



Creating Your Own Fonts

Part One

By Gary David Bouton

I am almost convinced that the art of typography is a dying one—or at least it has not grown in keeping with other electronic arts such as painting and drawing.

Much of the reason why there is not a proliferation of interesting typefaces is that there are only three programs that I know of that allow you to create typefaces:

- there's CorelDRAW®
- there's FontLab, which is available for Windows® only (with a Mac® version on the way)
- the prestigious URW foundry in Germany will sell you a proprietary software system for creating fonts, but with a price tag well into four figures

Hmmm ... let's explore the first two options in this article!

What's In a Font?

Surprisingly, there are a number of correlations between the product of typing on a typewriter and performing word processing and printing it.

The action on a typewriter follows this sequence:

1. You strike a key. Let's say it's the "f" key.
2. The lowercase "f" key is attached to a forged tine which strikes a printer ribbon.
3. The impression is made on a piece of paper, and the carriage is advanced by one keystroke.

Okay, on the digital side of things:

1. You strike the "f" key.
2. The scan code for "f" is sent from the keyboard to the application you're typing in.
3. The "f" appears onscreen in whatever font and size (and color) you've specified. This is a result of either Adobe Type Manager® (for Type 1 fonts) or your operating system's internal engine for displaying fonts onscreen (for TrueType® fonts).
4. You choose File|Print, and the outline information about the "f" is sent to the printer and the interior of the "f" is filled in with toner on paper.

Obviously, there are some surrogate components when you compare word processing to typing, but if we think about this one for a moment, there's an absolutely wonderful flexibility in the definition of a "font" when you use a PC.

If you had the tools, you could easily create outlines—and they don't even have to be characters—and assign them to the scan codes on your keyboard. A font can be in any language, or it can be a collection of symbols such as Wingdings or Zapf Dingbats (and actually, a legit font like Garamond is really just a collection of symbols, right?)

Let's begin this font creation exploration from within an application with which most Windows users and a few Mac users are familiar: CorelDRAW®.

Layout and Conventions

There are "rules" for correct electronic typography, some of which were handed down from traditional

typography. Let's get acquainted with them.

Grid Size for a Typeface

When creating a typeface, there naturally needs to be some limit as to the size of the characters. And there is: an imaginary grid is what you design each character within, and the grid is 1,000 units wide by 1,000 units tall.

You will not use all 1,000 units for any character. In fact, traditional designers have usually rendered characters to a 10:1 size. In other words, a 72-point (one inch) character is created on a CorelDRAW page that is 720 points on a side (10 inches). But the character doesn't occupy all 720 points, or there would be no room for leading—the space between lines of text (which CorelDRAW calls “interline spacing”).

Type Height and Position for a Font

If you'd like to try this one out, set up a CorelDRAW page at 720 points width and height. Double-click on the drop shadow on the page. In the Page Size area of the dialog box, choose points from the drop-down list, and then click on OK to get out of there. Drag the zero origin to the lower left corner of the page—accuracy is important here, so zoom in if you need to.

Now, type an uppercase character—it doesn't matter what the font is. Highlight the character and press CTRL (Mac: CMD) + T. In the Format Text dialog box, type 720 in the Size field and then click on OK. Drag the capital letter so that its node (the tiny square to the character's bottom left) is precisely on the zero, zero point on the page. Drag a horizontal guideline out of the ruler and place it precisely at the top of the character.

Surprised? I used a capital “A” in Avant Garde and the cap height is about 534 points, not 720. The font was created this way not only to allow leading, but also to provide for accents above characters, and descenders on g's, y's, q's and other characters.

If you highlight the letter again, and then type a lowercase letter, then you can drag a horizontal guideline to its top. This is called the x-height for characters in the font.

In figure 1, you can see that I've established a cap height and put the zero origin where it belongs, and now my template for creating an original typeface is all set. I'm not going to do any lowercase letters in this typeface, and when I need additional room to export more characters, I simply press PageDown and add some pages.



Figure 1

Everything on the page has to be connected

Regardless of whether the font you want to export is Type 1 PostScript® or TrueType, each character has to consist of a single path. Now this single path can have subpaths—you combine shapes using the

Combine (CTRL/CMD + L) command—but you cannot select more than one discrete object on the page and export it as a character.

I personally think this is a wonderful restriction. Why? Because for years, Fontographer® (formerly Altsys Fontographer) would allow separate, overlapping shapes to be exported as a single path in a typeface. This caused numerous problems with a lot of shareware authors who didn't know any better, and a lot of PostScript imagesetters would fault on a print due to the incorrectly designed character.

You might want to design stems for your font as separate pieces, and then use the Weld command to join them to the body of different characters to maintain a consistency with the look of the font. This is what the pros do.

In figure 2, you can see the wrong way and the right way to create a character.

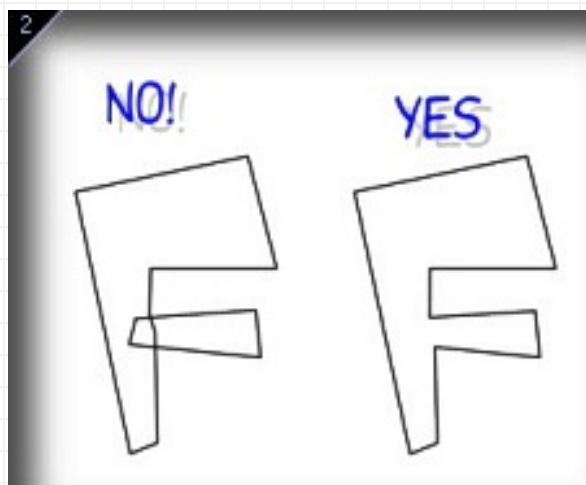


Figure 2

Don't Intersect Paths

Simply because a guy can leap off a cliff and survive (and Cliff survives too) doesn't mean he should. Similarly, because you can overlap closed paths and combine them when making a font does not mean that you should. This is sloppy work in TrueType fonts and this will cause a PostScript error in Type 1 fonts.

In figure 3, you can see a preview of the two shapes and what they look like when combined. On the bottom in color is a recreation of this shape where the two colored parts of the character are actually made up of two different shapes, positioned very closely together (then combined—shapes do not have to be inside one another to use the Combine command).

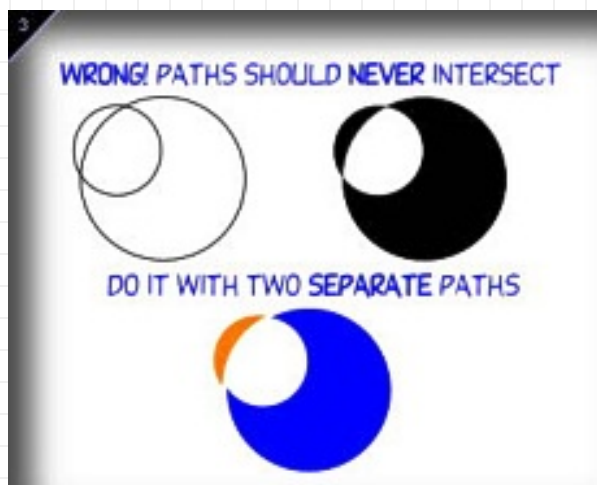


Figure 3

Mapping the Typeface Out and Exporting It

Let's talk a little about one or two more components of individual characters in a font and how you can squeeze all the professionalism out of CorelDRAW to make your font work the way you anticipate.

- **“Conciseness” is an important element with each character.**

In other words, do not go placing control points (nodes) all over the path of the outline font for no good reason. The character O, for example, can be described using only eight control points—four for the outer path and four for the inner subpath.

Extra control points needlessly complicate the character at output time, and also places a stress on an application to draw the font for the screen. I have an ornamental typeface from a commercial manufacturer that is so needlessly intricate that it absolutely will not render from within Photoshop.

As a general rule, PostScript fonts can't have any more than 200 control points per character.

- **Types of connections can lead to bizarre results when you use the font in a modeling program such as Xara 3D.**

Make sure that there are at least two control points at extremes within a character and make sure the control handles aren't going off into outer space or anything. Control handles, technically, are off-path control points, but you don't want control points too far away from the path—lay down an intermediate control point if necessary. You will get “spikes” in areas of a font if only one control point is used to support the vertex of two huge arcs.

If a curve should pass smoothly through a control point, then the type of connection for the control point is smooth, natch'. If your character is only straight path segments, then there should be no control handles—every control point should lay on the path.

- **If you're a “fontaholic” like me, you will know that your audience will definitely appreciate extended characters in your typeface set. Cent signs, fractions, curly single and double quotes, guillomets, accented e's ... you know what I mean.**

To find the proper slot in the typeface for these extended characters (in Windows), open the CharMap.exe utility, click on the symbol, and then remember what the number is that displays in the lower right of the applet.

For example, a cent sign is typed on a page by holding ALT and typing 0162 on the numeric keypad. Therefore, to add a cent sign you've designed to a font, choose 162 from Corel's Character Number list.

On the Mac, you can [download Character Chooser](#) for free from Letraset®. This utility will tell you what keystroke combinations are needed to access an extended character.

- **A character must be a closed path to be exported. Period.**

I'm getting ahead of myself here. First, you make all those nice characters on multiple pages in CorelDRAW. Then go back to page 1, click on the character to select it, click on the Export button and choose TrueType (*.ttf) or Type 1 PostScript (*.pfb) from your list of selections. Click the Selected Only checkbox. Click OK, and you'll see an Options dialog.

Let's pretend that this font is going to be a Type 1 (I'll explain some stuff about TrueType shortly). In figure 4, you can see the dialog box. The Family name will be the name that will appear in applications under their font menus. I'm naming this font “Whimsy”.

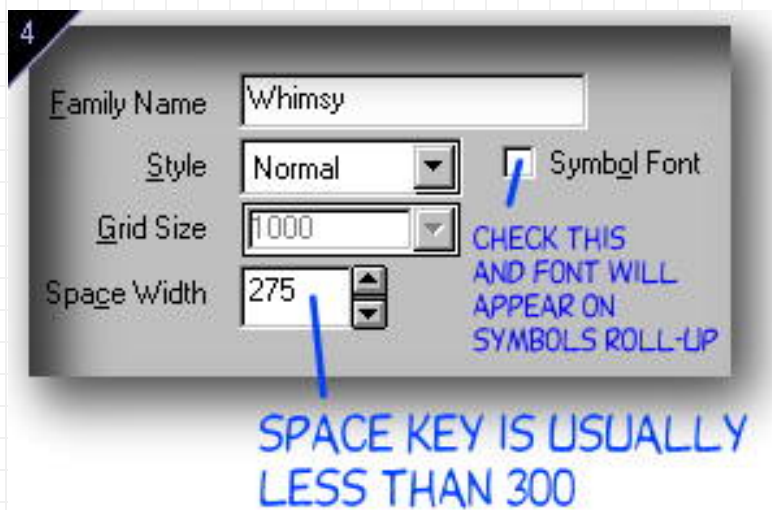


Figure 4

In Style, “Normal” is the only real option. CoreIDRAW presumes that you want to create bold, italic, and bold-italic members of a font. Me? I only want to do a lighthearted display font, so “Normal” is the correct choice.

The Grid Size is set to 1,000; this requires no change.

If you check the Symbol Font checkbox, your typeface will appear on the Symbols roll-up/dock. So think about this one—if you have not designed a symbol-type font, don’t check the box.

Finally, there is practically no typeface on earth that uses a space key set to 1,000 units. In figure 4, I’ve typed a more realistic 275 in the Space Width field. About 300 is the width of the space key in normal, well-adjusted fonts.

Click on OK, and then click on OK in the Save Changes to Font File dialog box.

We’re off and running now, and by default, the first character slot CoreIDRAW wants you to fill is the exclamation point. Um, I didn’t design this Whimsy font with an exclamation point as my first design effort. I started with “A”, and if you did too, you scroll the symbol box over to A.

In figure 5, you can see not only the character as it will be exported (I’m up to ‘F’ now), but a few other measurement items you should be aware of before exporting a character.

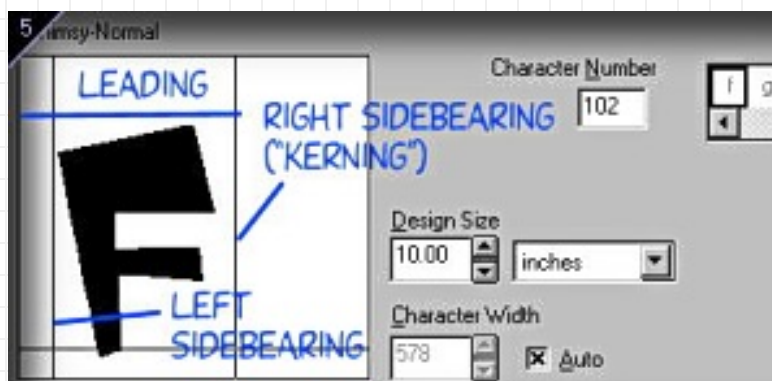


Figure 5

Examining the Character Export Box

If you made sure that the part of a character that is furthest left of the page just touches the left vertical guideline, then your character will never accidentally touch a preceding character. This left vertical line in the export box can be considered the left sidebearing ... but not exactly.

You see, in typography there is padding around characters that ensure that, for example, the right side of a capital “T” doesn’t smash into the following “H”—the stems on the “T” are taken into consideration with left and right sidebearings.

CorelDRAW doesn’t call the space a sidebearing, and it’s not truly a measured sidebearing ... however, the padding you see in the export window *will* help to keep characters from overlapping.

Similarly, the “right width” vertical line can be thought of as either a right sidebearing or a kerning measure to keep the following character from overlapping it. Again, I’m applying professional typography terms to the CorelDRAW export box, which doesn’t truly support these features ... but we can pretend, and the results work as though CorelDRAW had these options.

Finally, take a look at where the cap height is back in figure 5. There’s plenty of room for the next line of type (the space between lines of type is called leading), because the character only takes up about 575 units of the imaginary 1,000-unit grid. By the way, if you uncheck the Character Width box, you can manually drag the right sidebearing vertical line closer to or further from the character.

NOTE: CorelDRAW automatically advances the Character Number by one to make it convenient to export one character after another in sequence. Be aware of this if you are skipping around in the character registers!

Continue to ***Part Two***.

Gary David Bouton is a wise-aleck, illustrator and author of 15 books on computer graphics (and the guy isn’t even close to 50 yet!), most notably “Inside Adobe Photoshop 5” by New Riders.

Disclaimer: The information provided on Designer.com is intended to be general information. Articles reflect the opinion of the author and do not imply sponsorship, endorsement or approval by Corel Corporation.

[Return to the page-by-page view](#)



Last modified on: 5/29/01
[Terms of Use/Legal Disclaimer](#)

Copyright © 1999 - 2000 Corel Corporation. All rights reserved.
Company and product names are trademarks or registered trademarks of their respective companies.



Focus

Articles

Creating Your Own Fonts

Part Two

By Gary David Bouton

Ah, the Problems with PostScript

The following isn't a craftsmanship or consistency thing, but instead, it's a **rule** ... specifically, the **Winding Path Rule**. If you do not obey it, funky stuff will happen when you use your own Type 1 fonts in certain applications.

The Winding Path Rule goes as follows: if you were walking an imaginary line around a character, the fill for the character would go to the left. This more or less means that outline for characters should be traveling counterclockwise and that combined subpaths should be traveling clockwise. See figure 6.



Figure 6

So if you're using CorelDRAW 9 to design your fonts, make sure to set the path direction properly. You can do this using the "Reverse Curve Direction" button that appears on the Property Bar when you select the Shape Tool.

Now, some of you might ask, "Okay, so how do I set the outline vector direction for a path in older versions of CorelDRAW?" It's simple: you don't—it's not a feature.

To be fair, CorelDRAW has much better font creation tools than most font creation programs; it's simply that outline direction is a feature in Adobe Illustrator® and Macromedia® FreeHand™ because PostScript output on the Macintosh was a "must", so correct path direction was a mandate. Over the same period of years in the past, Windows users have not been too concerned about path direction because Hewlett Packard non-PostScript printers did not care about path direction—it was not a feature of the printer language.

Hang in there, and I'll tell you how to correct a Type 1 font created in older versions of CorelDRAW a little later.

TrueType is No Problem

If you choose to export a font to TrueType format, you will suffer none of this Winding Path nonsense, because for reasons I do not know about, CorelDRAW exports TrueType with correct path and subpath directions. However, this also means that if you've created a Type 1 font and use a utility to convert it to TrueType, there could be path errors.

By the way, use the default settings in the Export box for TrueType. You'll notice that the grid size is larger: 2048. Not to worry—TrueType increments are smaller in dimension than Type 1s, and both formats will generate a font of the same average point size.

The Path Direction Follies

This is a short section, but I'd like to share with you one of the results of creating and using a font with incorrect path directions. In figure 7, the outline is running in the right direction, but the subpaths are going in the same clockwise direction to fill the front face of the font and the outside edges.

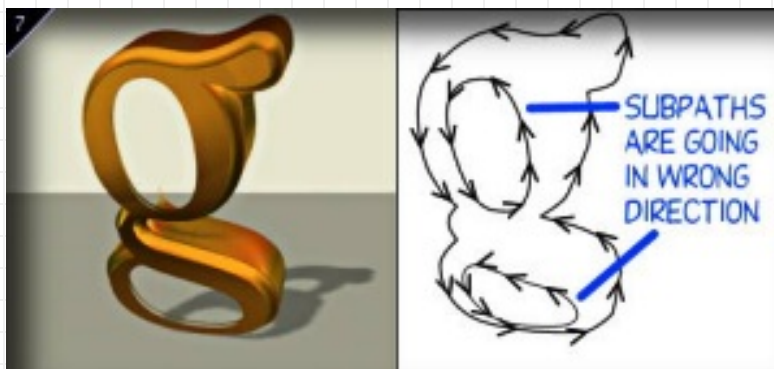


Figure 7

What's missing here are the inside edges of the cutouts in the character "g". By the way, the application used for these figures on the left side is PIXAR Typestry, an extinct product, but you'll run into the same problems with current modeling or font extrusion programs (such as Xara 3D).

In figure 8, the outline path is going in the wrong direction, but the subpaths are traveling in the correct direction. So the front side of the "g" and the interiors of the "holes" in the "g" are rendered, but the outside extrusion is missing.

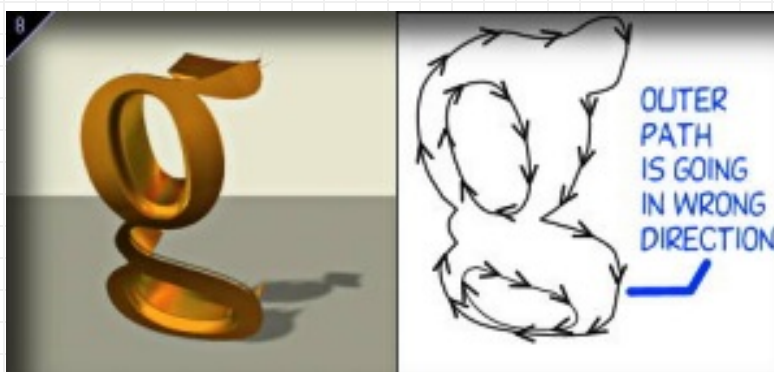


Figure 8

Figure 9 hits the jackpot. The paths and subpaths are all going in the right direction, and you can see every facet of the extruded gold letter.

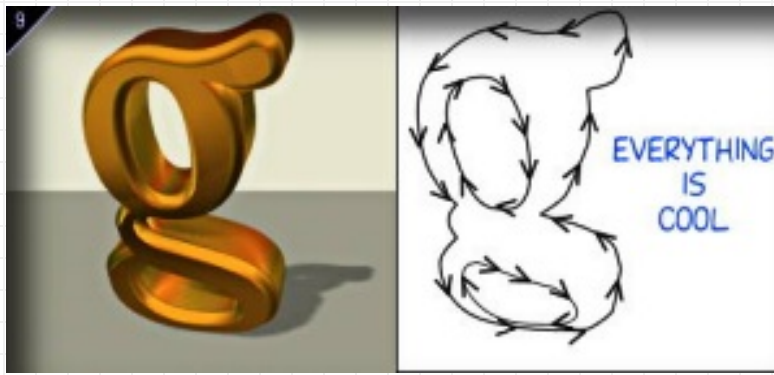


Figure 9

Options for Correcting Path Direction

Because of the new Reverse Curve Direction feature, CorelDRAW 9 users can correct path direction on their own. So those of you using that version can skip this next section.

For users of older versions, there aren't a whole lot of things you can do if CorelDRAW generates Type 1 outlines in reverse direction. Here are the options I use:

- I could use only TrueType. But this is like only eating apples when there's a world of bananas out there.
- I export a character to Adobe Illustrator 7 for Windows, and change the path direction through the Attributes palette. Then, I import the character outlines back to CorelDRAW for export.
- I use FontLab, a program that's actually written in PostScript. Pyrus Software sells the product for under \$400, and if you're serious about font-making, it is the only choice for Windows users. Originally founded as SoftUnion when Russia was still the U.S.S.R., these four engineers know PostScript at least as well as Adobe engineers do.

(For Macintosh users, I'd try to beg a copy of Fontographer out of Macromedia. There also might be some shareware programs out there that will inspect and correct a character.)

Figure 10 shows a small area of FontLab; I've got a character outline from a font in the editing window, and I'm correcting the curve direction. One of the nice things about FontLab is that you do not have to extract a character from a font face to correct it. You perform your surgery on a single character, and then export the font as though you were creating a whole typeface in FontLab.

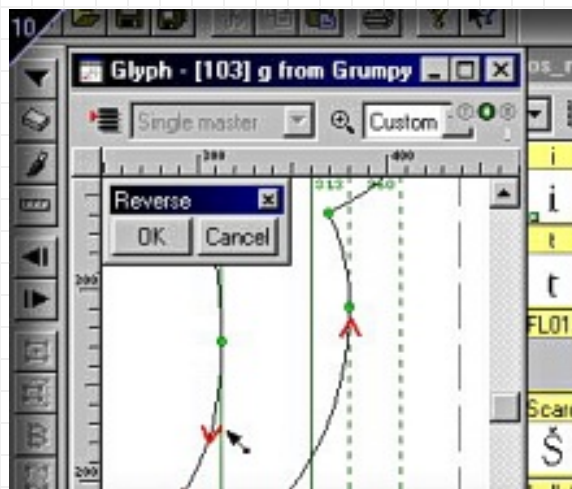


Figure 10

So why not do all your fonts in FontLab 3.0? Simple: I don't believe the tools and the design interface are nearly as intuitive as CorelDRAW. You create in CorelDRAW and then tweak the font in FontLab. It is here that you can add genuine sidebearings, kerning pairs between specific characters, hinting (which is the changing of a font outline when a small point size is specified) and other professional touches.

A Parting Gift

Because you've been such patient folks as I discuss the trivia of the typeface, please feel free to download and use the font "Whimsy" by clicking on its name here. It's TrueType Windows only, but Macintosh users can perform the conversion by using TT Converter, a shareware utility commonly available at Apple and mirror sites.

Figure 11 shows a Windows style sheet for the font.

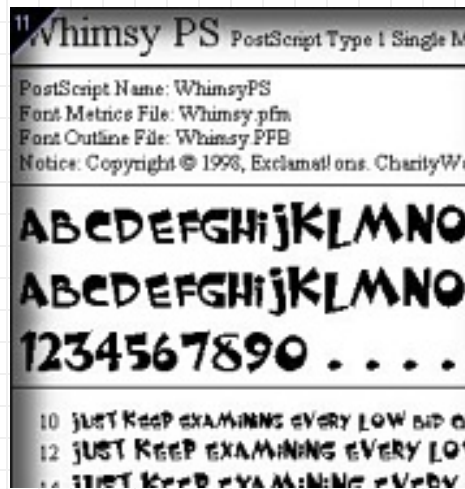


Figure 11

Conclusion

Font creation doesn't have to be a dying art if you don't want it to be. Imagine the thrill of using your own font on a site, a sign, or simply in a letter to a friend. The best thing about creating fonts is that you can share them. There is no restriction on something you created yourself!

Gary David Bouton is a wise-aleck, illustrator and author of 15 books on computer graphics (and the guy isn't even close to 50 yet!), most notably "Inside Adobe Photoshop 5" by New Riders.

Disclaimer: The information provided on Designer.com is intended to be general information. Articles reflect the opinion of the author and do not imply sponsorship, endorsement or approval by Corel Corporation.

[Return to the page-by-page view](#)



Last modified on: 5/29/01
[Terms of Use/Legal Disclaimer](#)

Copyright © 1999 - 2000 Corel Corporation. All rights reserved.
Company and product names are trademarks or registered trademarks of their respective companies.