

Quick Start Guide to the Wonderful Joy of Kerning Paths

Kerning makes characters sing and fills the font world with pleasure beyond measure. It is a totally fun task and will keep you happily engaged for endless hours of Gimping.

Well at least one of the above statements is true, but I will let you decide for yourself which one.

Text without kerning paths:



Text with kerning paths:



The accompanying resource folder contains the following items

Custom_Font_Kerning_For_UpperCase_v003b - upper-case kerning path making plug-in

Font_Slab-v0-12 – custom font writer plug-in required for the kerning filter to work

Bolonewt – a version of an uppercase white base font without kerning paths



What is a kerning path?

A kerning path is a path created with Gimp's path tool that stores the information about the width that you want the character to use, as opposed to the default action of just placing one character layer after another.

These kerning pairs can be viewed in the paths tool dock: They follow a similar naming convention to character layers and so the path 'AW.png' sets the distance between the start of letter 'A' layer and the beginning of letter 'W' layer. This replicates the local kerning feature used in real fonts.

If you come across a path with just a single letter name it indicates that the writing filter should use the amount of space between itself and all other letters: a type of global kerning. In my custom font writing filters, local kerning spacing over-rides global kerning spacing.

You can opt to add kerning paths manually or with my Custom Font Tool plug-ins; I use both methods. At the very least I would suggest that you think about adding some kerning paths for capital letter pairings. If you have a custom font with fancy flourishes it will really look odd without kerning – unless you simply place your letters one at a time in Gimp.

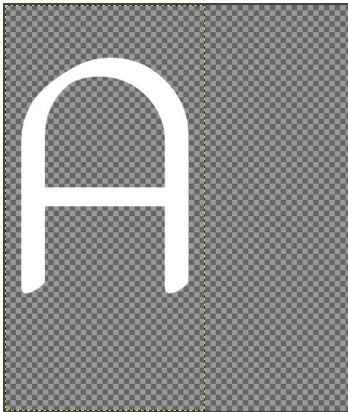
The Custom Font Tools release also contains plug-ins for adding upper to lower case kerning paths and lower-case kerning paths as well as a utility for adding accented character kerning paths for some non-English languages.

I apologise for the long-winded nature of this 'Quick Start' guide (but 'Slow Start' just doesn't have the same ring to it). It's a case of having to use a lot of words to explain something that is much quicker to do in real life. Bear with me.

Adding kerning paths manually in Gimp

The basic idea here is to make the two characters in a kerning pair visible and to slide the second character layer about until the space between them looks right. A kerning path can then be added with the path tool using a point from the beginning of the first character layer boundary to a point at the beginning of the second character's layer boundary.

Open up the font to which you want to add kerning paths and make a note of the image size. My Bolonewt example has an image size of 509 x 600 pixels.



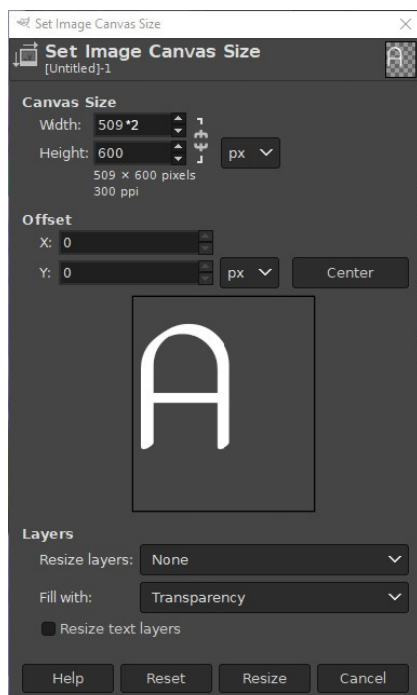
Since we will want to move the layers around it will be easier to temporarily double the width of the custom font canvas size.

Use:

Image>>Canvas Size...

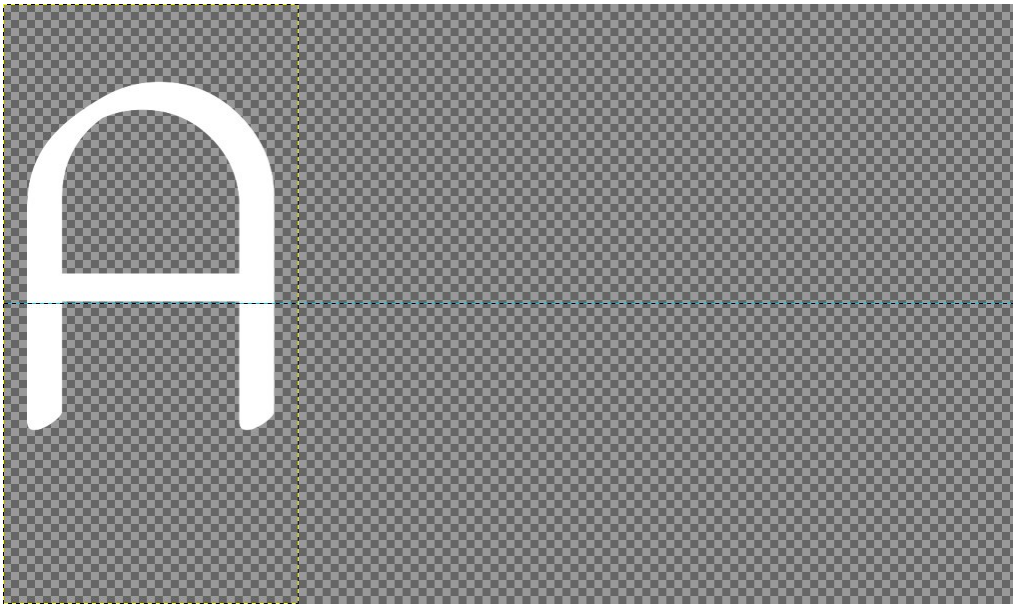
to bring up the canvas size dialog.

Ensure that the 'Fill with' option is set to 'Transparency'.



Place your cursor in the width value box and enter *2 to double the width and then click on resize.

To make it easier to slide the layers about, add a kerning path guideline by placing your cursor on the top ruler and drag it down to about the middle of the image height:



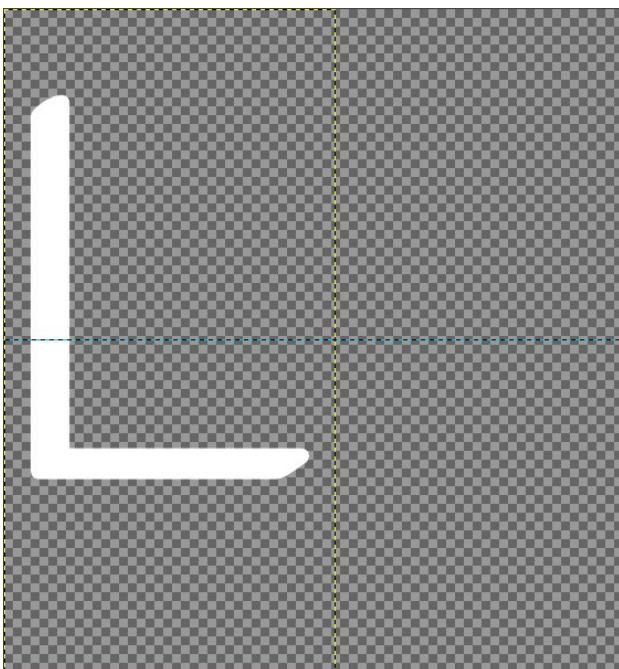
Make sure that ‘Snap to Guides’ and ‘Snap to Canvas Edges’ are checked:

View>>Snap to Canvas Edges

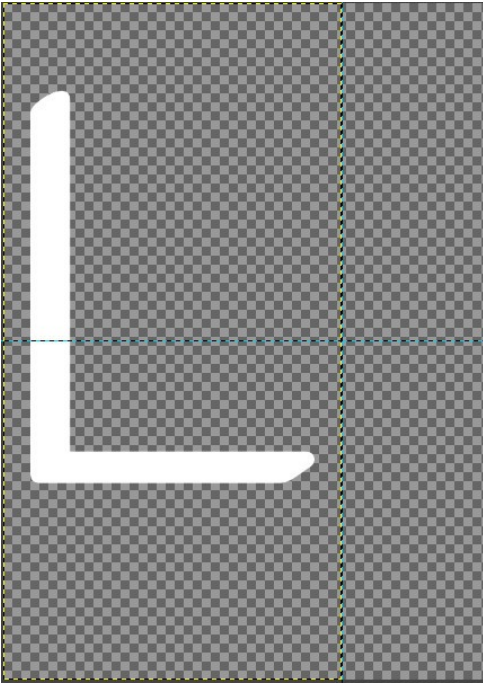
View>>Snap to Guides

Let’s add a kerning path for the ‘LT’ letter combination – a very common problem pair, so scroll down the character layers to letter ‘L’; hold down the Shift key and click where the visibility option ‘eye’ icon should be. This should make the ‘L’ the only visible layer.

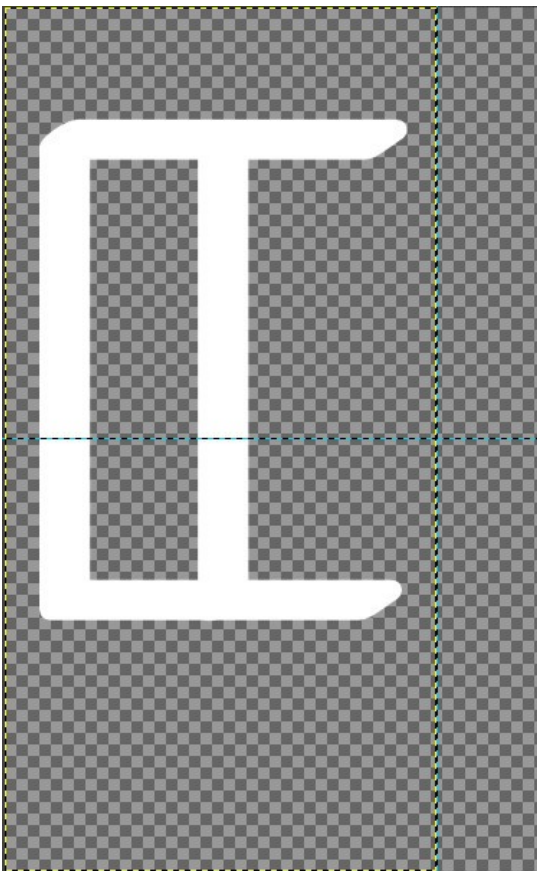
Click on the ‘L’ layer in the layer stack to make it active and so that you can see the layer boundary. (Cropped in the illustration below.)



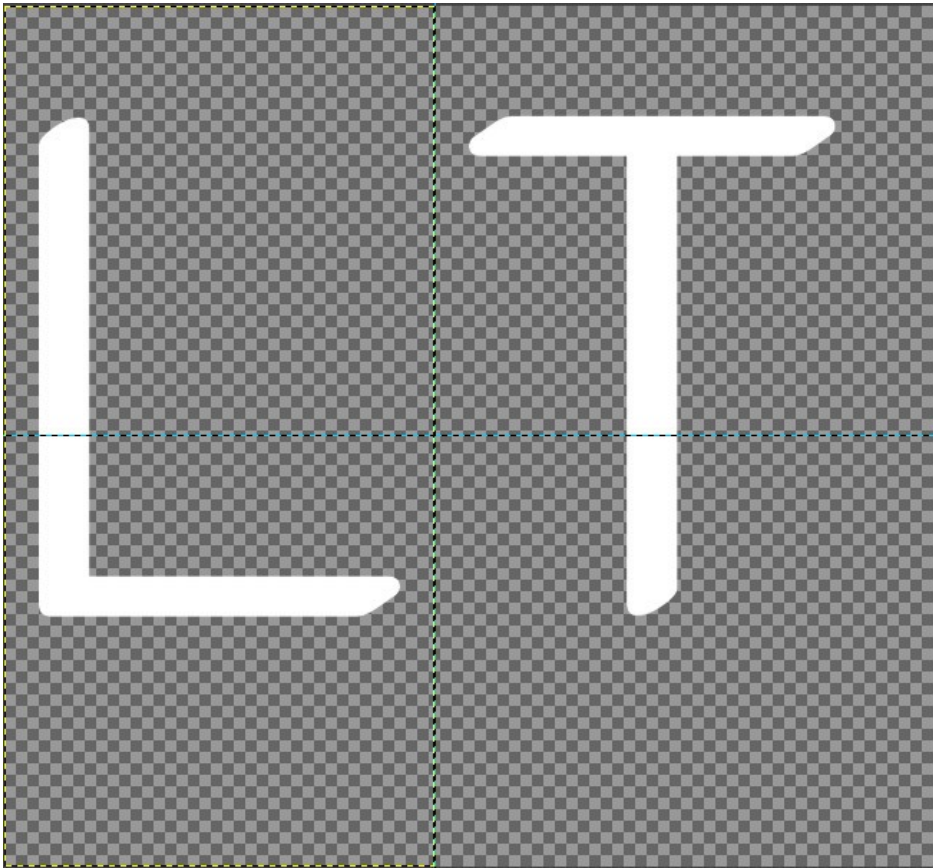
Place your cursor over the ruler at the side of the image and drag a vertical line guide and carefully place this guide on top of the layer boundary. Zoom in if necessary.



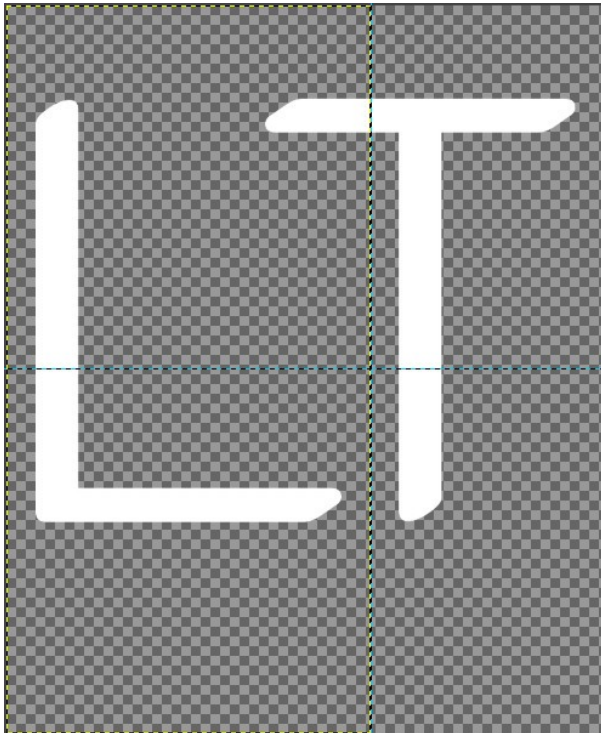
Locate letter 'L' in the layer stack and click on the eye icon to make it visible too. The two letters will probably look like this:



Now place the cursor over part of the 'T' letter and drag it away until it snaps onto the vertical guideline. This is how the characters would be displayed without kerning: A major gapping issue.



To reduce this gap, move the 'T' layer towards the 'L' layer until you are satisfied with the spacing between the two characters.

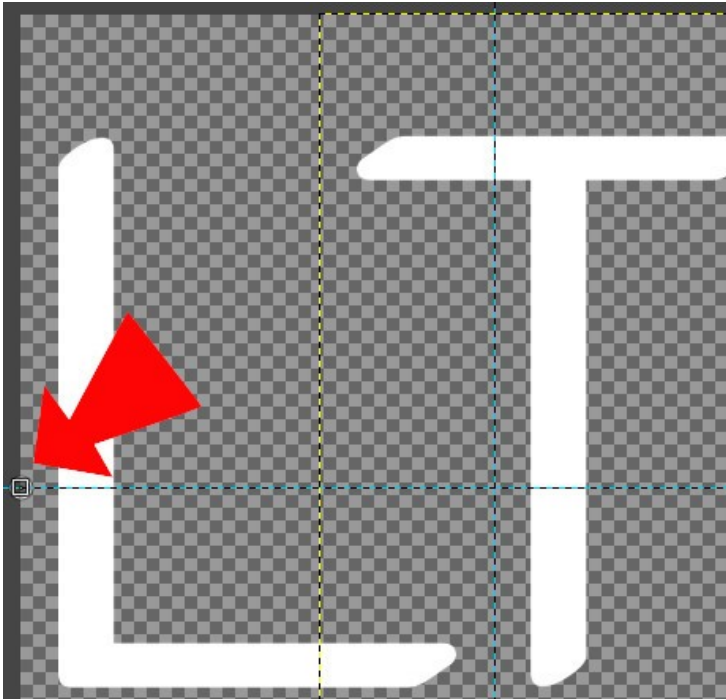


If you haven't already done so, click on the layer in the layer stack to make 'T' the active layer and so that you can see its layer boundaries.

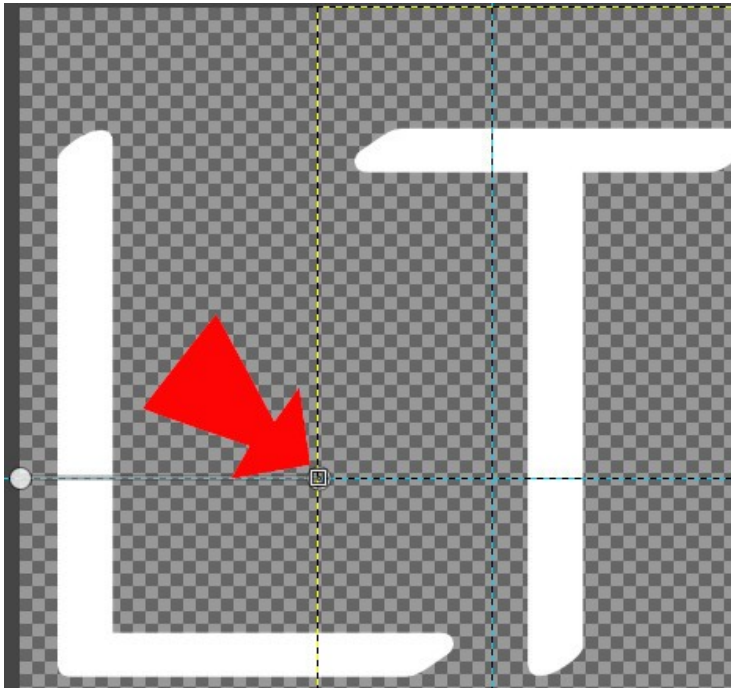
At last we are ready to add a kerning path. Click on the path tool in the gimp toolbox (or press 'B').



Click to add an anchor point at left-hand edge of the image, just above the horizontal guide-line. The anchor point should snap into place since we are using 'Snap to Canvas Edge' as well as a guideline.



Click again with the path tool at the place where the horizontal guideline meets the left-hand boundary of the 'T' layer and add a second anchor point.



That's it – the first kerning path has been created. Open the paths tab so that you can rename the path otherwise it cannot be used. (You may need to add a paths tab if you have not already done so.) I have occasionally mis-typed a kerning pair name and only discovered this later when the custom font does not display as expected. Very annoying indeed!



Double click on the default new name of the path and re-name it as 'LT.png'; your custom font writing plug-in can now use this kerning path information.



Repeat this process with all the letter combinations that you feel need some kerning path information. Use of the keyboard shortcuts 'B' to select the path tool and 'M' to select the move tool is an essential way to speed up the process.

It may sound like a daunting task but you will be surprised at how fast you can go once you build up the rhythm in your workflow.

When you have finished, you will need to return the character layers back to their starting positions to the origin of the image. You can do this manually or you can use the (7th) option from the Layer Utils plug-in; '*Ensure that all layers are positioned to the origin*'.

You can restore the canvas size by using:

Image>>Fit Canvas to Layers

Finally save your custom font to your custom font file location.

Using Custom Font Kerning Path Tools with Upper-case Letters

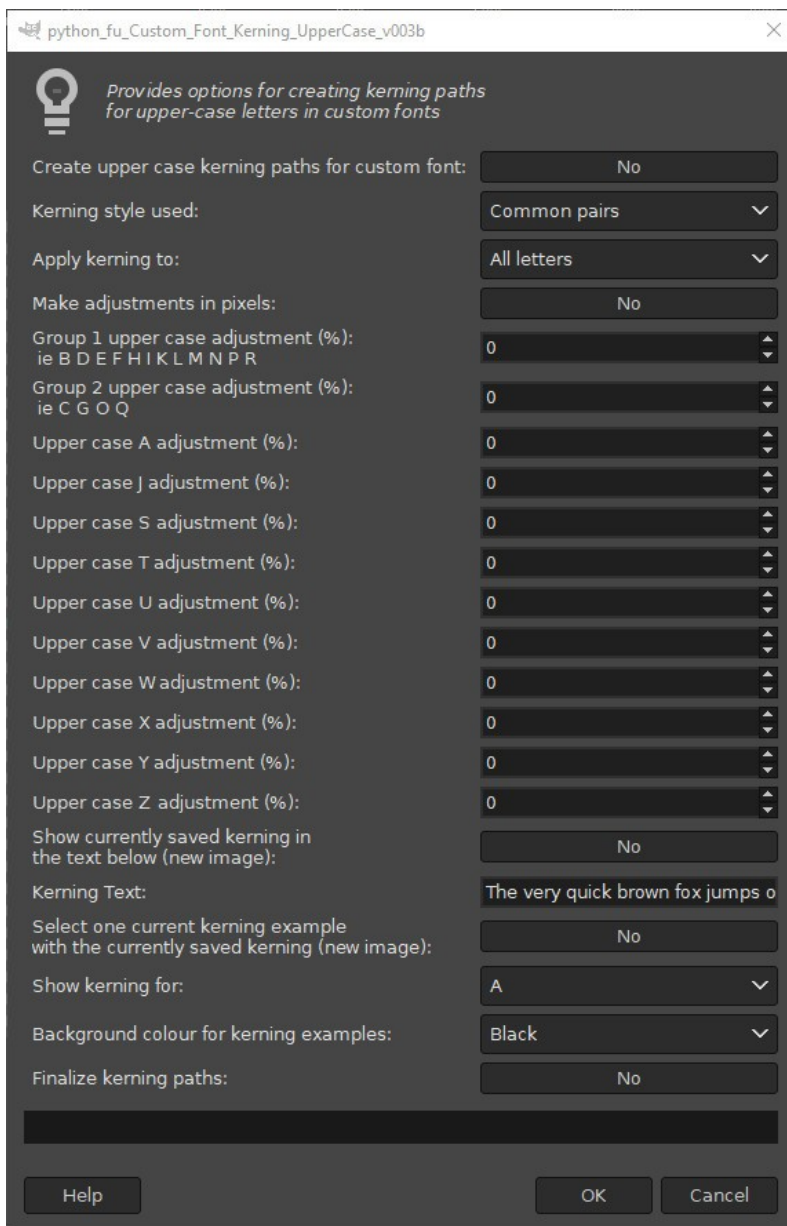
The filter works on the premiss of grouping similarly shaped letters and applying a kerning path to each letter in that set. This filter requires that the FontSlab filter is also installed; it will complain if you have not added it to your user plug-in folder.

If you have not already done so, open the un-kerned Bolonewt Whitebase custom font from your machine.

File>>Open

Call up the upper-case kerning filter:

Filters>>Custom Font Tools>>Kerning Tools>>Custom_Font_Kerning_For_UpperCase_v003b



Set 'Create upper case kerning paths for custom font' to 'Yes'
Set 'Kerning style used' to 'All pairs'.
Click on the dialog 'OK' button.

The filter will create a full set of upper-case (676 in total) kerning paths ready for the next step i.e. editing with your chosen values.

Re-save your whitebase custom font:

File>>Save (Ctrl + S)

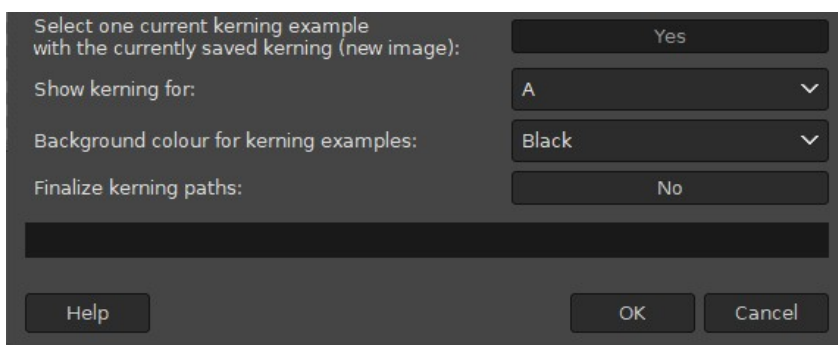
Before editing the values, we need to look at the letter pairs so call up the filter again. Let's look at the un-kerned pairs for letter 'A':

Filters>>Custom Font Tools>>Kerning Tools>>Custom_Font_Kerning_For_UpperCase_v003b

Set 'Create upper case kerning paths for custom font:' to 'No'

Set 'Select one current kerning example etc' to 'Yes'

From the drop-down menu for 'Show kerning for:', select the letter 'A'.



Click on the dialog 'OK' button.

A new window will open to display the kerning pairs.



This is the moment to decide which letter pair spacings to adjust. Normally letter 'A' needs quite a lot of adjustment with 'V', 'W' and 'Y' but this particular style needs only a little modification.

There are two options for setting values: You can use the default method of altering the amount as a percentage of the first character's width or you can alter the values in actual pixel amounts. I mainly use the default setting; I find it easier to visualise 10%, 25% etc of a letter.

Return to the original whitebase characters and re-call the filter dialog.

Leave 'Select one current kerning example etc' as 'Yes'

Set 'Create upper case kerning paths for custom font:' back to 'Yes'

From the drop-down menu for 'Apply kerning to:' select 'A' if it is not showing.

Enter the values for letter groups and individual letters as shown below:

Group 1 upper case adjustment (%): ie B D E F H I K L M N P R	0
Group 2 upper case adjustment (%): ie C G O Q	0
Upper case A adjustment (%):	0
Upper case J adjustment (%):	0
Upper case S adjustment (%):	0
Upper case T adjustment (%):	-10
Upper case U adjustment (%):	0
Upper case V adjustment (%):	-10
Upper case W adjustment (%):	-10
Upper case X adjustment (%):	0
Upper case Y adjustment (%):	-10
Upper case Z adjustment (%):	0

Click on the dialog 'OK'

Save the new values to the whitebase file each time you do this:

File>>Save (Ctrl + S)

To view the effects of the new values (which at -10% are quite subtle):

Set 'Create upper case kerning paths for custom font:' back to 'No'

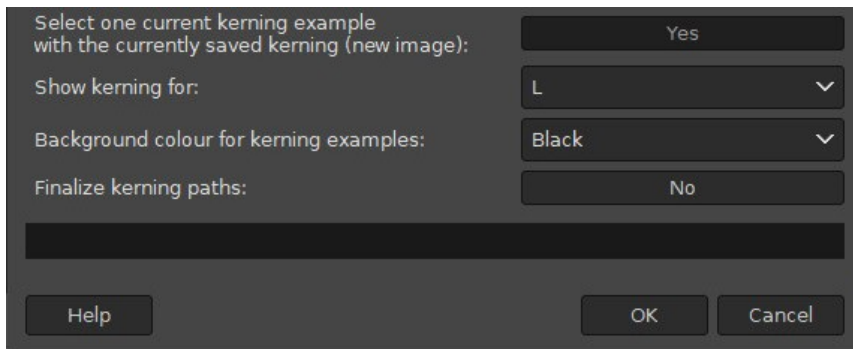
Click on the dialog 'OK' button.



For some bigger changes let's jump to letter 'L' to sort out that tricky 'LT' pairing. We need to look at the un-kerned pairs for letter 'L' to decide on the adjustments:

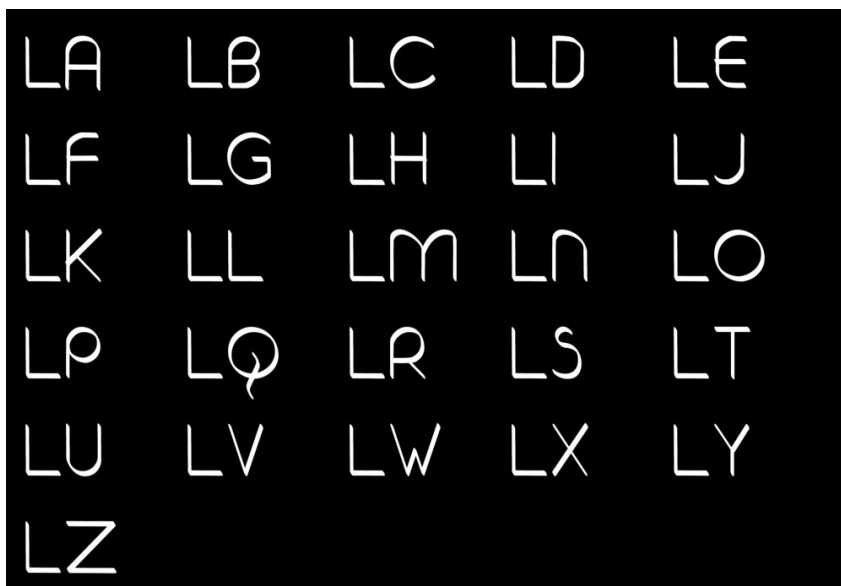
Filters>>Custom Font Tools>>Kerning Tools>>Custom_Font_Kerning_For_UpperCase_v003b

Set 'Create upper case kerning paths for custom font:' to 'No'.
From the drop-down menu for 'Show kerning for:', select the letter 'L'.



Click on the dialog 'OK' button.

A new window will open to display the kerning pairs.



I can live with everything apart from 'LC', 'LG', 'LO' 'LT', 'LU', 'LV' and 'LY'. I do not need 'LW' since I cannot think of a word that uses the two letters – but I will alter this for the sake of completeness. You may want to jot your choices down somewhere.

Return to the original whitebase characters and re-call the filter dialog.

Leave 'Select one current kerning example etc' as 'Yes'.

Set 'Create upper case kerning paths for custom font:' back to 'Yes'
From the drop-down menu for 'Apply kerning to:' select 'L'.
Enter the values for letter groups and individual letters as shown below:

Create upper case kerning paths for custom font:	Yes
Kerning style used:	All pairs
Apply kerning to:	L
Make adjustments in pixels:	No
Group 1 upper case adjustment (%): ie B D E F H I K L M N P R	0
Group 2 upper case adjustment (%): ie C G O Q	-20
Upper case A adjustment (%):	0
Upper case J adjustment (%):	0
Upper case S adjustment (%):	0
Upper case T adjustment (%):	-40
Upper case U adjustment (%):	-10
Upper case V adjustment (%):	-40
Upper case W adjustment (%):	-40
Upper case X adjustment (%):	0
Upper case Y adjustment (%):	-40
Upper case Z adjustment (%):	0

Click on the dialog 'OK'

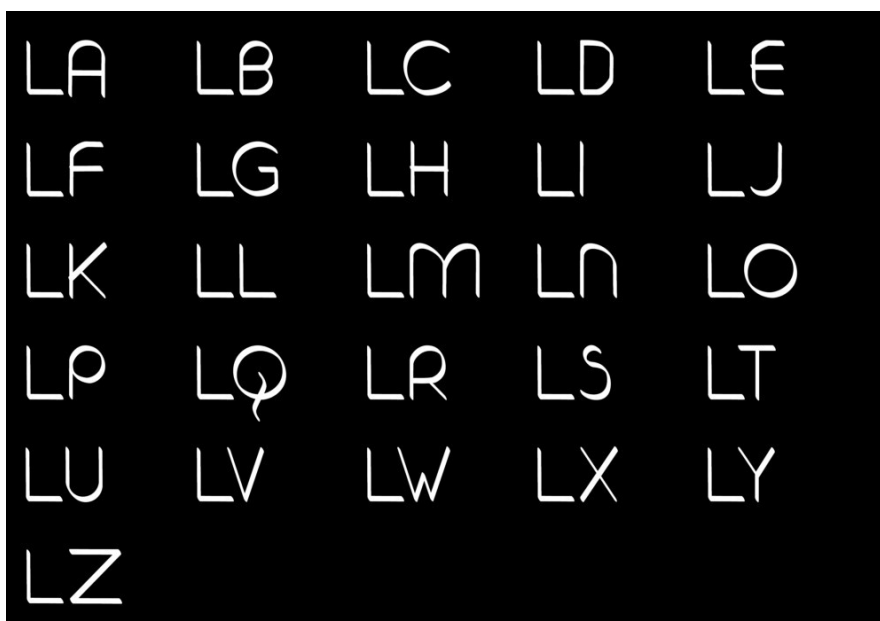
Save the new values to the whitebase file:

File>>Save (Ctrl + S)

View the effects of the new values:

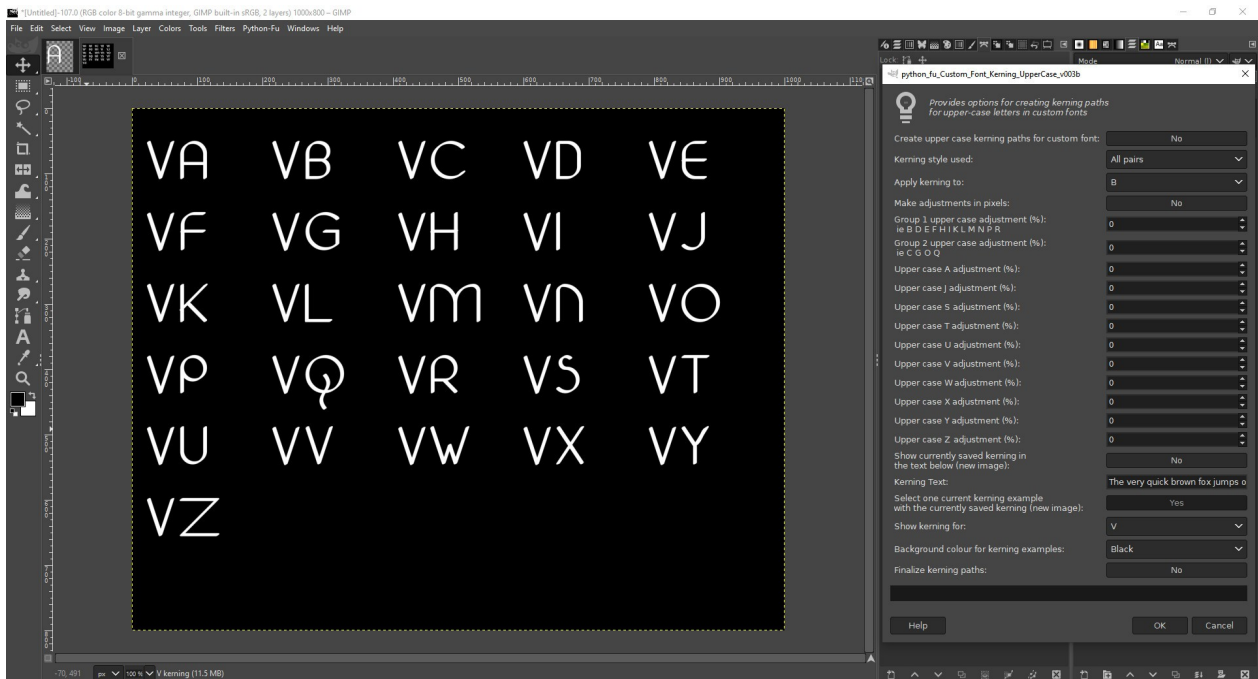
Set 'Create upper case kerning paths for custom font:' back to 'No'

Click on the dialog 'OK' button.



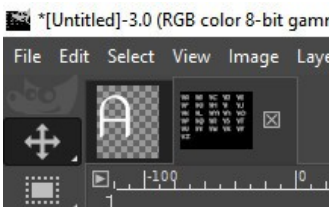
Normally you would step through each letter, in order, looking at the kerning pairs and making and saving any adjustments where necessary.

Sometimes it helps to be able to see both the letter pairings image and the filter dialog while you adjust the kerning values, as in the image below.

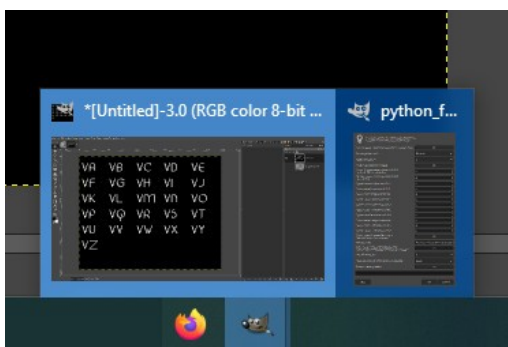


To set this up (in Windows at least) add an image for new letter pairings ('V' in this case) before setting any kerning values.

Return to the whitebase image by clicking on its thumbnail in the image dock area and call up the filter dialog.



With the dialog on screen, click on the letter pairing image thumbnail. The filter window will disappear, but you can make it reappear by clicking on the Gimp icon on the taskbar and then by clicking on the filter dialog thumbnail that appears.



This certainly helps me with estimating any new values for kerning.

So now it's just a case of repeating the process for each letter of the alphabet and, depending on the style of font, some may not need any adjustments making.

The workflow for each letter looks like this:

Call the filter from the custom font image.

In the filter dialog

Leave 'Select one current kerning example etc' to 'Yes' throughout the process.

Set 'Create upper case kerning paths for custom font' to 'No'.

From the drop-down menu for 'Show kerning for:', select the letter to display.

Leave 'Select one current kerning example etc' to 'Yes' throughout the process.

Click on the dialog 'OK' button.

View the kerning pair image and estimate the adjustment values to use.

Return to the whitebase image and re-call the filter dialog.

In the filter dialog

Set 'Create upper case kerning paths for custom font' to 'Yes'.

From the drop-down menu for 'Apply kerning to:' select the letter to display.

Click on the dialog 'OK' button.

Save the custom font (to apply these new adjustment values)

Re-call the filter dialog.

Set 'Create upper case kerning paths for custom font' to 'No'.

Click on the dialog 'OK' button to view the adjustments.

Repeat the process with the same letter until you are happy with the results.

Once again I cannot emphasise how quickly you build up a rhythm of work. Once you are familiar with the process, work for fairly short periods in between doing other more glamorous Gimp tasks.

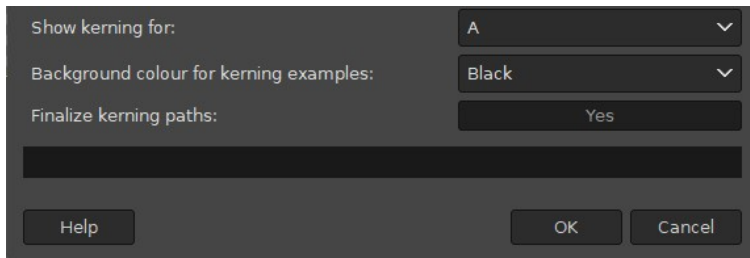
Finalizing

When you have your fully kerned set of characters there is one final job to do. You need to remove all the redundant kerning paths i.e. those that remain at the same width as the parent letter and therefore serve no purpose.

Open the filter dialog for the very last time!

Set 'Finalize kerning paths:' to 'Yes'.

Click on the dialog 'OK' button.



When finished save your custom font.

Use the Properties dialog to quickly see how many paths you have.

My custom font whitebase file properties dialogs before and after finalising show a significant reduction in the number of paths needed:

