:

Original size: These fonts have an x-height of 1710 units, approximately 240% larger than standard size. I would recommend that the original sized fonts be used for large scale text display, above 72 standard points, (equivalent to 30 point size for these fonts).

Standard size, (Std): The Std fonts have an x-height of 712 units. These fonts are best used for text at or below 72 points, or for text which uses more than one font.

Micro size: The Micro size has an x-height of 171 units, (10% as large as the Original size).

Nano size: The Nano size has an x-height of 119 units, (1/6 as large as the Std size).

The Micro and Nano sized fonts are included in the download on my home page:

<http://luc.devroye.org/bobistheowl.html>

but are generally not included in the download .zip available on other sites. I made them to test a theory regarding degradation of text when vectors are reduced in size. Using the Nano size as an example, since the Nano version is 1/6 as large as the Std version, text at 72 points in Nano will be the same size as text in Std at 12 points. In theory, the 72 point size of Nano should look better than the 12 point size in Std, because I expect that any degradation caused by reducing the size of a vector through a change in the vertical metrics/ UPM size will be less than the degradation caused by reducing the vector size by selection text display at less than 72 points.

The Std size looks great in text at 14 points* and larger, so the Micro and Nano versions are only useful for text display below 14 standard points.

***Please note:** The Cabbagetown fonts look much better in print than they do in electronic display on a computer monitor, particularly the smaller point sizes.

Regular Style: The regular style includes both the glyph and the visible side edge. The observer's point of view is below the glyph in two dimensions, and above the glyph and right of center in three dimensions, or the same point of view that one would have, if looking up at a store sign from the sidewalk. Each glyph is surrounded by a white

border. The ‘shadow’ represents the visible portion of the side edge of a three dimensional glyph, in perspective. Although the width of the white border and the side edges are roughly equivalent in the glyphs, (the side edges are approximately 4-8% wider than the white border, by design), in the three dimensional model, the side edges are significantly taller than the white border widths, but they are foreshortened by the observer’s position. At least, that’s the effect I was trying to convey.

Stone Style: The Stone style shows the side edges only. It can be used as a stand-alone font, or in conjunction with the SmCaps style to create composite images, where perhaps a different colour, texture, or effect is used for each of the main glyph and the side edge.

The Regular, Stone, and SmCaps versions have identical spacing between glyphs. Typing the Stone style on top of the SmCaps style will reproduce the Regular style in text, Or embossing the Stone style on top of SmCaps can create actual three dimensional glyphs.

SmCaps Style: This version is really only useful in combination with the Stone style.

Book Style: This is a stand-alone font of the main glyphs, without the side edges. The spacing on the right side of each glyph is different from that used in the other three styles. In addition, three glyphs are slightly different from the SmCaps version, W, % and \$.

This graphic shows the differences in the dollar sign glyph, for the four styles:



Left to right: Regular, Stone, SmCaps, Book.

For the W, the upper left interior serif is wider in the Book version than in the Regular or SmCaps styles, and for percent, the two O’s are closer to the forward slash.

History of the design:

The original design for Cabbagetown dates from 1874, and an alphabet patent named Light Shade, by Richard Smith:

<http://cl.ly/image/3E3C1Q1J052s/1874%20Light%20Shade%20-%20Richard%20Smith%20Patent.png>

It later appeared in Dan X. Solo's *The Solotype Catalog of 4,147 Display Typefaces* on page 17 as *Night Shade*.

The first known digital version of this typeface was *Nigel SadeSH*, by Soft Horizons, (1993). Other versions include *Shadowed Serif* by James Fordyce, (1994), *Cameo Antique* by Character, (2009), and my version, *Outstanding*, from 2012.

Shadowed Serif appears to have been made from poor quality source graphics. *Cameo Antique* uses the same vectors as *Nigel SadeSH*, and adds an outline version of the alphabet glyphs, as well as correcting several encoding errors present in *Nigel SadeSH*. *Outstanding* was created from the large sized .jpeg images available under the title *Vintage Letter* at bigletters.org:

<http://bigletters.org/big-decorative-vintage-letters.html>

After completing *Outstanding*, I was dissatisfied with the results. I liked the concept of this alphabet, but not the rendering of the design. I thought the letters were too wide, too rough, and inconsistent with each other to be of any practical use, so I decided to redesign this alphabet, and add additional glyphs, (*Outstanding* had no numbers, symbols, nor punctuation).

The first design change I made was to reduce the width of each glyph by 20%, without changing the height. This was done using *Stretch/ Skew* in *MS Paint*, using the monochrome bitmap source graphics I had prepared for *Outstanding*, (essentially the .jpeg images from bigletters.org, cropped and resized, then manually traced to two colours, composed for monochrome, then saved as monochrome bitmaps, for importing into *ScanFont 3*).

After importing the reduced width monochrome bitmaps, my first editing stage, as with *Outstanding*, was to remove all nodes from the contour outlines that were not necessary to hold the contours in place. Each vector in the *Cabbagetown* font series uses the minimum number of nodes, (nodes are points located on intersections of editing units in a font editor).

A contour, (black section in text), can be created from two or more nodes, connected to each other. In ScanFont 3, there are three types of nodes, (red green, and blue), which behave differently:

A line between two red nodes, or between a red and a blue node, will be straight, with no control handles attached to either node.

A line between a red and green or a blue and a green node will have one control handle attached to each node.

A line between two green nodes will have two control handles attached to each node. In addition, the blue node, if converted to green, allows for an additional node to be created, if the node's position is moved. Control handles are also located on e horizontal and vertical editing unit intersections. Their positions, relative to the positions of the nodes to which they are attached, determine the degree and location of curvature in a line that isn't straight).

Wherever a node appears on a curved line, some 'flat spot' will necessarily appear. By using as few nodes as possible, the number of flat spots is reduced, resulting in smoother curves. The flat spots can also be de-emphasized by placing the nodes on curved lines in places where they are least noticed.

Conventional wisdom in typography states the nodes should be placed on the extreme points of curvature, (ie: at North, South, East and West, when making a circle or an ellipse). This guarantees a noticeable flat spot at each of those locations, when the text is viewed at very large point sizes.

I found that a smoother circle or ellipse can be created using three nodes instead of four, with the curve on top created by the control handles being placed on very specific horizontal planes. The proper planes on which the handles should be placed required a lot of trial and error.

For the original sized fonts, the x-height of 1710 units was determined by the size of my monochrome bitmap source graphics, which have an equivalent height of 171 pixels, not including the side edge below the glyph, which is placed below the base line. Rounded top glyphs are 1736 units high, (about 1.5% taller).

If the round topped glyphs were 1735 or 1737 units high, a flat spot would appear in word processing text at large point sizes. I have no doubt that round top display could be created at heights other than

1736 units, but that would have created a discrepancy in the size of the glyphs. If all glyphs in the Original size were 1710 units high, the round topped glyphs would appear, to the eye, to be shorter than those with flat tops.

I began working on Cabbagetown in February, 2013. In March of that year, I moved to the Cabbagetown neighbourhood in Toronto, Canada, which, according to Wikipedia, has the largest number of Victorian Era homes in North America. Since the original design of Light Shade is Victorian, and since most of the font was made after my move, Cabbagetown was a logical choice for the font name.

After removing the unnecessary nodes from each vector, I spent a few weeks designing and drawing source graphics for numbers, symbols and punctuation, as well as experimenting with a sans serif version of the font, which I ultimately chose to not complete.

At the beginning of June, 2013, I started to use large scale bitmap image versions of semi-edited glyphs as an editing tool. My procedure was as follows:

- 1) I created and installed a beta version of the font, after the most recent positive changes.**

- 2) I opened MS Word 2007, selected the most current beta font version, and displayed text at 288 points, (about 690 standard points, because I was using the Original fonts , with 1710 unit x-height). To do this, I had to first change the paper size to maximum, (22 by 22 inches), but I had no intention of printing the glyph on paper. I generally edited only one glyph at a time, to completion, unless I was making a similar change to several different glyphs at the same time).**

- 3) I selected and copied the glyph from MS Word, and pasted it into a blank document of MS Paint. This procedure reduces the size of the image to 75%, so the image in Paint is at 216 points, but at this size, it is monochromatic, (smaller sized text will usually produce a multicoloured image that contains various shades of grey and beige, when the image is enlarged). I generally edited the bitmap images at eight times enlargement, (where individual pixels are easily discerned), with grid lines enabled.**

This large scale pixel representation of the vector allowed me to see the actual line slopes of diagonal straight lines, and also allowed me to make quantitative measurements of distances, and to make comparisons between one glyph and another. It also allowed me to create diagonal lines that are parallel to each other. I could also measure the widths of the white borders, side edges, cross stems, serif lengths, etc., to ensure that they were consistent from one glyph to the next.

For diagonal lines, I wanted the line slopes to be equivalent to simple fractional numbers like $3/1$, $7/4$. Or $8/3$, rather than approximating them 'by eye'. Through trial and error, I found that there was almost a 4:1* correlation between the size of my vectors at maximum enlargement, and the size of the bitmap, at eight times enlargement, meaning that if I moved a node four units in one direction in the vector, it would result in a pixel moving one unit in the same direction in a subsequent bitmap. Because the bitmap was of larger scale, however, moving a node 1-5 units might move the pixel in a subsequent bitmap. If I wanted a diagonal line to have a slope of $8/3$, that would appear in the bitmap as a series of square steps with a recurring period of 3,3,2. By drawing the 3,3,2 line in a different colour onto the bitmap. And comparing the actual line in black to the desired line slope, I got a rough idea of how many editing units I should move a node in the vector, to achieve the desired line slope.

*** The correlation was almost exactly 3.94 editing units in the vector per pixel in the bitmap. I divided the x-height of 1710 units by the equivalent height in the 216 point bitmap, (434 pixels).**

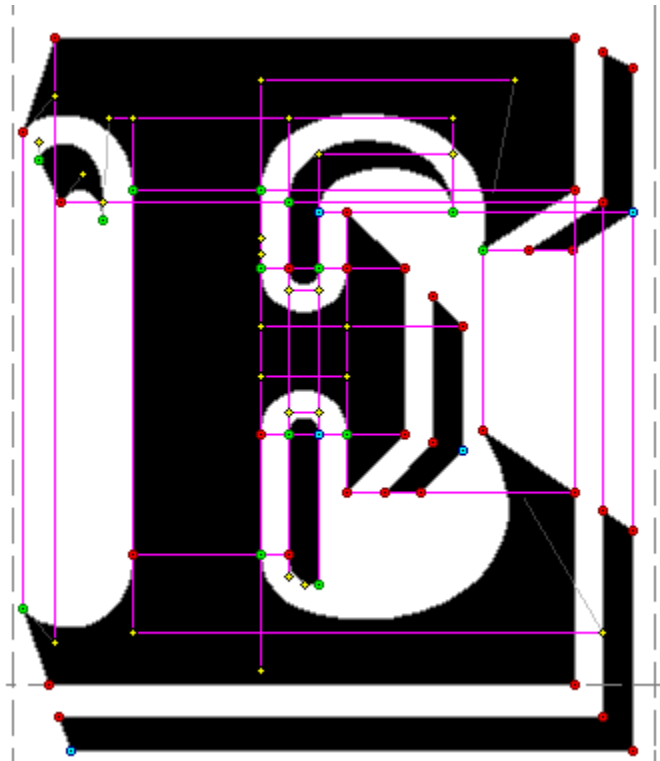
- 4) After making one or more positive changes, I generated a new beta font, and repeated steps 1-3, until I could make no further improvements.**

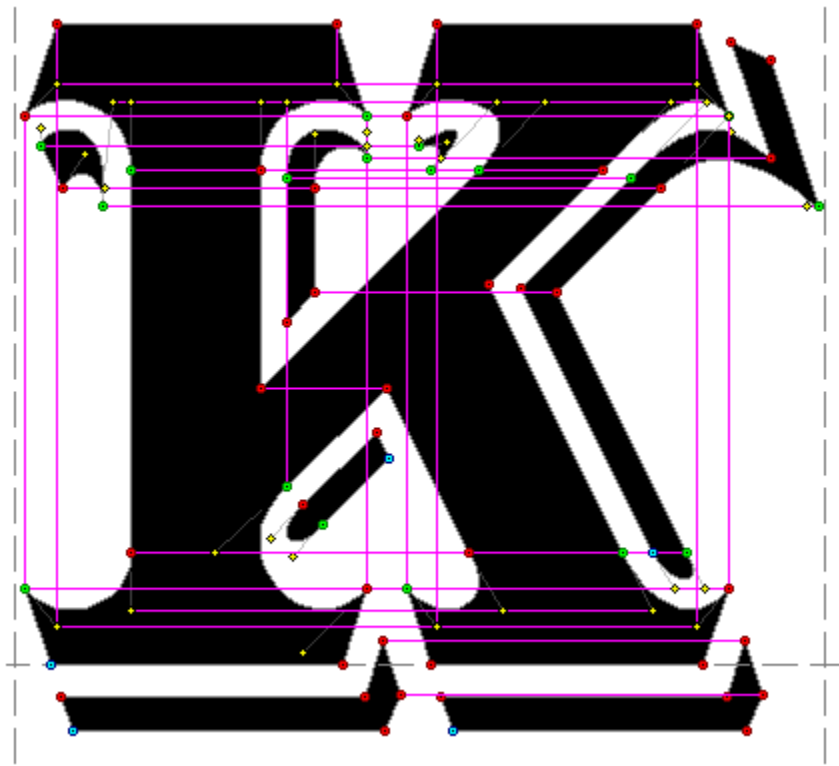
After editing all of the alphabet glyphs with this procedure, I found that, by holding the left mouse button down in ScanFont 3 while pointing at a node, handle, or the intersection of a horizontal and vertical editing unit plane, (with perpendicular guidelines enabled), the X/Y co-ordinates of that location were displayed in the glyph window, at lower right. These were measured from the left side bearing for vertical, and from the baseline for horizontal. This information allowed me to refine the line slopes more precisely that I could with the large scale bitmaps, as I could now measure, exactly, the horizontal and vertical distances between the nodes at the beginning and end of the line. If I wanted a line to have a slope of $7/4$, the height of the top node minus the height of the bottom node would need to be exactly divisible by 7, and the

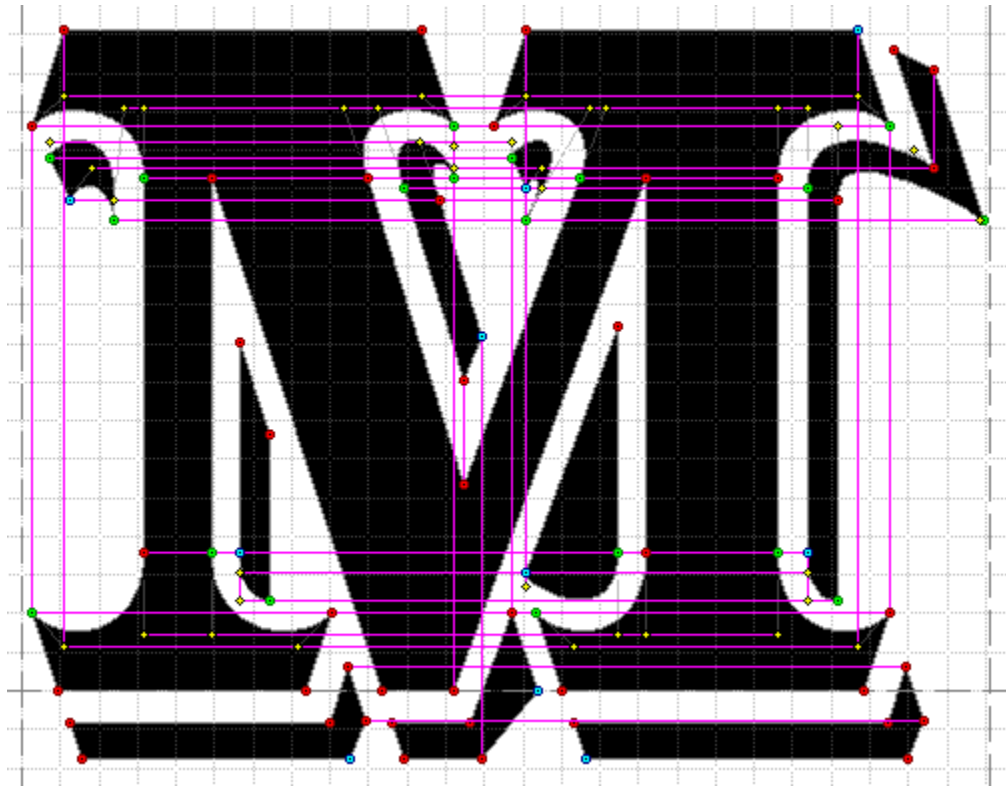
distance left or right would be exactly $\frac{4}{7}$ of the change in height, divided by 7.

After refining my vectors further through precise calculation of the line slopes, I noticed that many of the nodes and control handles appeared to be exactly or almost exactly aligned on specific horizontal or vertical planes. This was completely unexpected, and I began to attempt to align points, while maintaining the same line slopes. These horizontal and vertical alignments appeared in nearly every glyph, except glyphs like the comma/ single quote, which were composed entirely of curved lines, with few nodes and handles. The horizontal and vertical alignments of nodes and handles appeared just as frequently in glyphs that had mostly curved lines, in addition to glyphs like M or A where almost all of the lines are straight.

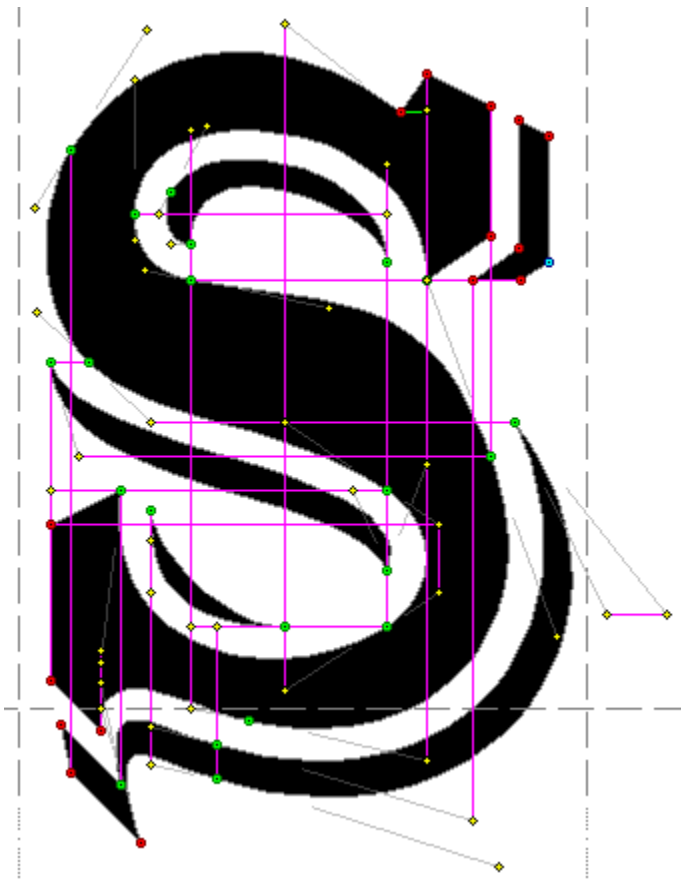
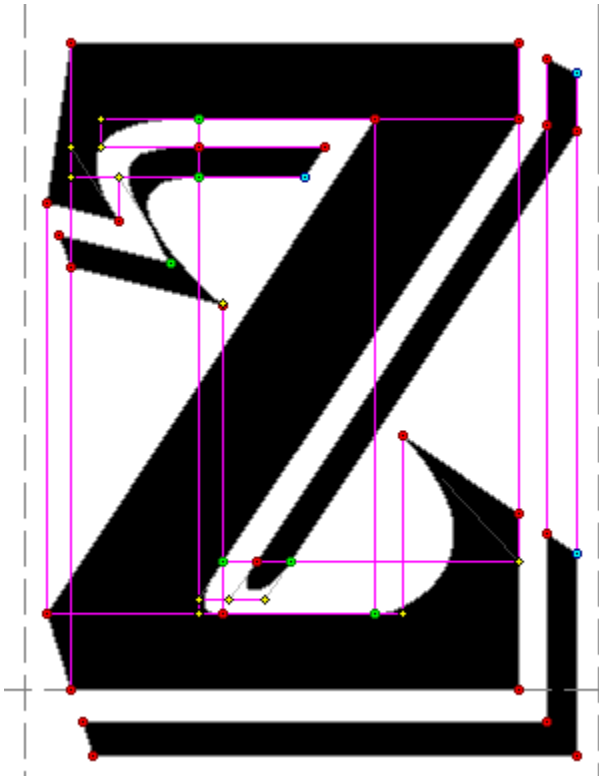
Here are a few examples, from screen captures of the vectors from the Original fonts, at 50% size. Nodes are blue, red, or green. Handles are small yellow + signs. The purple lines drawn on top represent the horizontal or vertical alignment planes. All of these alignments are exact, not approximations based on the scale of the screen captures. At maximum enlargement in ScanFont 3, A glyph can appear approximately 14 feet tall, (about 4.2 meters tall), with only a small portion visible on my computer monitor:







The grid lines in the background represent 100 editing units horizontally and vertically, so within each grid square, there are 10,000 possible X/Y intersection points.





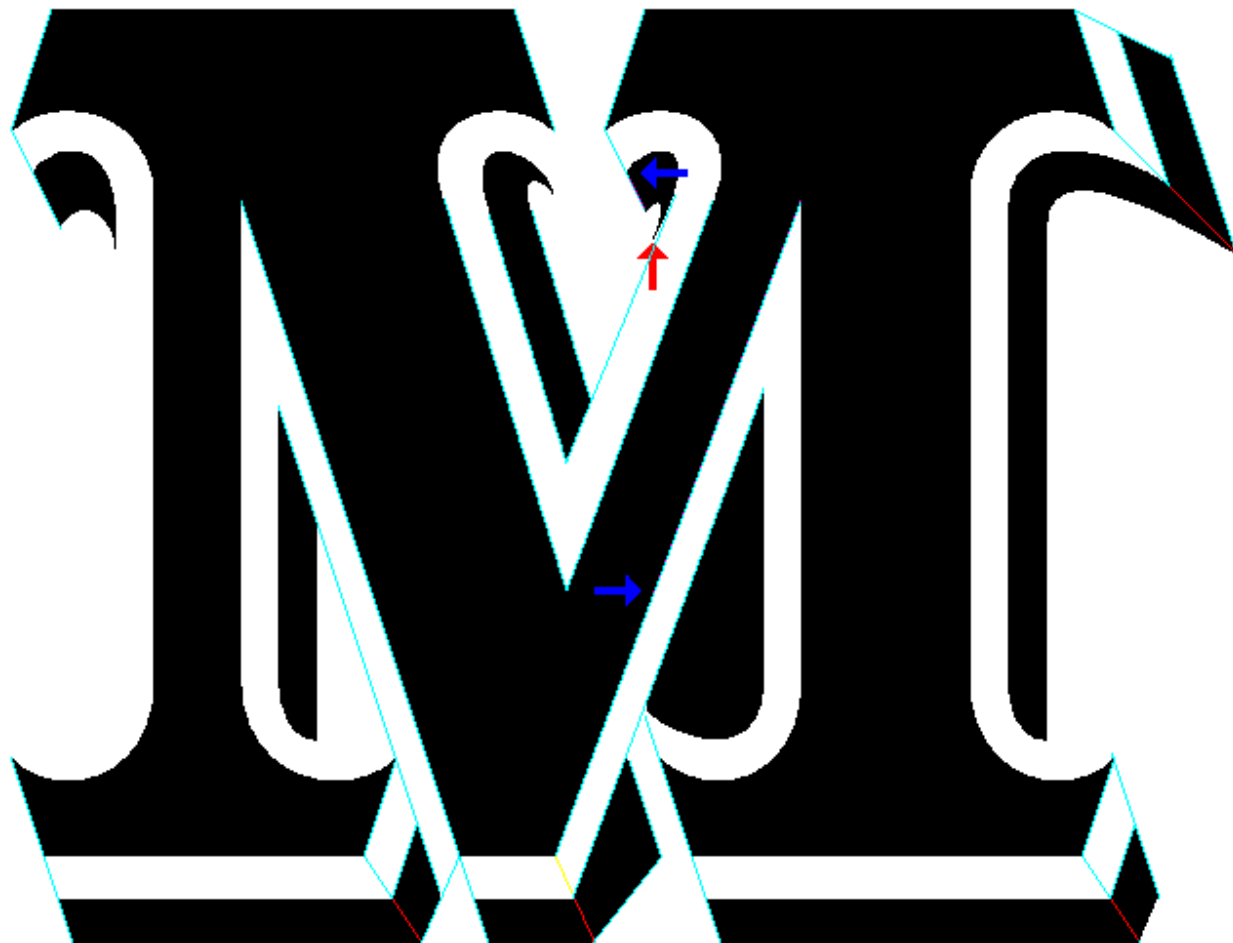
In the lower portion of the 2, there are six nodes aligned diagonally, with a $3/2$ slope.

For the most part, every alignment of point had either a neutral or beneficial effect on the appearance of each glyph. The alignments were also helpful, when I changed the UPM size and vertical metrics, as most of them were automatically incorporated into the Standard sized version.

Here's a 'mashup' of the letter M glyph, before I started the bitmap editing stage, versus the fully edited glyph. Black represents portions of the contours that were solid, before and after. Light blue and black is what the glyph looked like after I had removed unnecessary nodes, but had made no design changes, other than reducing the width of the source images by 20%. Red and black is the final version:



Here's the 216 point bitmap of the M, prior to a couple of minor last minute changes:



The red arrow indicates a line that is slightly off, because I chose to vertically align the node at the bottom of the contour. The blue arrow at top center has the same explanation

The blue arrow in mid center represents an error I can't correct. The line slope on the left side of the central V portion is $3/1$. And on the right, it's $8/3$. The horizontal distance between the nodes on the top of the points, and on the bottom, (baseline), does not divide exactly by 24, the lowest common denominator between 3 and 8, so the line slope on the left deviates from exactly $8/3$ by about $1/1330$.