

All About Outline Fonts (Part 3)

In his concluding article about fonts, Kell Gatherer looks at font design and the use of fonts in publishing.

FONT DESIGN

Font design is a labour-intensive task, and has as much to do with art as computing. There are some applications available which can help however. Acorn's FontEd was mentioned in the last article, and remains the only serious piece of available software dedicated to font creation. Regrettably, you can't import Draw files into FontEd, but there's an excellent application called D2Font from David Pilling, which converts Draw files to outline fonts. Along with D2Font comes an application called Trace, which converts sprites to Draw files - you can therefore scan fonts, drag them into Trace and once converted drag them into D2Font to create a font. In fact, that process is very time-consuming, and you'll still find a lot of tweaking to be done in FontEd at the end.

Getting used to FontEd can take some time, but it does make extensive use of the interactive help system which can provide quite a few clues to some of the more obscure facilities. It is easy to copy characters from one ASCII position to another, meaning you can create an E from an F for example. A sprite of a scanned character can be dragged into FontEd as a template to assist with the creation of a character. Also, because FontEd allows you to have more than one font loaded at a time, it is possible to borrow characters from an existing font and import them into the one you are creating, by dragging from one window into the other. You can even drag a character from one window to another, alter its design size, and drag it back again - so having created the numerals 1 and 2, you can reduce

them and merge them to create the 1/2.

FontEd's path editing capabilities are in fact fairly basic, and D2Font allows you to make proper use of the better facilities in Draw (or even ArtWorks, exporting in Draw format). Really, this has to be seen as the only proper way to get geometric designs such as concentric circles into a font.

A RISC OS 2 font (version 6 or earlier) can have a maximum of 223 characters - ASCII 32 to 126, and 128 to 255 (character codes 0-31 and 127 are not printable). Only characters below 127 and above 139 have specific definitions in the Latin 1 alphabet, and a lot of the characters above 139 are in fact composites - for example the double-dot that appears above a vowel in certain European languages is known as the dieresis, and appears as ASCII 168. Character 246 is o-dieresis (ö), and character 255 is y-dieresis (ÿ). Rather than create a new outline for each of these characters, the font format allows a character to make reference to others that are already

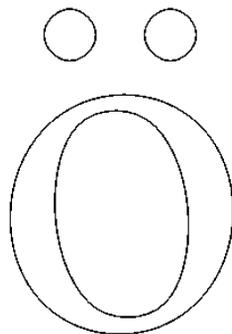


Figure 1.
o-dieresis created by merging an o and a

defined - in other words the o is taken from the normal o (ASCII 111) and the dieresis from the dieresis (ASCII 168). This is done by Shift-Adjust-dragging characters in FontEd, but the effect is to reduce the overall file size and save a lot of design time. The plus-minus sign is thus simply and quickly created from the plus and minus signs you've already designed, and you'll probably find yourself using the full-stop time and again in the definitions of other characters.

The font specifications have been changing over the past few years, and the latest RISC OS 3 fonts are now version 8, which may have many more characters defined. In RISC OS 3, if you load Homerton.Medium into FontEd from

ROM (click menu on the Apps folder and enter the Fonts directory) you'll see that there are actually 416 characters defined, and that a considerable number of them are not where you would expect them to be. This is because the RISC OS 3 font definitions can make reference to separate encoding files which can alter the positions of characters according to the country of usage.

RISC OS 3 fonts use a considerably more complex file structure than the earlier versions, and we can be thankful that the Font Manager copes with all variants invisibly. RISC OS 3 fonts are now becoming more readily available, although the only discernible difference from their predecessors is in the provision of kerning tables.

At the time of writing, few software applications can display auto-kerning, but one is the RISC OS 3 version of Draw. However, you cannot enter such text directly. On this month's

Trinity:
 unknerned: **AVIATION**
 kerned: **AVIATION**

Figure 2.

magazine disc is an application called Kerner which demonstrates this facility - you may enter a text line in Draw, drag it to the Kerner icon and then back into Draw. The resultant text line will show auto-kerning in action (if indeed there is kerning information for any of the character pairs). Kerner works by converting an ordinary text area object (object type 9) into the new RISC OS 3 transformed text object (object type 12). Setting bit 0 of the font flags in this object type kerns the text.

ITALICS AND OBLIQUES

Many typefaces have an associated italic version, which is usually an entirely different alphabet in a similar style, and which must be created separately. An oblique version is in fact exactly the same alphabet which has been made to lean to the right - see figure 3. Another excellent application which helps with the

creation of an oblique version is the misnamed *Italic* from Design Concept. Having created your font, *Italic* quickly and painlessly creates an oblique version in the same directory.

THE PRODUCTION OF ARTWORK

There is now a profusion of public domain fonts available for the Archimedes, with a wide range of headline and fancy fonts becoming available, and some suitable for body text. In certain applications, however, this is not necessarily an advantage. The serious application of DTP has to be seen as the production of camera-ready artwork for printing, via an image-setting machine at a DTP bureau. The output from an image-setter is of very high quality indeed, normally 1270 dpi, and will appear on a kind of photographic paper called a bromide. Such image-setters make use of the PostScript image language, and generally have copies of hundreds of commercial fonts in memory.

Because the font held in the image-setter is a separate entity to the font in your Archimedes, the relationship between the two is crucial. If a PD Archimedes font calls itself Garamond, it is entitled to do so, because the copyright on that particular typeface expired some time ago. But that is not the same as saying that it exactly corresponds with the Garamond font produced by, let's say, Monotype, which may be resident

Roman: The Quick Brown Fox
Oblique: The Quick Brown Fox
Italic: The Quick Brown Fox

in the PostScript printer. The differences may be practically invisible to the eye, but tiny variations in the widths of individual characters may result in a line of type spilling over the end of the right hand margin, making full justification an impossibility.

Anyone wishing to design fonts for the Archimedes should bear this in mind - you can only claim that a font is PostScript compatible by paying a licence fee to the originators of the font and gaining access to the exact character

continued on page 62

Figure 3.
 The difference between Italic and Oblique