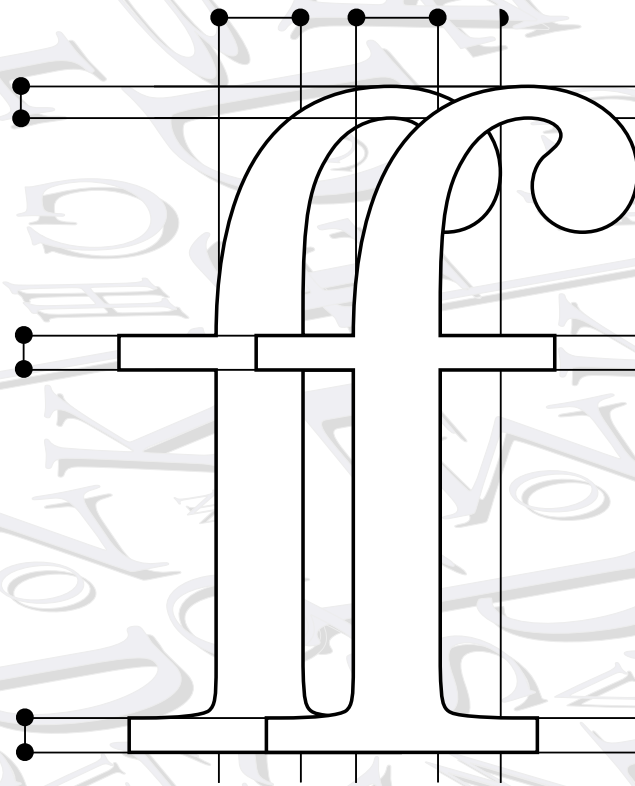




Fontfiend

The
Fiendish
Font
Editor



iSV
Products

3QD
DEVELOPMENTS LTD

APDL

FontFiend

© 3QD Developments Ltd 2026

All rights reserved

No part of this product may be reproduced in whole or part by any means without written permission of the publisher. Unauthorised hiring, renting, lending, public performance, broadcasting or networking of this product or its constituent parts is prohibited.

While every care is taken, the publisher cannot be held responsible for any errors in this product (either within the software or the documentation), or for the loss of data or consequential effects from the use of this package

This Documentation supplied by APDL

a part of

3QD Developments Ltd

Single User version downloaded from the free of charge software section of
www.riscos.com

If you would like to make an optional donation to help extend the range of free software we provide you can do so using the above link, thank you

Please note that any free of charge downloads from RISCOS.com are supplied without any technical support package as per the licence agreement.

This text last updated 29-09-26



If you have found this application and this manual useful please consider making a donation to help cover the costs of cataloguing, documenting, updating and hosting the APDL applications. Scanning the QR code will take you to our donations page.

Contents



Chapter 1 What is a font

Acorn Format fonts	1.1
Outlines	1.2
Composites	1.3
Skeletons	1.5
Widths	1.5
Scaffolds	1.6
Bounding boxes	1.8
Kerning	1.9
Misc Data	1.10

Chapter 2 Getting Started

Setting up memory	2.1
Loading FontFiend	2.1
Loading a Font	2.1
Looking at characters	2.3
Filled character windows	2.3
The FontFiend main menu	2.4
Character menus	2.4
Filled character windows (2)	2.4
The icon bar menu	2.5
Getting started summary	2.5

Chapter 3 FontFiend Tutorial

Loading the example font	3.1
Adding the missing letters	3.2
Adding skeletons	3.3
Adding scaffolds	3.4
Scaffolding the entire font	3.6
Saving the finished font	3.7
Making a bolder version	3.7

Chapter 4 FontFiend main table

Dragging with Select	4.1
Dragging with Adjust	4.2
Exporting characters as drawfiles	4.2
Importing characters as drawfiles	4.2
Importing other files	4.3
Importing AFM files	4.3



Importing Encoding files	4.4
Transformations	4.4

Chapter 5 FontFiend main menu

The Misc submenu	5.1
Misc font information	5.2
Misc data	5.2
Misc data offsets	5.3
Encoding	5.3
Display (rendering preferences)	5.4
The Bounding sub menu	5.4
Bounding Calculate	5.4
Bounding global	5.4
Set to zero	5.4
The characters sub menu	5.5
All active	5.5
Characters copy	5.5
Characters delete	5.5
Char build	5.6
Char build stroke width	5.6
Italic offsets	5.7
Overwrites	5.7
Misc data button	5.7
Encoding button	5.8
Small caps	5.8
Building characters	5.8
Transform	5.9
Transformation examples	5.10
The Composites sub menu	5.11
Delete	5.11
Make real	5.11
The save sub menu	5.11
Accents	5.11
Composites	5.12
Font	5.12
Kerning	5.12
Scaffolds	5.12
Skeletons	5.12
Postscript	5.13
The Outlines sub menu	5.13
Change to curve	5.13

Change to line	5.14
Delete all	5.14
FX Process sub menu	5.14
Clean points	5.15
Flip X	5.16
Flip Y	5.16
Special FX	5.17
Make Open	5.17
Alter Weight	5.17
Make Ribbon	5.18
Thin Shadow	5.19
Combination effects	5.19
Smooth	5.19
Gridlock	5.19
Alter X Pos	5.21
Alter Y Pos	5.21
The Kerning sub menu	5.22
Delete	5.22
Edit	5.22
The kern pairs menu	5.22
Create	5.23
Delete	5.23
Next Pair	5.23
Editing kerning	5.23
Global kerning	5.23
The scaffolds sub menu	5.24
Automatic	5.24
Scaffold model	5.24
Global scaffold settings	5.26
Connect	5.28
Make Local	5.28
Make Tree	5.29
Delete all	5.29
Delete index	5.29
Disconnect all	5.29
X & Y Scaffolds	5.30
Check integrity	5.30
Repair Tree	5.30
The skeleton sub menu	5.30
Automatic	5.30
Convert to outlines	5.32





Delete	5.32
The widths sub menu	5.32
Mono Digits	5.32
Remove LSB	5.32
Set to Bbox	5.33
Summary	5.34
 Chapter 6 Outline character windows	
Bounding boxes	6.2
Composites	6.2
Scaffolds	6.2
Outlines	6.3
Skeletons	6.4
The currently selected character	6.4
 Chapter 7 FontFiend character menus	
Bounding sub menu	7.1
Calculate	7.1
Display	7.2
Centre to width	7.2
Composites sub Menu	7.2
Delete	7.2
List	7.3
Make Real	7.3
Save	7.4
Outlines sub menu	7.4
Scaffolds sub menu	7.4
Automatic	7.4
Connect	7.4
List	7.4
Manual	7.5
Make local	7.7
Delete all	7.7
The rest of the menu	7.7
The Skeletons sub menu	7.7
Transform	7.7
Width	7.7
Zoom Sub Menu	7.7
in	7.8
out	7.8
1:1	7.8

**Full Char
Summary**

**7.8
7.8**



Chapter 7 The Preferences Window

Save Intmetrics with Outlines	8.1
Protect font when saving	8.1
Debug outlines on loading	8.1
Default encoding	8.2
Display char names	8.2
Re-calculate bounding boxes	8.2
Auto-generate scaffold tree	8.2
Notify character overwrites	8.2
Enable on-line help	8.3
Font Rendering Preferences	8.3
Render Outline control points	8.3
Render Skeleton control points	8.3
Render scaffold lines	8.3
Render outlines filled grey	8.3
Setting preferences	8.4

Appendix 1 Glossary of Terms 9.1

Appendix 2 Font Problem Solving 9.8

Appendix 3 Encoding Tables 10.1

Appendix 4 Loading Base 0 fonts 10.6

Appendix 5 How do I....? 10.7

Appendix 6 A page for your notes 10.10



Chapter 1

What is a font?

At its simplest form a font can be thought of as a collection of characters (not necessarily letters) that can be drawn at any size either on screen or to a printer. These characters are very similar in construction to draw files. They contain combinations of either straight lines, curved lines (called beziers) or move commands. The shapes that make up the characters in a font are filled with the required colour when they are drawn by the computer.

However a modern computer font actually consists of far more than just the characters, it also has details on the width of each character, its size and a host of other information. When a font is broken down into its constituent parts we find that a font is actually made up of a number of components. When these components are combined we have a full working font.

These components are best split up as follows:

- Outlines & Composites (The drawings of the characters)
- Skeletons & Scaffolds (Hinting)
- Widths & Bounding boxes (sizes of the characters)
- Kerning (adjustment to the widths in special circumstances)
- Misc data

Acorn format fonts

Acorn fonts are actually made up of two files, one is called Outlines and the other Intmetrics. They are stored in sets of directories that mimic the font name.

For example : ADFS::4.\$.!Fonts.**iSVSans.Condensed**.Outlines

The Outlines file contains the actual fonts outlines, as well as the skeletons, scaffolds and the bounding boxes.

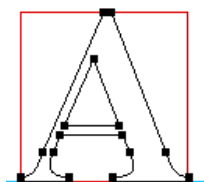
The Intmetrics file contains the characters widths, another set of bounding boxes, optionally kerning data and the Misc data for the font. Intmetrics are stored as a separate file so the computer can quickly look at the size of a font and the space it will take up on screen without having to load the much more complex Outlines file.

Font Components

We have already explained that a font contains sets of drawings (called Outlines) as well as a number of other parts, but what are they and what do they do.

Outlines

These are the letter shapes that you actually see when a font is rendered. As explained earlier they are very similar to draw files. They are scaled up and down to the right size to plot the font correctly at a given point size. As you can see from the illustration on the left the outlines really consist of a set of points linked together by straight lines and curves.

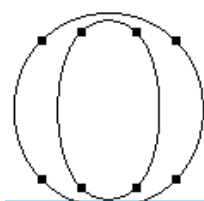


An outline of a character with its control points

FontFiend allows outlines to be constructed using the Draw application supplied with all Acorn computers. These can then be imported into the right character slot in the font.

Control Points

Acorn fonts have a slightly different format for the control points than PostScript fonts. Acorn format control points are located at 45 degrees, whereas PostScript requires the control points to be at 90 degrees. The two illustrations on the left show the difference. PostScript has its control points at the horizontal and vertical extremities, but the Acorn font has them at 45 degrees.

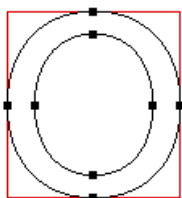


An outline of a character with Acorn control points

FontFiend can operate with either type of control point. If however you ever intend using the font on any machine other than an Acorn you should use the PostScript control points in preference.

Bezier curves

The main type of join between points in a character is called a bezier curve. These are identical to the bezier curves used in Draw. They have a start point, an end point and two control points that can be moved around in order to change the shape of the curve.



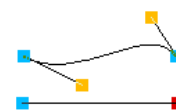
An outline of a character with PostScript control points

Straight lines

Many characters in a font eg. H may be made entirely of straight lines, these are not the same as bezier curves. They have a start point and an end point, but they have no control points.

Moves

Outlines also contain move commands. These operate in the same way as the move tool in Draw. The current path (a collection of lines) is ended and a new path starts from another point. This allows you to make a single character that has more than one path. For example an **O** consists of two paths, one for the inside of the letter and one for the outside.

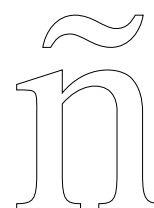


A bezier curve and a straight line compared

FontFiend supports a number of automatic operations that can be applied to outlines. FontFiend can even draw some of the outlines for your font automatically.

Design Size

The x,y position of control points in Acorn fonts are stored as 12 bit numbers in the range -2048 to +2048. The design size of the font is the number of these units that occupy 1 em. A font with a design size of 1000 has 1000 units (possible positions) per em. This means that the maximum size of a character can be 4096 ($2048 * 2$)/1000 ems, or 4.096 ems. Having a design size greater than 1000 can limit the size of characters. A design size of less than 1000 can have detrimental effects on the quality of a font since there may not be enough possible positions to draw outlines accurately.



A composite character

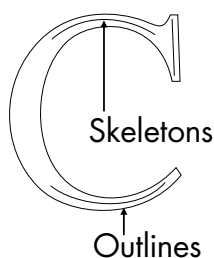
Composites

Composite characters are characters that are made up of copies of the outlines from other characters. For example an accented letter such as ñ (ntilde) will be made up of two composite characters (n & tilde). Any change made to the outline of a character will effect any other characters that use the character as a composite.

If for example the shape of the letter n was changed, then the shape of ñ would also change, since this character is just a copy of two other characters superimposed on top of each other.

FontFiend allows you extensive control over composites, including converting them to real letters. If we converted ñ to a real letter and then changed character n then ñ would not change since it would no longer be a composite. Composites have a number of advantages, firstly using composites makes a font much smaller, secondly any changes made to the real letter will automatically affect any composites that use it.

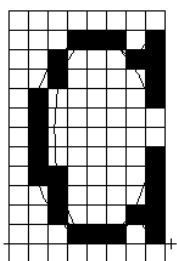
Skeletons



A symbol from a font with both Outlines and Skeletons

Skeletons are drawn inside or around the character shapes (Outlines) to prevent any part of the character “dropping-out” at small sizes. They are usually placed inside parts of the outlines that are quite thin. In the example on the left you can see that the skeletons do not extend all the way around the letter C. Skeletons are generally needed inside curved parts of letters, or parts of the letter that are not completely vertical or horizontal. It is quite unusual to place skeletons inside an upper case H for example, as it just contains vertical and horizontal strokes. However if the H is very fine, or oblique, skeletons may be needed.

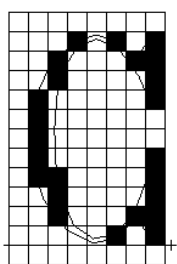
It is not entirely clear why skeletons are required. If the font manager ensured that it rendered one side of a stroke then skeletons would not be needed. PostScript fonts, for example, do not need skeletons to prevent drop outs. However as you can see from the two further illustrations on the left Acorn fonts definitely do need skeletons if they are to look good at very small sizes.



A bitmap of a character with skeletons.
(12pt at 90dpi)

In short if you are designing a quality font you will need to include skeletons, but only in the very thin parts of your characters.

FontFiend incorporates a number of automatic skeleton options, see the appropriate section in the manual for more details.



A bitmap of a character without skeletons.
(12pt at 90dpi)

If you have used FontED you may well be under the impression the outlines are filled shapes and skeletons are single lines inside the character. However, although FontED itself does not support it, skeletons can actually be complete paths and do not have to be inside the strokes of the character.

For example you could make a font that has no outlines, but has skeletons that make up the character shapes. This would result in a font that is rendered as a 1 pixel thick “open” font at any point size.

Widths

The widths specify how far the cursor, or caret, should move after a letter has been printed. If the width of a letter is not great enough then the next character printed will overlap. If the width is too great the next character will be too far away.

Setting up the correct widths for a font manually can be very time consuming. FontFiend allows you to set widths for characters up automatically based upon the size of the letter. FontFiend will also automatically set the widths for accented letters and any other letters or symbols it makes.

FontFiend will set widths using both mathematical models and an understanding of the font you are working on. Traditionally widths of a font are set manually by visually comparing letter gaps. FontFiend is very accurate when setting character widths, however it may be worth checking the results on complex script fonts.

Note: Some characters eg. accents, should have no width as they are only ever used in conjunction with another character. Please also note that the Risc OS 2 font manager can crash on some zero width characters in an unpredictable manner.

Widths are normally positive numbers specified in thousandths of an em. In theory widths can be negative as well as positive. However negative width characters can have an undesirable effect on many applications and we suggest that you stick with positive widths at all costs.

The image shows the characters 'A' and 'B' in a serif font. The 'A' is significantly narrower than the 'B', causing the 'B' to appear cramped and slightly overlapping the 'A'.

Character A with too small a width

The image shows the characters 'A' and 'b' in a serif font. The 'A' is much wider than the 'b', creating a large gap between them that makes the pair look unbalanced.

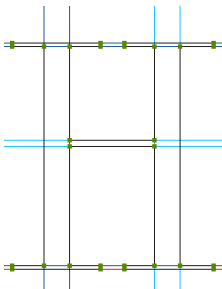
Character A with too large a width

The image shows the characters 'A' and 'b' in a serif font. The 'A' and 'b' are well-proportioned and fit together naturally, representing the correct width setting.

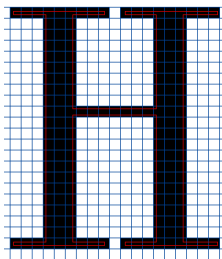
Character A with the correct width

Scaffolds

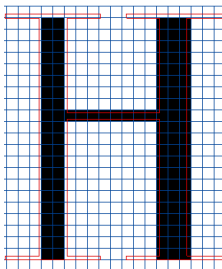
Scaffolds and skeletons are often referred to as hinting. The character's scaffolds lock parts of the character together as well as locking the same parts of different characters together. On the left is the outline of character H with scaffolds applied, you can see how they match the shape of the letter. Below are two bitmaps, one with scaffolding applied, and one without. The difference is fairly obvious.



Character H with scaffolds applied



Bitmap of character H with scaffolds applied



Bitmap of character H with no scaffolds applied

Scaffolds adjust the outlines of the character so that they always sit neatly on a pixel grid, if part of the letter goes below one pixel it will be adjusted so that it is exactly one pixel. Also similar parts of the letter are adjusted so that they are the same size.

Control points in both outlines and skeletons are attached to scaffolds.

The two main types of scaffold are "tangents" and "wide links", either of these can be vertical or horizontal and can either relate to one character (local) or to several characters (global). Tangents can also be either left or right, or top and bottom, this gives us a total of 12 different scaffold types. Scaffolds are used by the font manager to determine which parts of a character should be rendered and adjusted to the pixel grid. Control points attached to scaffold 1 will be adjusted first and then those attached to scaffold 2 and so on.

Wide links

These are the same type of scaffolds as we can see on the character H opposite. As you can see by placing two wide links, the same width apart, over the two stems of the H, the stems stay the same width when the font is rendered. Wide links are also used on the serifs to ensure that they appear. Wide links can either be horizontal or vertical, there are however a couple of restrictions on their use. They can only be a maximum of 253 design units in width, this can make scaffolding some fonts very tricky. Also they will only work correctly on rectangular objects, such as the stems on the H shown at the top of the page.

Tangent links

These are designed to prevent unsightly "pips" at the edges of round characters. They are usually used on characters like O or C. The example show character O with tangent links at all the edges. The two bit maps show the results of tangent links and not having tangent links.

Global scaffolds

Scaffolds can relate to a single character or to a range of characters. For example you could have a global scaffold across the top of all upper case letters to ensure that they have the same cap-height when rendered at small sizes. This scaffold would be defined in one upper case character and a copy if it appear in all the others. FontFiend automatically builds a scaffold tree when hinting a font.

Scaffold trees

A scaffold tree defines the way that global scaffolds are linked together inside a font. A scaffold may be defined in one character only, but appear in many others. If you delete the original scaffold then the tree will be corrupted and the font will no longer work. Global scaffolds propagate through the font in specific ways. If you look at a plan of this it looks as though the letters have grown like leaves on a tree.

Linear linking

This is a specialised form of linking. In the character E, for example, the centre scaffold would be linked between the top and bottom scaffold. This means that the position of the centre of the E will always remain the same in relation to the top and bottom of the character at any size.

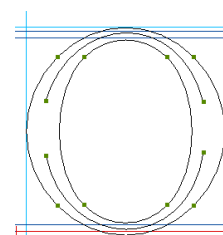
X & Y Scaffolds

Scaffolds can be placed either horizontally or vertically. Those that are placed across the letter are the X scaffolds (since they move in the X axis). This may initially be confusing but with time will make sense. For example the scaffold across the centre stroke of an H is a Y scaffold, since it can move up or down (in the Y axis). The scaffolds that attach to the two vertical strokes of the H are X scaffolds since they can move left or right (in the X axis).

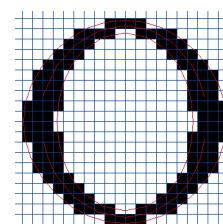
Connections

Scaffolds are linked to the control points that make up the outlines of a character. A scaffold can only work if it is connected to the right control points. FontFiend will automatically attach control points to scaffolds.

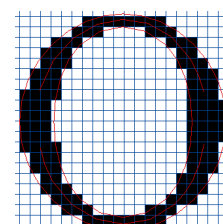
FontFiend supports automatic scaffolding of characters which means that in the vast majority of cases no manual editing will be required. FontFiend also links the scaffolds together itself so that no manual linking is needed.



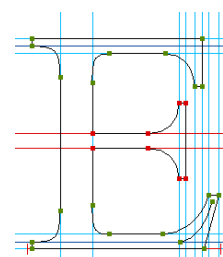
Character O with tangent links



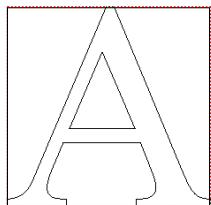
Bitmap of character O with tangent links



Bitmap of character O without tangent links



Character E with linked scaffolds



A character with its
bounding box

Bounding Boxes

A bounding box literally defines the furthest parts of a character, it is the smallest possible rectangular box that can enclose the entire character. Bounding boxes are very important, if a font has faulty bounding boxes it will become very un-reliable and cause "Font Cache Full" errors. Bounding boxes are stored in the file "Outlines" and optionally in the file called "Intmetrics".

FontFiend automatically recalculates bounding boxes after any change to a character or to a font so that the correct values are stored automatically. FontFiend also calculates the bounding boxes for any characters that are imported from Draw.

Note: If a font does have a corrupted bounding box then on loading it into FontFiend you should choose the "Bounding Recalculate" option and let FontFiend check all the bounding boxes and change them as necessary.

Global bounding boxes

A font also has a global bounding box, this is a box that can enclose any character in the font. You may sometimes find a font that works correctly for normal letters, but parts of the accents over accented letters do not get rendered correctly. This is because the font's global bounding box is not big enough to enclose accented letters. You can recalculate the global bounding box from the FontFiend global bounding box dialogue window.

Kerning

Kerning is stored in the “Intmetrics” part of a font. Kerning is the adjustment of gaps between letters. It allows the computer to alter the width of a letter (the amount the caret moves after printing the letter) depending on the next letter to be printed.



A kerned letter pair

The illustrations on the left show the effect of kerning. In the first kerned pair the characters overlap slightly, but in the un-kerned pair they do not. Kerning can make a great deal of difference to the appearance of text when typeset using a particular font.



An un-kerned letter pair

Kerning is defined as a pair of characters and an adjustment value. If this pair of characters is to be rendered the first character is drawn, then the position of the second character is moved by the adjustment value before that character is rendered.

Only Acorn Risc OS 3 (Version 8) fonts have kerning data, Risc OS 2 (Version 6) fonts do not. FontFiend loads and saves fonts with the kerning data intact. You can also edit any kern pairs present using the built in kerning editor.

Misc Data

This contains all sorts of interesting data about the font. Only Risc OS 3 fonts have a misc data area. If FontFiend finds a font without a misc data area then it will automatically construct the information in the area itself.

The misc data includes:

- x-height of the font
- Cap-height of the font
- Descender & Ascender heights for the font etc

All of this information can be examined in the FontFiend misc data window.

The most interesting piece of information stored in this area are the offsets. The default X offset value specifies a standard width that is applied to all characters in addition to the width they have themselves. For example if you set all the widths in a font to zero and then set the default X offset to 1000 the caret would advance 1 em after every character was rendered, ie the font would be mono-spaced.

The misc data area also includes a default Y offset value. A font can easily be made that does not advance across the page when printed, but moves down the page. However most applications will get horribly confused if this is attempted.

The italic offset is used by FontFiend when building letters so that it can correctly position accented letters on a font that has a slope.

More details on the misc data area can be found in the chapter FontFiend main menu.

Chapter 2

Getting Started



Setting up Memory

FontFiend requires quite a lot of free memory in order to run. Before trying to load FontFiend ensure that you have at least 4Mb of free memory. When loaded FontFiend will grab either 4Mb or the size of the *Next* slot in the Task Manager, whichever is the greater. We recommend that you drag the *Next* slot up to the largest amount possible, leaving about 300K spare for workspace.

To change to the size of the *Next* slot simply click *Select* on the Acorn icon on the Icon Bar. A window will open containing a lot of sliders which show memory usage. Drag the *Next* slot slider up to the amount you want and then close the Task Manager window. Do not drag the next slot all the way as then there will not be enough workspace memory, leave around 300K in the *Free* slot.

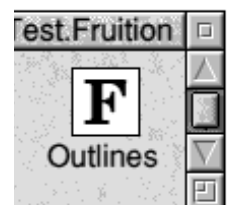
Loading FontFiend

FontFiend is loaded by simply double clicking on its icon. Provided you have enough memory the program will then load. Firstly you will see the FontFiend title screen and then the application will load. If you do get any messages about memory for example *"3800K free memory is needed before the application will start. Quit any unwanted applications or see the Risc OS user guide for ways to maximise memory."* then ensure you have enough memory free, for example quit any loaded tasks and try again.

If all is well when you next double click on FontFiend its icon will appear on the right hand side of the icon bar.

Loading a Font

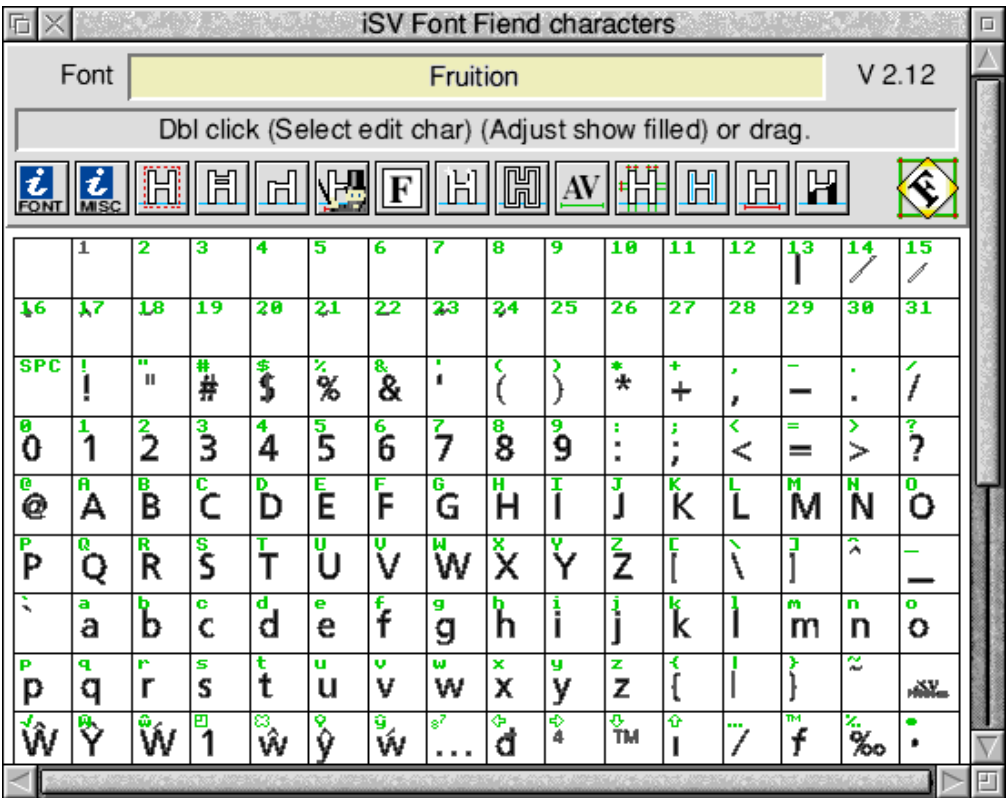
To load a font into FontFiend drag the fonts *Outlines* file to the FontFiend icon on the icon bar. You can also double click on an *Outlines* file in order to load it into FontFiend. Also see Appendix 4 **"Loading Base 0 fonts"**.



The hourglass will then appear and the font will load. You may see a percentage hourglass appear, this happens when FontFiend is producing the small thumbnails for the main window. Once the font has been loaded the main window will appear

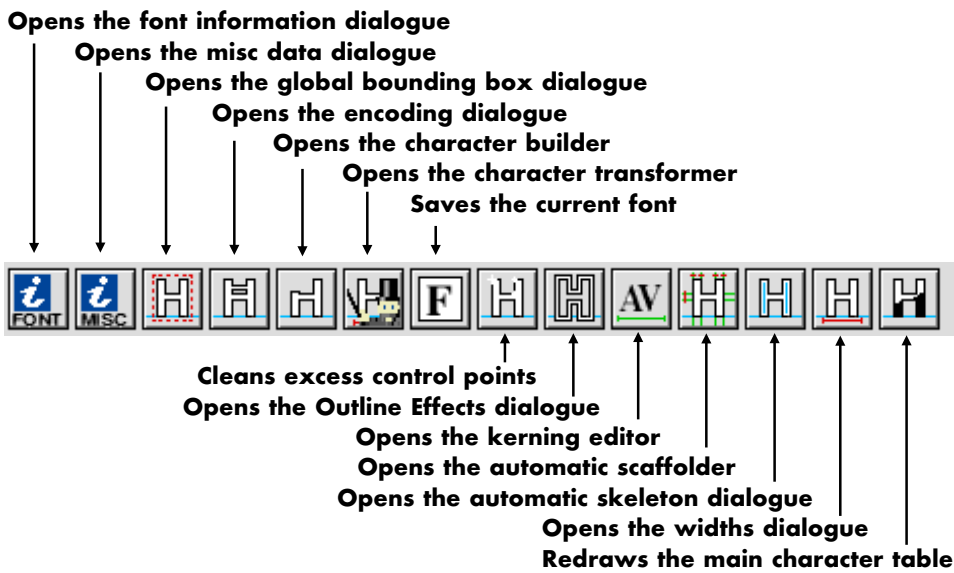


The FontFiend main window showing the font Fruition. The title bar is at the top with the thumbnails for each letter below.



The FontFiend main window consists of two sections, the button bar and the main character thumbnails table. The button bar shows the fonts name and displays helpful messages from FontFiend as you move the mouse around. Below these are the buttons themselves.

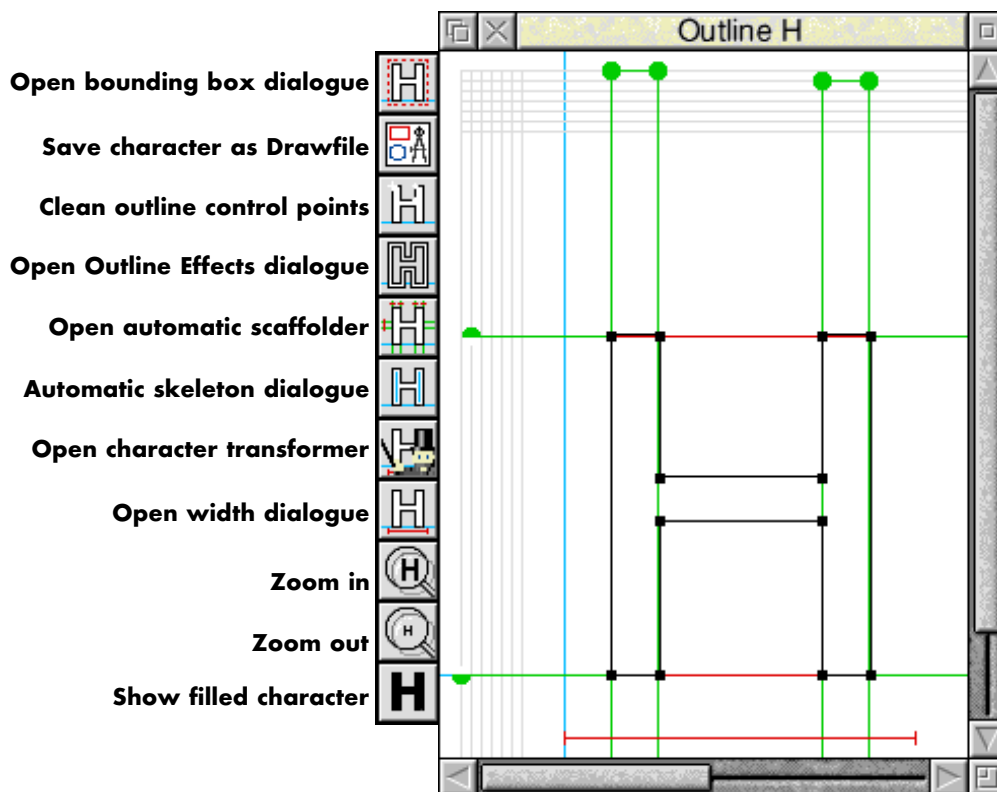
The FontFiend button bar.



FontFiend will provide on-line help as you move the mouse over the buttons. Also note that FontFiend supports interactive help as well.

Looking at characters

FontFiend allows you to view the characters in a font in two different ways. If you move the mouse pointer over the top of one of the character thumbnails in the main window and double click with SELECT then a window will open showing the outline version of the particular character along with another button bar.



An Outline window showing the outlines of the character H. This window also shows the skeletons & scaffolds.

This "Outline window" displays not only the outlines that make up the character but also shows the following:

- Scaffolds applied to the character (green lines)
- Scaffold base lines (grey lines along left/top of the window)
- Skeletons applied to the character (cyan lines/control points)
- The characters width (red bar under the character)
- Any composite characters (grey outlines with no control points)

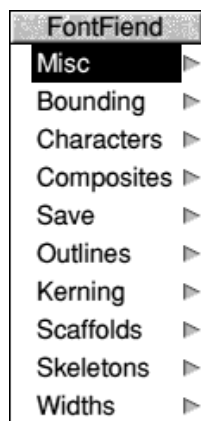
Filled character windows

You can also see what a character will look like when it is rendered by double clicking ADJUST on one of the character thumbnails in the main window. This "Filled character" window can be scaled to alter the size of the character. Users of FontED should be able to see that FontFiend has been designed to operate in a similar manner.



The FontFiend main menu

FontFiend has two main sets of menu. One menu applies to the entire font, this is opened by pressing MENU over FontFiends main window.



The FontFiend main menu.

As you can see from the illustration on the left this menu has various options. For example going past the Bounding option will lead to a submenu that is concerned with bounding boxes for the entire font.

The *Characters* option leads to a sub-menu that is concerned with entire characters (including Outlines, Skeletons, Scaffolds and Composites).

All of the other options should be obvious. If any are not then see the first chapter in this manual "What is a Font?". The FontFiend main menu is covered in chapter 5.

Character menus

The second type of menu is concerned with individual characters. To open the menu for a character you need to first open the Outline Window for that character. You do this by double clicking with SELECT on one of the character thumbnails on the main window.

The Outline Window for this character will then open. If you then press MENU in this window the menu for this character will open.



The FontFiend character menu.

This menu is arranged in a similar way to the main menu. Each of the options (except Transform, Full char & Restore) leads to a sub menu.

The Transform option opens the Character Transformation window for this character. This is covered later in the manual

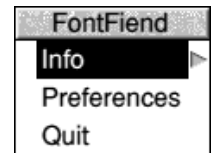
The Full char option will open the filled character window for this character.

Filled character windows (2)

The filled character windows do not have a menu since they are only designed to show you what the particular character will look like when rendered. However if you click any mouse button inside one of these windows the Outline window for the same character will open on top of the filled window. This can be useful as it means you do not have to close the filled window and then find the correct thumbnail to click on in the FontFiend main table in order to open the Outline window.

The Icon bar menu

FontFiend also has an icon bar menu. This is opened by going to the FontFiend icon on the icon bar and clicking the menu button. The Info option leads to an information dialogue box that tells you about the copy of FontFiend that you are running.



The quit option will quit FontFiend. If you are editing a font at the time a confirmation window will open first. This prevents you quitting accidentally.

The preferences option will open the FontFiend preferences window. This controls a number of settings regarding how FontFiend will work.

The preferences window has its own chapter later in the manual.

Getting started summary

These 5 pages cover where things are located in FontFiend. To sum up:

1. Options that alter the whole font are found by clicking MENU on the main table.
2. You can view a single character by double clicking Select on its thumbnail in the main table. This opens an Outline Window for the particular character.
3. Options that alter a single character are found by clicking MENU over the Outline Window for a single character.
4. Commonly used parts of FontFiend can be quickly found using the button bars on top of the main window and attached to the side of the outline windows.

We recommend that you have a play with FontFiend at this point. Remember that whatever you do to a loaded font will not matter provided that you do not save the font back out of FontFiend.

We also recommend that initially you use the menus rather than the button bars, as many less frequently used options are only found on the menus. Once you have become familiar with the menu structures then try using the buttons bars to speed up operations.

3QD Font Fiend



Those who are experienced with Acorn outline fonts will quickly be able to pick up how FontFiend works. From the outset FontFiend was designed to be quick and easy to use. For this reason many of the conventions established by Acorns own FontED application have been followed. If you can use FontED then you are already halfway towards being a FontFiend yourself!

Chapter 3

FontFiend Tutorial

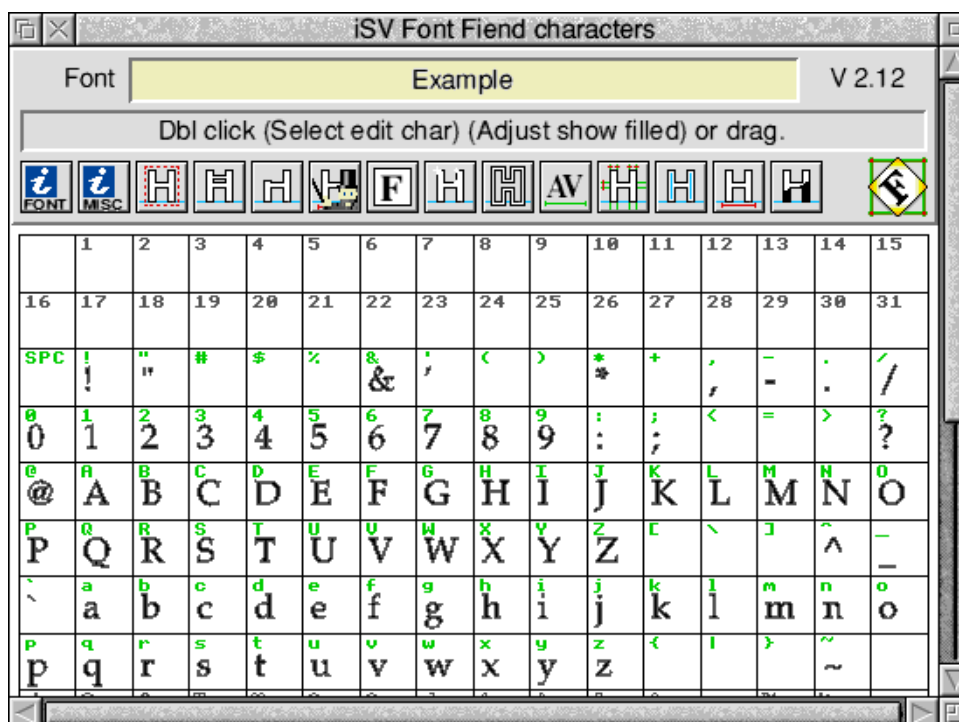


At this point it would be worthwhile running through an example of using FontFiend to complete a half finished font.

Supplied with FontFiend is a directory called Examples. This contains various files including a directory called QuickStart. Inside this directory is a font called Example. You can use FontFiend to finish this font in a matter of minutes.

Loading the Example font

Load the Example font into FontFiend by dragging the fonts Outlines file to the FontFiend icon on the icon bar. The hourglass will appear and the font will then load.



The example font loaded into FontFiend.

As you can see the font has upper and lower case letters, and some numbers/symbols. Also if you have a look at any of the Outline Windows for this font you will see that the font does not have scaffolds or skeletons.

The font does not have any accents or characters above ASCII 127. Also a number of symbols are missing from the third row down, and all brackets and bars are missing.

Adding the missing letters

3QD Font Fiend



FontFiend can add all of the missing letters in this font. FontFiend can do this by combining parts of existing letters to make new ones, or by drawing new letter shapes itself from scratch.

Most of the time FontFiend's letters will be exactly right. Occasionally some further work may be required, for example if the font does not have proper lower case letters FontFiend will not be able to make some symbols such as thorn (þ) correctly.

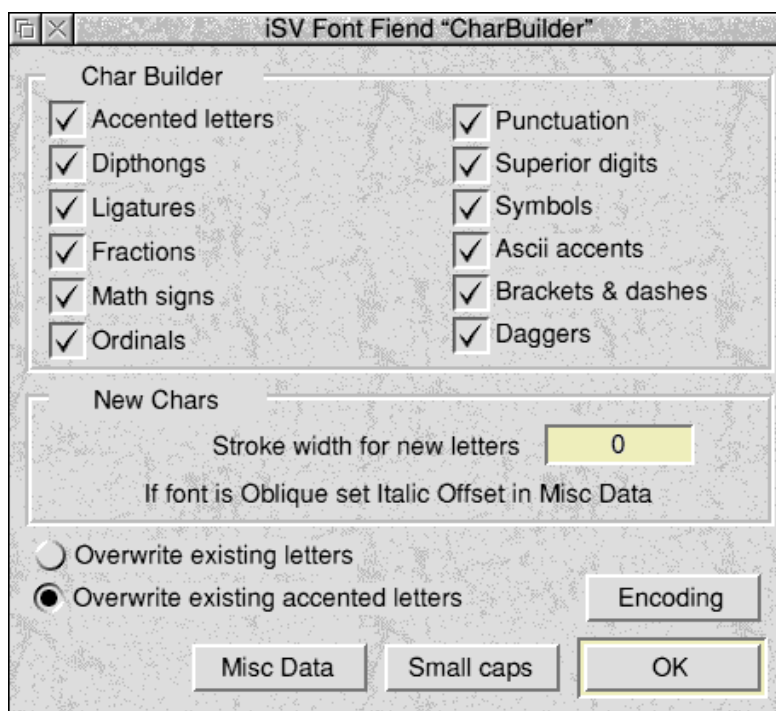
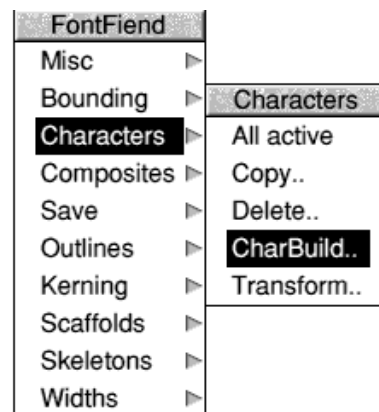
The menu required to open the Character Builder window. This can also be accessed using the following button.



The Character Builder window.

To add the missing letters simply press **MENU** over the main table and then go to the *CharBuild* option. Click on CharBuild and a window will open.

This window (below) is called the Character Builder. It is designed to make adding missing letters to a font very easy. Leave all the buttons set as they are and



simply click on the **OK** button in the bottom right corner. FontFiend will now start checking the font and adding any missing letters. Almost immediately a confirmation window will open asking if you want to load an accent file. FontFiend has checked the font and found missing accents. Click on *Continue*.



Font: Example V.2.12

Dbl click (Select edit char) (Adjust show filled) or drag.

















	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15
16	17	18	19	20	21	22	23	24	25	26	27	28	29	30	31
SPC	!	"	#	\$	%	&	'	(	)	*	+	,	-	.	/
0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
p	q	r	s	t	u	v	w	x	y	z	{		}	~	■
Ŵ	Ŷ	Ÿ	1	ŵ	ŷ	Ź	...	đ	4	™	1	/	f	%/∞	•
`	'	<	>	"	"	"	-	-	-	œ	œ	†	‡	fi	fl
NBS	i	¢	£	¤	¥	¦	§	¨	©	ª	«	¬	®	¯	
°	±	²	³	´	µ	¶	·	¸	¹	º	»	¼	½	¾	¿
À	Á	Â	Ã	Ä	Å	Æ	Ç	È	É	Ê	Ë	Ì	Í	Î	Ï
Ð	Ñ	Ò	Ó	Ô	Õ	Ö	×	Ø	Ù	Ú	Û	Ü	Ý	Þ	ß
à	á	â	ã	ä	å	æ	ç	è	é	ê	ë	ì	í	î	ï
ø	ñ	ò	ó	ô	õ	÷	ø	ù	ú	û	ü	ý	þ	ÿ	

Adding Skeletons

Press the **MENU** button over the main table and go to the *Skeletons* option. Then click on the *Automatic* option in the submenu. The automatic skeleton



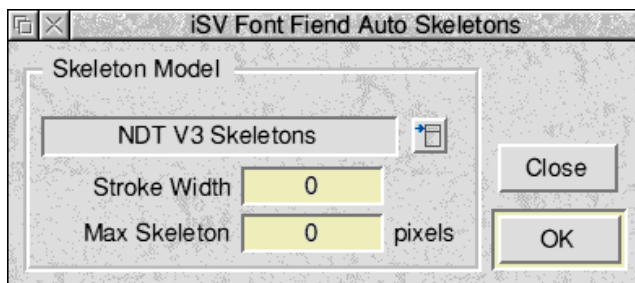
The Automatic Skeletons window. This can also be opened with this button.



The Outline Window for character H.

window will now open.

As you can see there is a menu button in this window. Click on this to open a list of the skeleton models supported. Choose "NDT V3 Skeletons" from



the list.

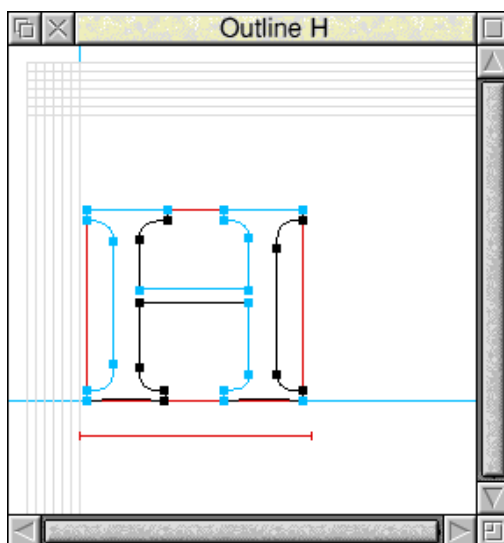
Leave the Stroke Width and Max Skeleton values at 0 and click on the OK button. FontFiend will now add skeletons to the font. The "NDT V3 Skeletons" option places skeletons along one side of each stroke in the character.

All that remains now is to add scaffolds to the font.

Adding Scaffolds

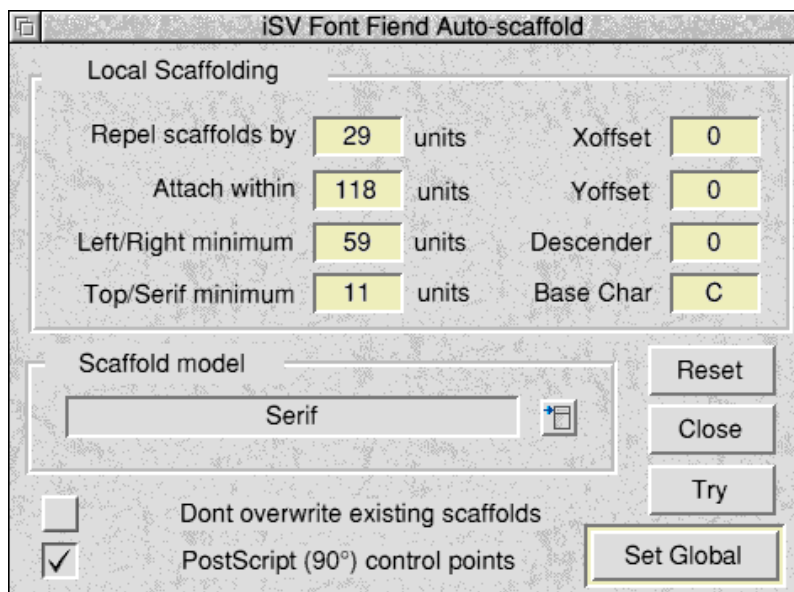
Adding scaffolds is a little bit more complex, however FontFiend can still save many hours of work. Before scaffolding an entire font we first need to check the scaffolding on a couple of letters.

Double click on the H character in the main table with SELECT. The Outline Window for H will open. At this point you will be able to see the outlines for the letter (in black) and the skeletons (in light blue).



Press MENU over this window and go to the *Scaffolds* entry in the menu. Go past this and click on *Automatic* to open the Automatic Scaffold window.

As you can see the Auto-Scaffolder looks quite complicated, but with a bit of practice it is very easy to use. Under the scaffold model section click on the menu button to



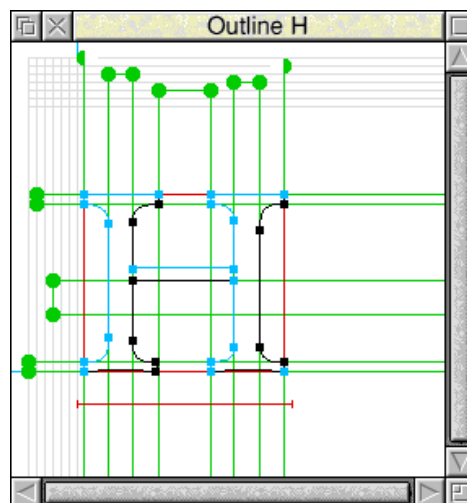
The single character Auto-Scaffolder. This can also be opened using this button. on the outline window toolbar.



open the list of supported scaffold models. Choose *Serif* from the list and click on it. (We are going to use *Serif* since our example font has serifs).

Now click on the *Try* button. Scaffolds will then appear in the outline window for character H. As you can see FontFiend has got most of the scaffolds correct, however the cross bar of the H is not scaffolded correctly. This is because the Y Offset in the scaffold window was not set correctly. We need to alter this value for the scaffolding to work.

Click in the writable icon next to *Yoffset* in the auto-scaffolder window and then change the value to 25. This will alter where FontFiend considers the centre of the characters height will be. Now click on *Try* again.

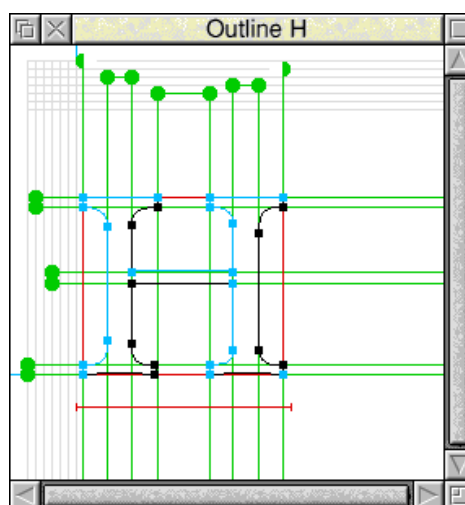


FontFiend will now apply scaffolds to the font again. This time the cross bar of H will be correctly scaffolded. You can now try these settings on other individual characters in the font. To do this keep the Auto-Scaffold window on screen and simply close the outline window for character H. Now go back to the main table and double click on another character thumbnail to open its outline window.



Character H correctly scaffolded.

The FontFiend automatic serif scaffolder can scaffold a font in under 15 seconds with over 95% accuracy. Some manual tweaking of the resulting scaffolds may be required. However this is still much quicker than trying to scaffold a font by hand with FontED, which can take many days!



Scaffolding the entire font

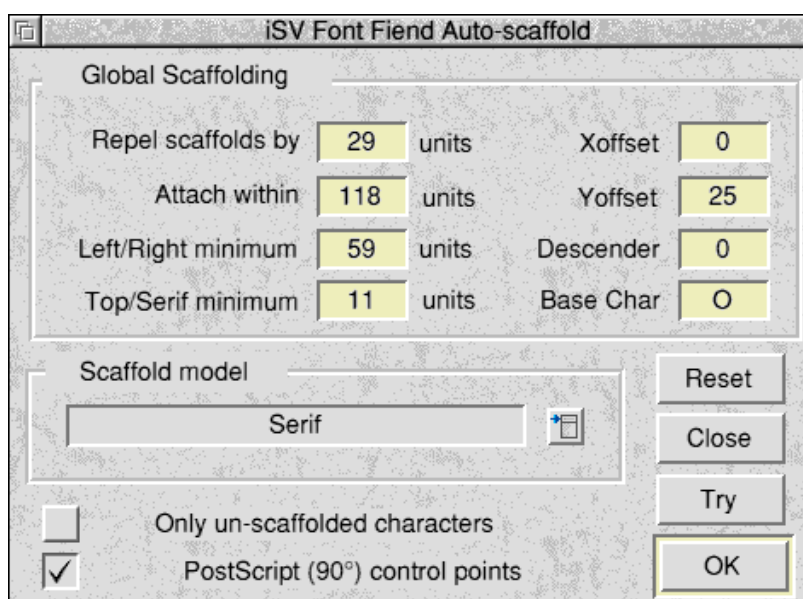
You are now ready to scaffold the entire font, firstly the values from the local character auto-scaffolder need to be copied to the global automatic scaffolder. Click on the *Set Global* button to copy these values over.

Now click MENU over the main table, go to the *Scaffolds* option and click on the *Automatic* entry from the submenu. This will open the global automatic scaffolder. This is very similar to the local version except it does not have a Set Global button. Click on the OK button and the hourglass will appear while the font is scaffolded.

Saving the finished font

The font is now completed, it has a full character set, scaffolds and skeletons. You can now save this font to disk. Click MENU over the main table, go to the *Save* option, go to the sub menu and click on *Font*.

The global Auto-Scaffolder. This can be opened from this button on the main character table.



The hourglass will appear for a couple of seconds while FontFiend converts the font from its own format to Acorn format. A save dialogue box will then appear.

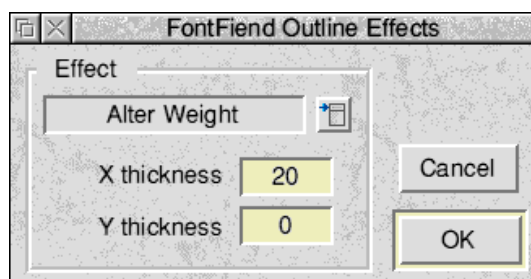


Drag the file (called Outlines) from this box to a filer window. Two files will be saved, one called Outlines and one called Intmetrics. The finished font is now saved.

Making a bolder version

You can also make a bolder version of this font quickly and easily. Press MENU over the main table. Go past the *Outlines* option, then past *FX/Process* and click on *Special FX*. A window will open.

Leave the settings as they are and click on the *OK* button. FontFiend will now increase the weight of the font in the X direction. The hourglass will appear for a few seconds while this is happening. Once FontFiend has finished this you can change the name of the font to Example.Bold and save it.



The global Outline Effects dialogue box. This can be opened using this button on the main character toolbar.



That's it for the FontFiend tutorial. The rest of the manual will cover FontFiend in greater detail. However remember that Fontfiend has an on-line help system which is available all the time and provides a lot of useful information. In fact the on-line help system contains almost 400 messages. By moving the mouse pointer over FontFiend windows you can discover a great deal about how to use FontFiend.



FontFiend main
character table.



Dragging with Select

Holding down SHIFT while dragging will move the character from one place to another.



Dragging with Adjust

Dragging a character with ADJUST will copy its scaffolds to whichever character you drop it on.

Holding down SHIFT while dragging with ADJUST will insert the character you are dragging as a composite into whichever character you drop it on.

Holding down CTRL while dragging with ADJUST will remove a composite entry for this character from the one you drop it on. This of course assumes that the character you drop it on actually has the character you are dragging as a composite.

Exporting characters as drawfiles

You can also drag a character out of the main table and into a filer window or a window owned by another application. In this case FontFiend will attempt to export a drawfile of the particular characters outlines and skeletons.

Importing characters as drawfiles

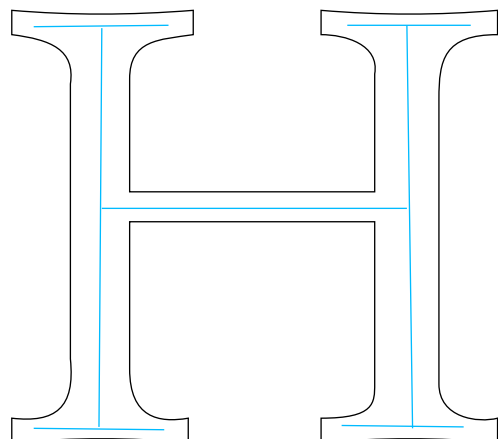
You can also import drawfiles into the main table. Drag in a drawfile and drop it on the character you want. FontFiend will then attempt to convert the drawfile to an outline and skeleton font definition.

Obviously any drawfile you drag into FontFiend should only include path objects, not sprites or lines of text that are not converted to a path.

When importing a drawfile FontFiend treats any path objects that have a line colour of black as outlines. Any path objects that do not have a line colour of black are treated as skeletons.

In this example the drawfile has a full set of outlines (in black) and some skeletons in grey (the lines inside the letter).

This character was exported from FontFiend as a drawfile.



Importing other files

You can also import other types of files into FontFiend by dropping them on the main window. These include the following:



Accents	as saved by FontFiend
Composites	as saved by FontFiend
Encoding files	as used by the font manager
Kerning	as saved by FontFiend or FontKern
Scaffolds	as saved by FontFiend
Skeletons	as saved by FontFiend
AFM	Adobe Font Metrics files
Transformations	as saved by FontFiend or FontTrixPro

For example you can load a font, then save out the skeletons. You can then discard this font and load a new font and then drag in the skeleton file you saved earlier.

Whenever FontFiend imports a file it is always altered to fit the current font. For example you could save the skeletons out from an extended weight of a font and then import them into a condensed font. FontFiend will automatically re-size and re-scale these skeletons to try and fit them within the bounding boxes for the font you are working on.

Importing AFM files

You can also import AFM files from PostScript fonts by dropping them into the main window.

AFM files contain a lot of information including the bounding boxes, composites and widths of a PostScript font. So you could, for example, design your own version of a popular font. Then drag in the PostScript fonts AFM file. FontFiend will then load this AFM and after confirming the action do the following:

Load the widths information from the AFM file and replace all existing widths with these new values.

Load the bounding boxes from the AFM file and then fit all the current characters to these new bounding boxes.

Optionally load the kerning from the AFM file and use this in place of any existing kerning. Also you can choose to load any composite character definitions from the AFM file.



Importing Encoding files

FontFiend can also import custom encoding files. Encodings are very important to FontFiend, which is why they will crop up time and time again.

Examine the in-built FontFiend encoding files (stored in the Encodings directory inside FontFiend) for an idea how they are written. Encoding files contain a header line (started with a % character) and then contain 256 character names in a list. The character names before space (ASCII 32) must be the same as those in the default encodings for FontFiend to be able to use the encoding properly.

For more information on encodings see the glossary at the back of this manual and the Appendix concerned with encodings.

FontFiend will always confirm that you are sure that you want to load a new file before actually loading it.

Transformations

Transformation files contain a transformation matrix that can be used to alter the appearance of a font. They can extend it, condense it or make it oblique. Transformation files are loaded into the Transform window (which is covered in Chapter 5).

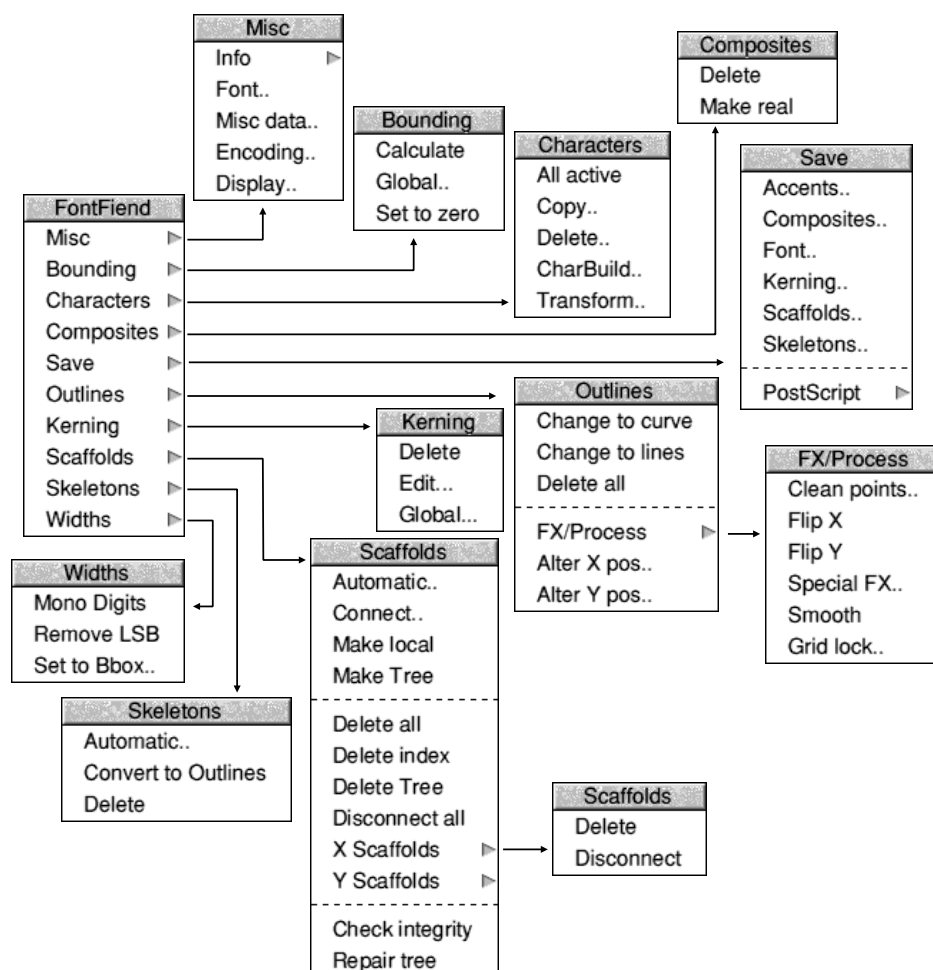
There are some example transformation files in the examples directory supplied with FontFiend.

Chapter 5

FontFiend Main Menu



The FontFiend main menu is opened by clicking MENU over the main character table.



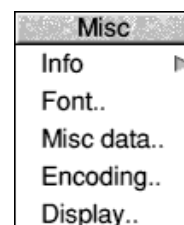
The structure of the main menu. From FontFiend8 version 2.12. Later versions may have slight differences.

The illustration above shows the main menu tree. Each of these sections will be covered in the next few pages.

The Misc submenu

This sub menu leads to a number of dialogue boxes. The top option "Info" opens a copy of FontFiends info dialogue box. This is the same as the one opened from the icon bar menu. The info dialogue box is opened by moving past the arrow.

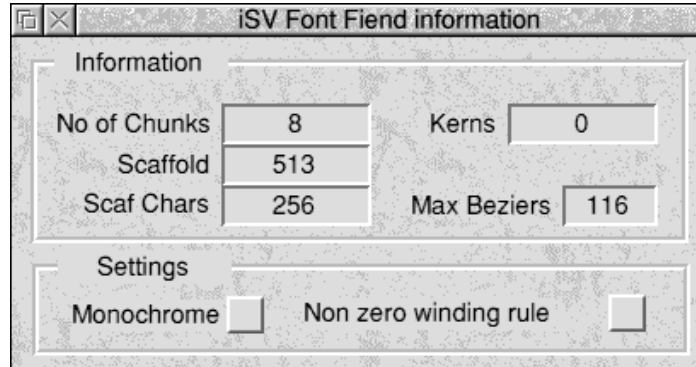
All of the other options work by clicking on them, rather than moving past them. This is indicated in FontFiend by having some ... characters after the menu entry. For example Misc data...





Misc Font information

This dialogue box provides miscellaneous information about the font you are editing.



The dialog box titled "iSV Font Fiend information" contains two sections: "Information" and "Settings".

Information	
No of Chunks	8
Scaffold	513
Scaf Chars	256
Kerns	0
Max Beziers	116

Settings	
Monochrome	☐
Non zero winding rule	☐

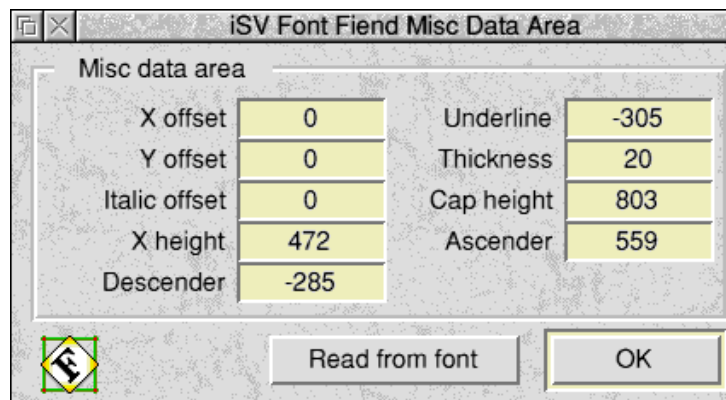
This includes the number of chunks in the font (each chunk contains 32 characters), the scaffold area size and the number of characters that have an index in the scaffold area. Note that just because a character has an index into the scaffold area does not mean that it actually has any scaffolds!

This window also displays the number of kern pairs and the maximum number of bezier curves that can be used in each characters outlines.

There are also two option buttons. Setting Monochrome on will disable the font managers anti-aliasing when the font is rendered. The Non zero winding rule button alters the fonts winding rule. If this button is on then the font will be rendered using a non zero winding rule system. With this button off the font will be rendered using an even-odd winding rule. Setting this button on can cause some letters to be rendered wrongly.

Misc data

Risc OS 3 & PostScript fonts have a misc data area, this contains



The dialog box titled "iSV Font Fiend Misc Data Area" contains a "Misc data area" section with two columns of data.

Misc data area	
X offset	0
Y offset	0
Italic offset	0
X height	472
Descender	-285
Underline	-305
Thickness	20
Cap height	803
Ascender	559

At the bottom left is a small icon of a yellow diamond with a black 'F'. At the bottom right are two buttons: "Read from font" and "OK".

information about the font that does not have any other place to live. This includes the underline position and the underline thickness. If the font you are loading into FontFiend has a misc data area then its contents will be displayed here. If the font does not have a misc data area then FontFiend will automatically fill in this window by reading the information directly from the font.



Some PD fonts may have a misc data area, but with all the settings set to zero. If this is the case then just click on the Read from font button to get FontFiend to replace these zero values.

Misc data offsets

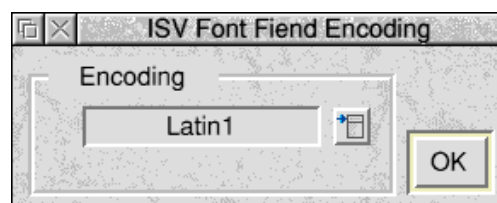
The most useful sections of the misc data area are the offsets. The X offset is added to the width of all characters when they are rendered by the font manager. So, for example, you can make a monospaced font by giving all characters a width of zero and then setting the default X offset to a reasonable value.

The Y offset works in a similar way, however it alters the amount the next character is rendered up or down from the last character. Unless you want to seriously confuse the font manager always leave this at zero.

Italic offset is the most useful offset. If FontFiend is building letters then it needs to know if the font is Italic or straight. If your font is Italic or Oblique then you must set this value. If you transform an existing font to make it Oblique using FontFiend then this value will be automatically set for you. A good starting value for an Oblique/Italic font is 0.15. See the section on the Transform window in Chapter 5 for more details.

Encoding

The encoding window allows you to select an encoding to use when building letters. The built in encodings are for Latin 1,2,3 and 4. You can also drag a custom encoding file into here to load that instead. For normal use this can be left set to latin 1.



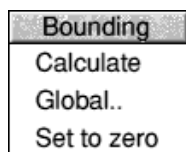


Display (Rendering preferences)

This window allows you to alter how FontFiend displays information in the outline windows. You can also access this window from the preferences window which is found by clicking MENU on the FontFiend icon on the icon bar. For more information see the chapter about Preferences.

The Bounding sub menu

This sub menu is concerned with the bounding boxes for the entire font.



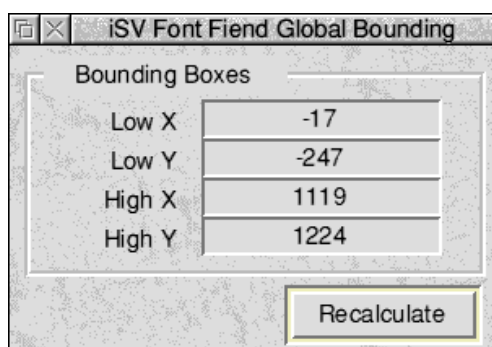
Bounding calculate

The first option "*Calculate*" will simply make FontFiend check every single character in the font and re-calculate its bounding box. You will not normally need to do this very often as FontFiend automatically calculates bounding boxes as it is working.

Note that for bounding boxes to be re-calculated the "*Re-calculate bounding boxes*" button in preferences must be on.

Bounding Global

Clicking on this option will open the global bounding box display window. This shows the global bounding box for the font. This is the smallest possible box that can enclose any letter in the font (including accented



letters). You can recalculate the global bounding box by clicking on the *Recalculate* button.

Set to zero

This final option simply clears all bounding boxes so that they are blank. A font will not work in this

state, its primary purpose is to ensure that FontFiend does recalculate all boxes correctly. Unless you are sure what you are doing we do not recommend that you use this option. If you do accidentally click on it then simply go and click on the "*Calculate*" option to restore all the bounding boxes to sensible values.

The characters sub menu

This menu is concerned with entire characters, including outlines, skeletons,

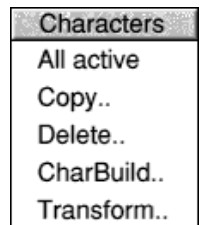
composites and scaffolds.



All active

Clicking on this option will turn on all the characters in a font. This does not mean that the character will be defined, just that the Font will then contain offsets to all characters.

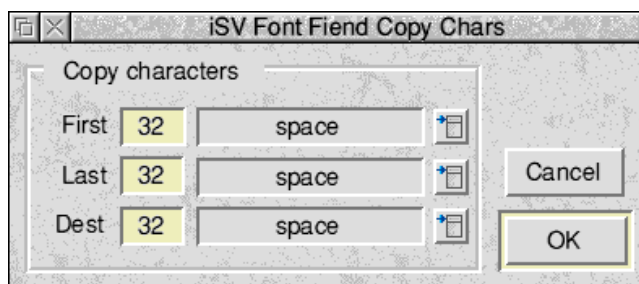
Some PD fonts can cause problems because not all of the characters are actually defined in the main table inside the font. For example dragging letters around the font in FontED can cause FontED to crash. If this happens you can load the font into FontFiend, click on the "All active" option then save the font back out. The font will then be editable by FontED.



Copy

Clicking on this option will open up the FontFiend copy characters dialogue box.

You can use this dialogue to copy blocks of letters inside the font. For example to copy upper case A to Z on top of lower case a to z you would enter the following:



First character to copy
65 (ASCII value for
A)
Last character to copy
90 (ASCII value for
Z)
Destination start

character 97 (ASCII value for a)

You can enter these values by either typing in the number of the character or by choosing the character by name from a menu.

Delete

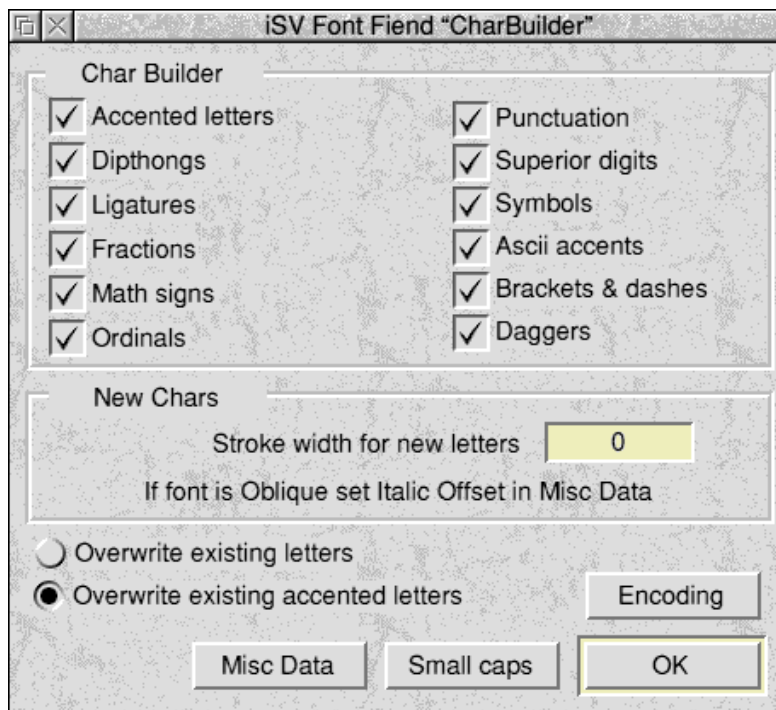
Clicking on this will open the delete characters window. This is similar to the copy window but allows you to delete characters from a start value to an end value. To delete all upper case letters you would enter the *From* value as 65 and the *To* value as 90 (see copy above).

Char build

Clicking on this option will open the FontFiend Character Builder. This is the part of FontFiend that can fill in missing letters in a font.



The first part of this window control what sorts of characters FontFiend will make. If a button is on for this type of character then FontFiend will make it. If it is off then FontFiend won't.

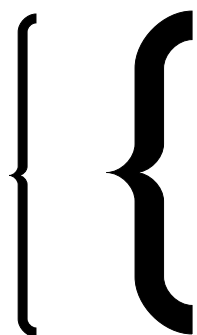


All of these characters are made according to the currently loaded encoding, by default this is Latin 1.

Char build stroke width

This is the stroke width of any new letters that FontFiend might add to the font. For fonts that do not have an *Oblique/Italic offset* (i.e. do not slope) you can leave this value at 0. FontFiend will scan the font itself and work out the stroke width of any new letters automatically.

The example on the left shows two brackets drawn by FontFiend. The first was drawn with a stroke width of 25. The 2nd with a stroke width of 80.



As you can see the bolder your font the higher the stroke width will be.

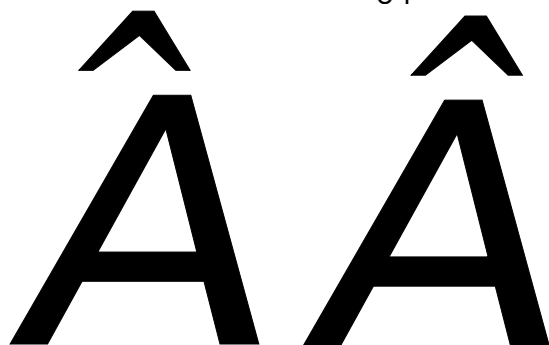
Italic offsets

If you are working on an Oblique or Italic font then you will need to set the Italic Offset in the Misc Data window before using the Character Builder. This is so

that FontFiend knows that the font has an angle and can adjust its stroke width and the angle of new letters it produces automatically. This also allows FontFiend to offset any accents it places over characters so that they are above the visual centre of the letter.



The example on the left shows two accented letters in an oblique font. The first was produced without setting the Italic offset in Misc Data. As you can see the accent is in the wrong place. The second was produced with the



Italic offset correctly set up. Setting the *Italic offset* is best done by trial and error.

Overwrites

There are two buttons that control how FontFiend behaves if it finds that a letter already exists.

Turning on *Overwrite existing letters* will allow FontFiend to replace any letters in the font with the ones it has made itself. If this button is off then only those letters that do not exist in the font will be drawn by FontFiend. Existing letters will be left alone.

For example say that you wanted to add all the math signs to a font. You could turn on just the *Math Signs* button and the *Overwrite existing letters* button. Then clicking on *OK* would just add math signs to the font, replacing any existing math signs with the new ones.

Overwrite existing accented letters should be left on most of the time. This allows FontFiend to replace any accented letters with its own. However other characters will only be replaced if the *Overwrite existing letters* button is on.

Misc Data button

Clicking on this button will open the misc data window. This is covered earlier in this chapter.

Encoding button

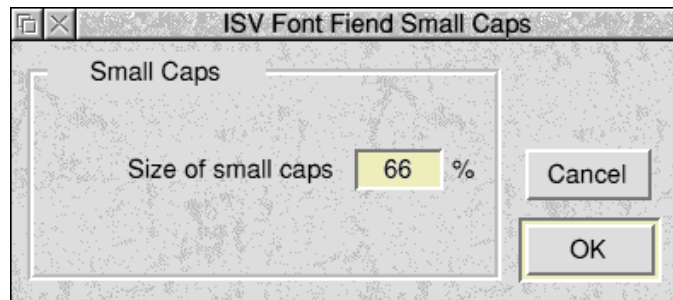
Clicking on this button will open the encoding window dialogue box from where you can choose a different encoding.

Small Caps



Clicking on this button will open the small caps window. From here you can produce small caps in your font.

Small caps are just that. Small versions of capital letters instead of having specially designed lower case letters. The size of the small caps should be between 66% and 80% for most fonts. Clicking *OK* in this window will generate the small caps. If you are going to add small caps to a font then you must do this before using the main char builder.



You may also need to alter the weight of each of the small caps using the *Alter Weight* option from the *Outlines.FX Process.Special FX* dialogue box. This can be used to make the small caps bolder so that their visual weight is consistent with other characters in the font.

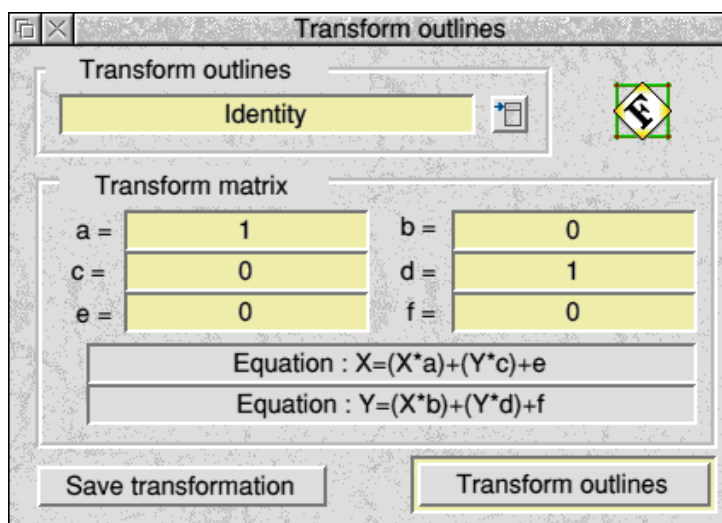
Building characters

Once you have set up the Character Builder you can then click on the *OK* button. FontFiend will then check to see if the font has the required accents. If not, a confirmation window will open asking if you want to load one of the default sets of accents. Click on *Continue* to load the accents suggested by FontFiend. The hourglass will then appear and FontFiend will build the letters you have requested.

It may seem obvious but FontFiend cannot produce miracles. In order for the Character Builder to work you will need a font with at least upper case letters, numbers and some symbols such as full stop and an exclamation mark. You cannot expect FontFiend to be able to produce missing lower case letters, or draw upper case letters in the style you have designed the rest of the font. Character Builder can save many hours of work. However you still have to do some of the work yourself when you are designing a font.

Transform

Clicking on this option will open the character transformation editor.



This dialogue box allows you to alter the appearance of a font and to make new font weights easily and quickly. The transformation used is a simple 6 entry one and is shown in the window in the X and Y equations sections.

If mathematical transformations are not your cup of tea then don't worry. FontFiend has a number of built in transformations, and extra ones available on disk in the examples directory.

You can choose a built in transformation by clicking on the menu button and choosing one from the menu that opens. Once you have chosen the transformation you wish to use click on the *Transform outlines* button.

If you have made your own transformation then you can save it to disk by clicking on the *Save transformation* button. A save dialogue box will then open from where you can drag the transformation to a filer window.

When you transform a font it is not just the outlines that are changed. Composite character inclusions will be re-calculated, accents will be re-centered over letters, scaffolds will be shifted so that they correctly fit the changed font. Also if you alter the X/Y oblique angle of the font the relevant scaffolds will be automatically disconnected as they will no longer be valid. Users of FontTrix and FontTrix Pro should be familiar with transformation matrices. Also see Appendix 5 for some hints on dealing with stroke width problems in transformed fonts.

Transformation examples

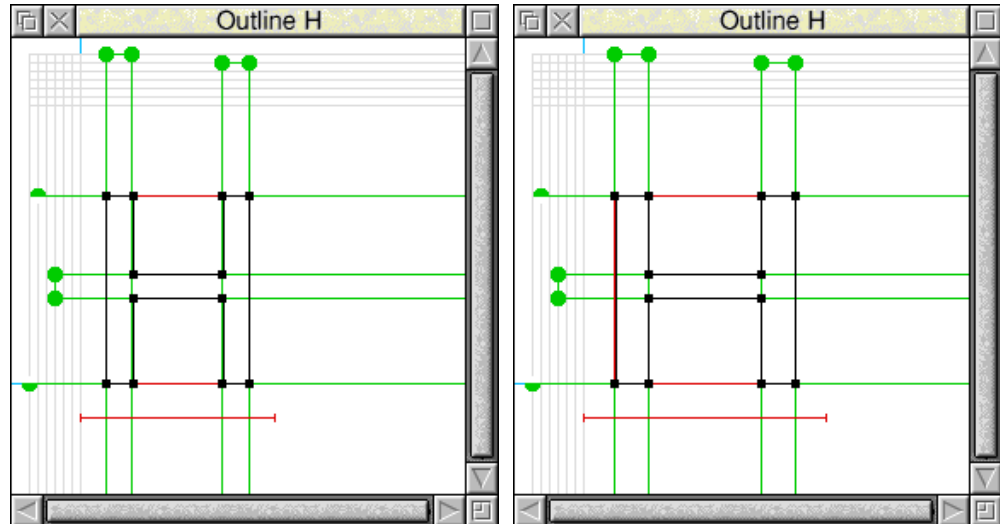
3QD Font Fiend



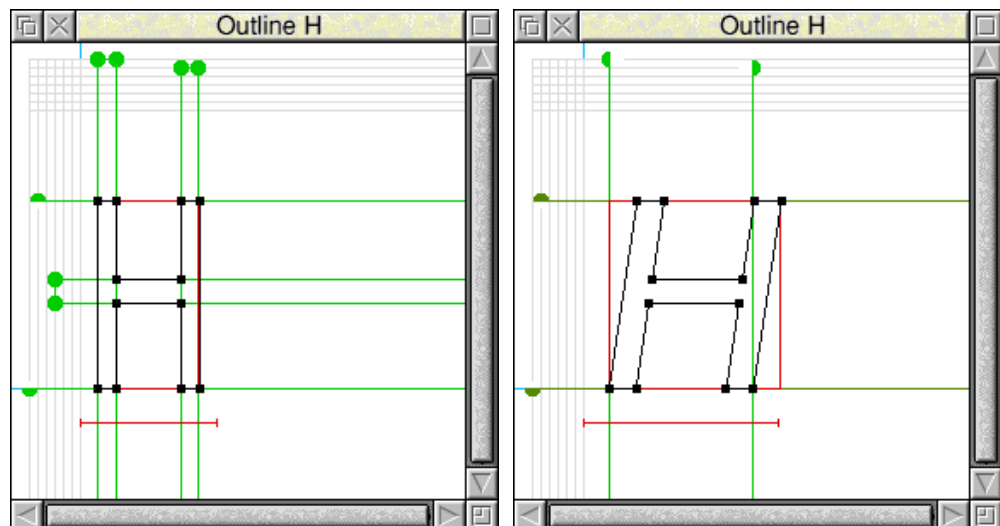
The following set of diagrams show some of the built in transformations and the effect they have on characters.

The first screen shot is the original character H (identity).

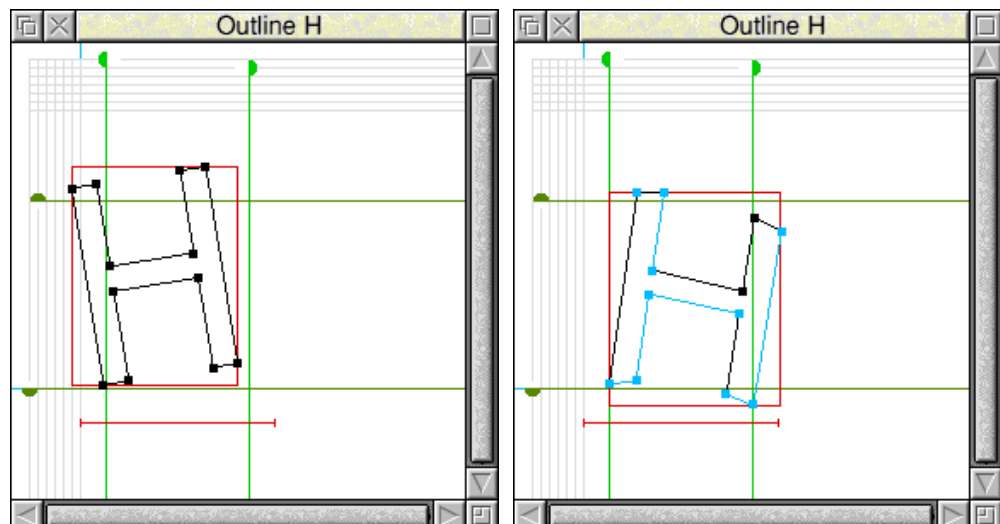
The 2nd is an extended version produced by FontFiend.



This pair of screen shots show Condensed and Oblique transformations.



You can of course combine transformations to produce other effects.



The Composites sub menu

This sub menu only has two entries.

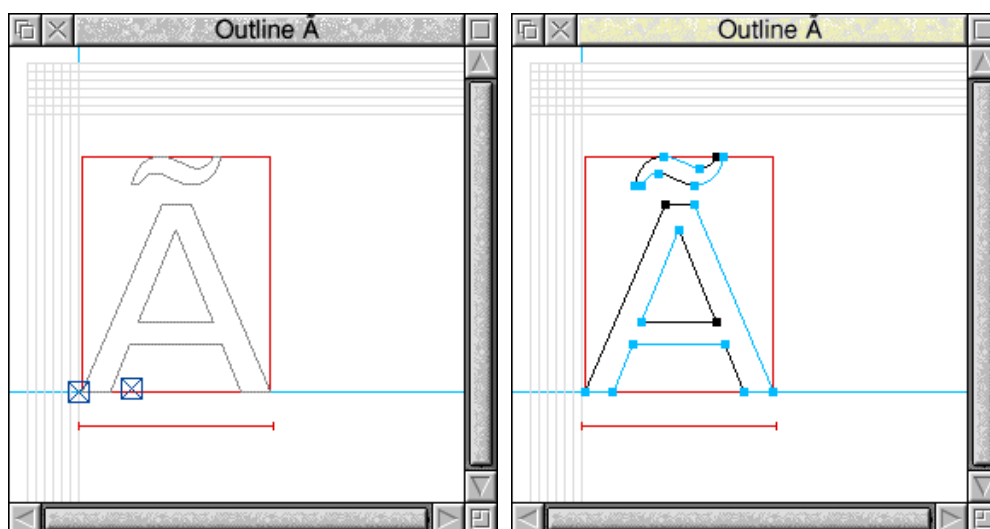


Delete

Clicking on this option will delete all composites from the font. This may well include all accented letters and quite a few symbols. Only click on this if you are sure you need to get rid of all the composites

Make real

Clicking on this option will also delete any composites in the font. Any composite character inclusions will be converted into real outlines.



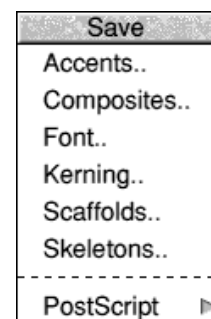
The example above left shows an accented letter. This composite character is produced from two normal characters. The position of the composites is marked by the two rectangles near the bottom left corner. The example above right is the same letter after *Make real* has been clicked on. The character now consists of outlines & skeletons.

The Save sub menu

This menu allows you to save out an entire font, or parts of a font. Any files you save out will be filetype as a font. However only the *Font* option will save out a file that can be used by another application.

Accents

Clicking on this will open a save dialogue box that allows you to save just the accents from the font you are working on to disk. They can then be imported into another font at a later date. Note that this file does not contain any accented letters, just the accents required to make accented letters.





Composites

Clicking on this will open a save dialogue box from where you can save the composites in your font to disk. Note that this file will only contain composite inclusions not actual letter shapes. So, for example, it may not contain definitions for most characters, accented letters may be defined as a copy of character n1 and a copy of character n2.

Font

When you click on this FontFiend will re-compile the font you are working on back into Acorn format. You can then save the font to disk. If the version number is shown as 6 in the FontFiend button bar then the font will work on any Risc OS computer. If the font is saved in version 8 format it will only work on Risc OS 3 or later, also version 8 fonts will be saved with misc data & kerning.

Kerning

Allows you to save the fonts kerning to disk. For example to keep the kerning intact before loading the font into FontED. You can also load this kerning file into iSV Products **FontKern™** kerning editor. Although you may need to change the file type of the saved kerning to data (FFD) before **FontKern** will be able to load it.

Scaffolds

Allows you to save a fonts scaffolds to disk. You could then import these into another font at a later date. The scaffolds file will contain all the scaffolds in a font, complete with the scaffold tree. When a scaffolds file is re-loaded the scaffolds are re-calculated to fit the new fonts bounding boxes.

Skeletons

Allows you to save the fonts skeletons to disk for later use. This can be very useful as the skeletons from one sans-serif font will very often be quite a good match for those from another sans-serif font. Also skeletons from one weight of a font can often be fitted to another weight by FontFiend.

See the section in this manual called "Importing other files" for more information.

PostScript

If the version of FontFiend you are using has a PostScript sub heading then FontFiend may be able to save out the following other types of files.



AFM

Adobe Font Metrics. This is the equivalent of an Acorn fonts Intmetrics file. It includes data on bounding boxes, widths, composites and kerning.

PFB

The PostScript equivalent of an Outlines file

Type 3

A complete Type 3 font

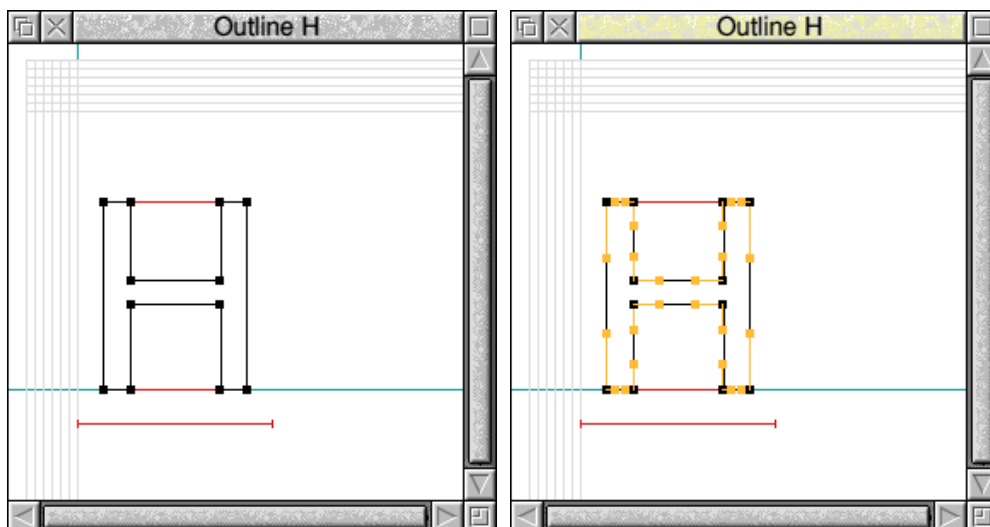
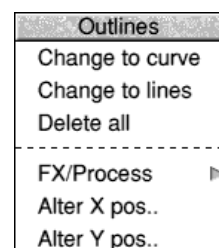
The base version of FontFiend (FontFiend8) can only save AFM files. These other files will be covered in an addendum to the manual supplied with the PostScript version of FontFiend.

The Outlines sub menu

This sub menu is entirely concerned with the outlines of a font. Most of the options on this menu will have no effect on skeletons, scaffolds or composite inclusions.

Change to Curve

Clicking on this option will convert all of the straight lines in a font to curves. If you click on this you may not see any change to the outlines in the font. However characters such as H which may be made of just straight lines (with no bezier control points) will now have bezier control points.



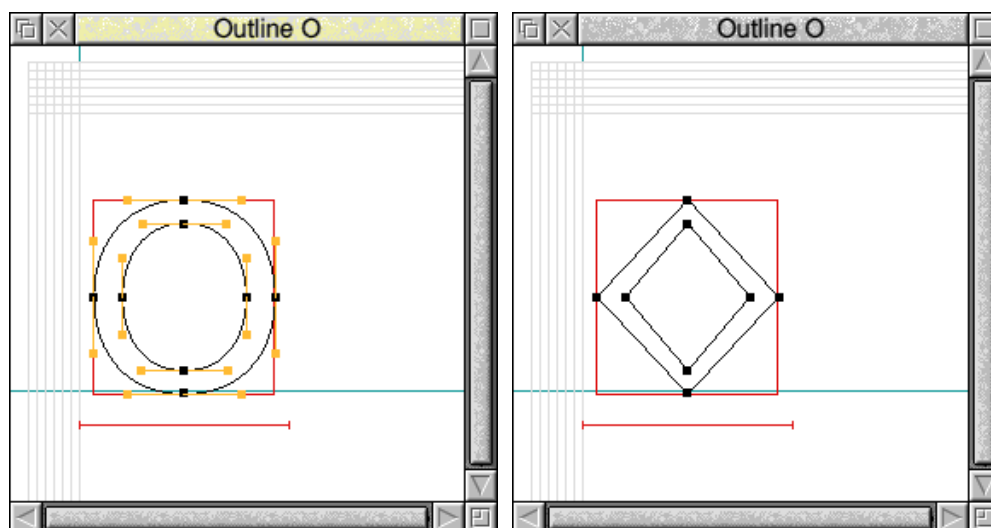
The letter H on the left does not contain any curves just straight lines. The letter on the right has had all the lines changed to curves.



Change to line

This option is the exact opposite of the last one. Any curves in the outlines of the font will be converted to straight lines.

The letter O on the left is the original. The one on the right is the result on changing all curves to lines.



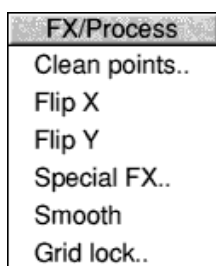
Delete all

Clicking on this option will delete all outlines from the font. This may not initially seem very useful since surely a font without outlines will be invisible, but it can be used for some interesting effects.

Try this for example, open the automatic skeleton window for the entire font from the FontFiend main menu. Now choose Copy Outlines as the skeleton model and click on OK.

This will produce a copy of the outlines as skeletons. Now delete the outlines from the font. The result is a font with a 1 pixel line making the shapes of the letters.

The FX process sub menu.

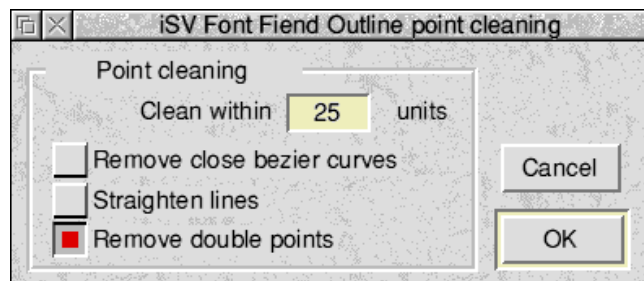


FX Process Sub Menu

The FX Process option leads to another sub menu concerned with special effects that can be applied to the fonts outlines. These effects will only apply to the outlines of the characters. They will have no effect on the skeletons or scaffolds.

Clean Points

Clicking on this option will open the clean points dialogue box for the font. This can be used to try and tidy up badly auto traced outlines in a font.



Note that it should be used with caution, if it is used on a well designed serif font the serifs can get damaged.

Clean within sets the minimum number of design units that can exist between two control points. Setting this value to 100 on a font with a design size of 1000 would cause points less than 0.1em apart to be removed from the outlines.

If the *Remove close bezier curves* option is ticked then FontFiend will try and remove bezier curves that are less then the clean within value. With clean within set to 100 curves less than 0.1em would be removed (on a font with a design size of 1000).

Straighten lines will attempt to straighten any lines that FontFiend thinks should be straight but in fact have a very slight curve.

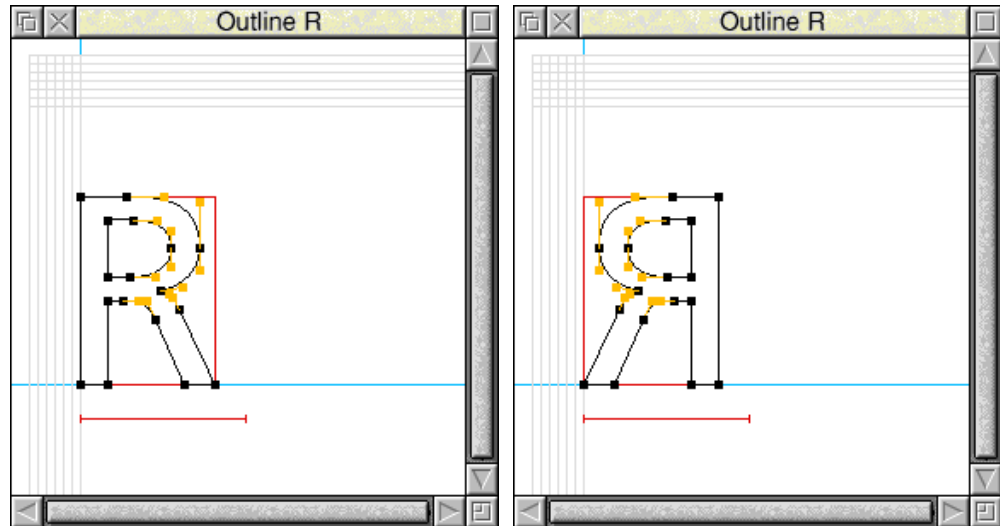
Remove double points will try and remove 2 points that sit almost on top of each other.

Clicking on *OK* will start the point cleaning process. On a well designed non decorative font using point cleaning may have no effect since points will not be close enough to be cleaned. On imported PostScript fonts that have been converted with Acorns Type1 to font application you may well find two control points almost on top of each other at the edges of curved character shapes. These points were inserted by Fontographer (a Macintosh font editor) to enable it to auto-scaffold the font. Clean points can be used to remove these points. Fonts with many very close control points can also crash PostScript devices and fail to render correctly using the Acorn font manager.

FlipX

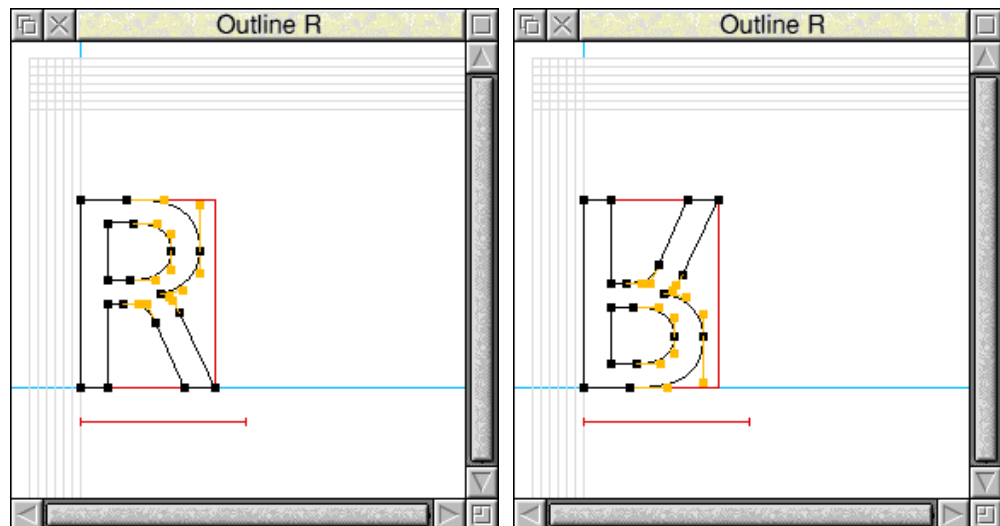


FlipX does just that, it flips the outlines of a letter over so that what you get is a mirror image of the original letter.



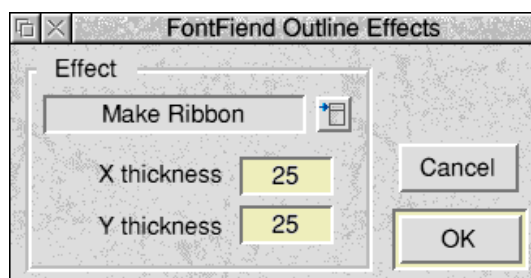
FlipY

FlipY is very similar to Flip X except the outlines are changed as though they were reflected in water.



SpecialFX

Clicking on this option will open up the SpecialFX dialogue box. This contains a number of special effects that can be applied to the entire font. The effects listed will depend on which version of FontFiend you have. To open a menu of effects click on the menu button.

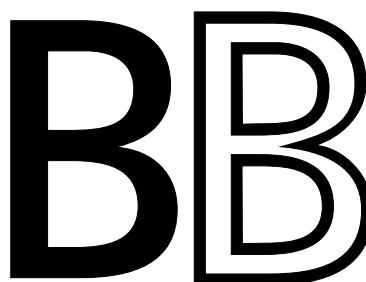


The following effects are standard on all copies of FontFiend. Some helpful hints on using SpecialFX can be found in Appendix 5.

Make Open

The X and Y thickness are an offset from the original outlines of the characters.

Shown on the right are the original character and the result of using *Make Open* set to an X and Y thickness of 15. Increasing the X and Y thickness will increase the thickness of the black sections in the 2nd letter.

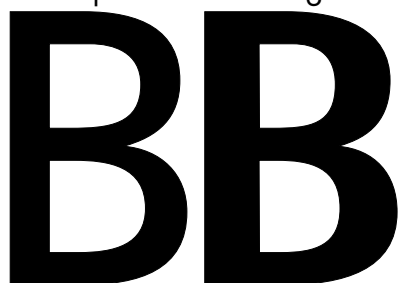


Make Open used on character B. Make Open will only work correctly on a font that does not have a non zero winding rule.

Alter Weight

The alter weight option allows you to alter the weight of a character in either or both of the X or Y directions.

This example show the original character and the result of increasing the weight by 20 in the X axis. As you can see the second character appears much bolder, but has the same cap height as the original character.



Altering the weight of a character in just one direction can be very useful,

for example when making small caps you will ideally want to alter the weight of these letters to match the rest of the font.

Alter weight can also be used to make a "lighter" version of an existing font.



In the example on the right the “lighter” character has been made by altering the weight of the original character by -10 in both the X and Y

axis. However trying to go too far and make a character too light can corrupt its outlines.

axis. However trying to go too far and make a character too light can corrupt its outlines.

Make Ribbon

Make Ribbon will turn the outlines of the font into a ribbon effect.

In this example the second character is produced using *Make ribbon* with a setting of 20 in both the X and Y Axis. This type of effect can be used to

produce quite dramatic looking characters.

produce quite dramatic looking characters.

The W character on the left was produced using *Make Ribbon* with a change of 50 in the X axis, but with no change in the Y axis.

The W character on the left was produced using *Make Ribbon* with a change of 50 in the X axis, but with no change in the Y axis.

This example is the same letter as above, however after using *Make Ribbon* the *Make Open* effect has also been applied using offsets of 5 units in both the X and Y axis.

This example is the same letter as above, however after using *Make Ribbon* the *Make Open* effect has also been applied using offsets of 5 units in both the X and Y axis.

You can try just setting one of the offsets and putting the other one to zero for some interesting effects. You can also combine the effects together as in the example W above.

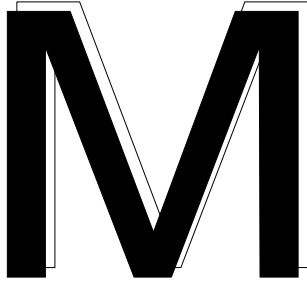
After using one of these effects you may need to re-apply skeletons to the font as they may no longer be valid.

Thin Shadow

This option will copy the outlines of the characters as skeletons and then shift them by the required amount in the X and Y axis.



In this example a *Thin Shadow* has been added with an offset of 20 in both the X and Y axis. This produces a kind of 3D letter effect. Using the



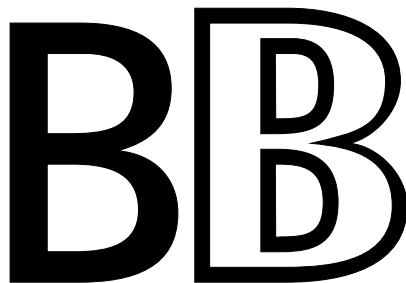
Thin Shadow option will overwrite any existing skeletons in the font.

Combination effects

As we saw on the previous page the effects can be combined together

In this example the original character B has had its weight altered in the X axis and then been made open by 20 in both the X and Y axis.

The best way to get familiar with the *Special Fx* available is to play with



them. A copy of this dialogue box can also be used from the single character menus (later in the manual).

Smooth

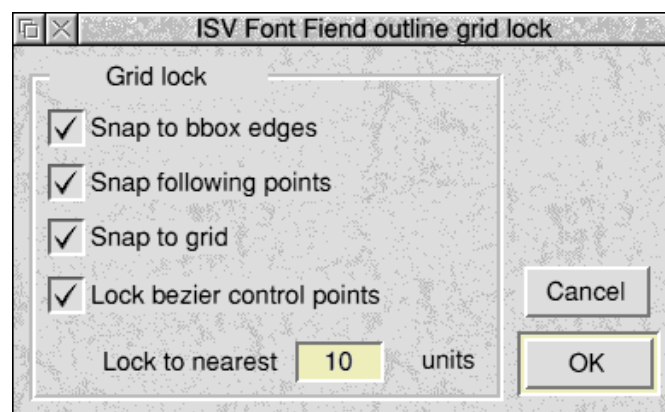
This tries to smooth out the outlines to remove any ugly pips or badly

designed bezier curve joins. It works in a similar way to the flatten joint option in Draws edit menu. Smooth is only designed to be used on decorative fonts and not on serif or sanserif fonts.

Grid lock

Clicking on this option will open the gridlock dialogue box, this is designed to attempt to tidy up slightly messy outlines by locking all of the control points to an invisible grid.

The *lock to nearest* value determines the size of the grid. The default is 10 which on a font with a design size of 1000 would be 1000/10 or a 1/

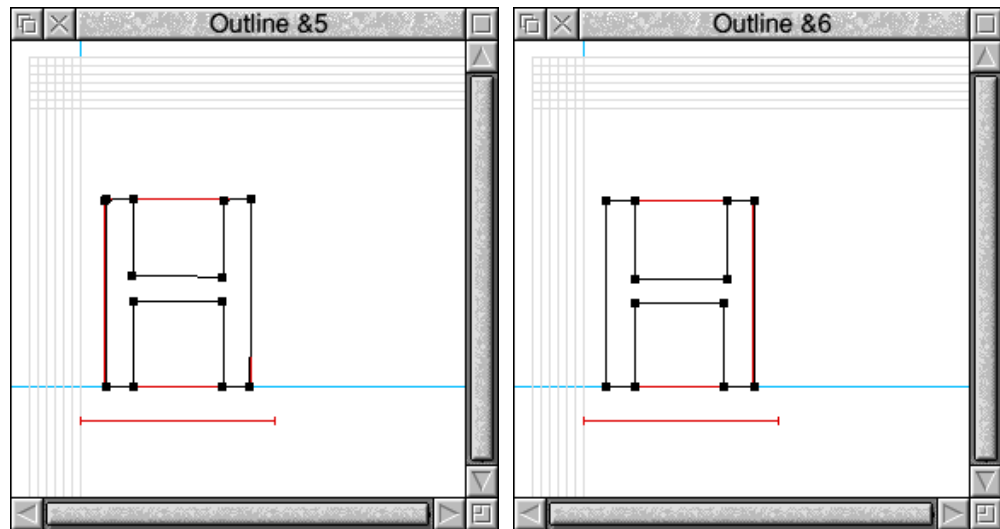




100 of an em. The larger this value to coarser the grid will be. *Snap to bbox edges* will force control points within the *lock to nearest* value of the bounding box to be snapped to the bounding box.

Snap following points will go around the letter and try and ensure that each point around the letter snaps to the same X or Y value as the previous

The original badly designed letter and the result of locking it to a grid using the options shown on the previous page.



control point.

Snap to grid will snap control points to the grid positions. You can, for example, have this option off and the *snap to bbox edges* option on. If this is the case then only those points within the *lock to nearest* value of the bounding box will be moved.

Lock bezier control points if selected will also snap all bezier control points, as well as normal control points, to the grid. If this option is off then bezier control points will not be snapped to the grid but will instead be moved by the same amount as their control point was moved by.

You can of course use the grid lock with any combination of these options. Generally the default (with all these options selected) works well. Note however that you may still have to do a little manual tidying yourself after grid locking a fonts outlines.

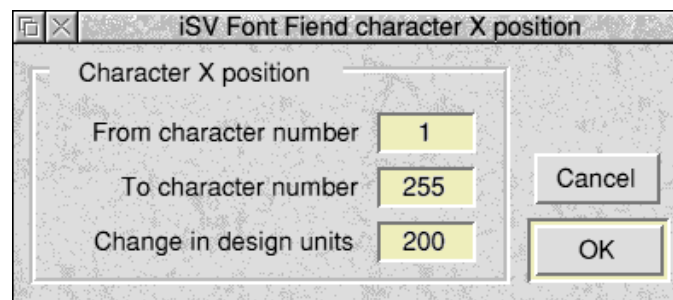
Alter X Pos

This option will alter the X position of the outlines of the font.

This dialogue box works in a similar way to the copy and delete characters dialogue boxes. You enter the first character you want to move and the last one. You then enter a change in design units, all characters covered will then be altered.



In the example above all characters would have a 200 unit offset added

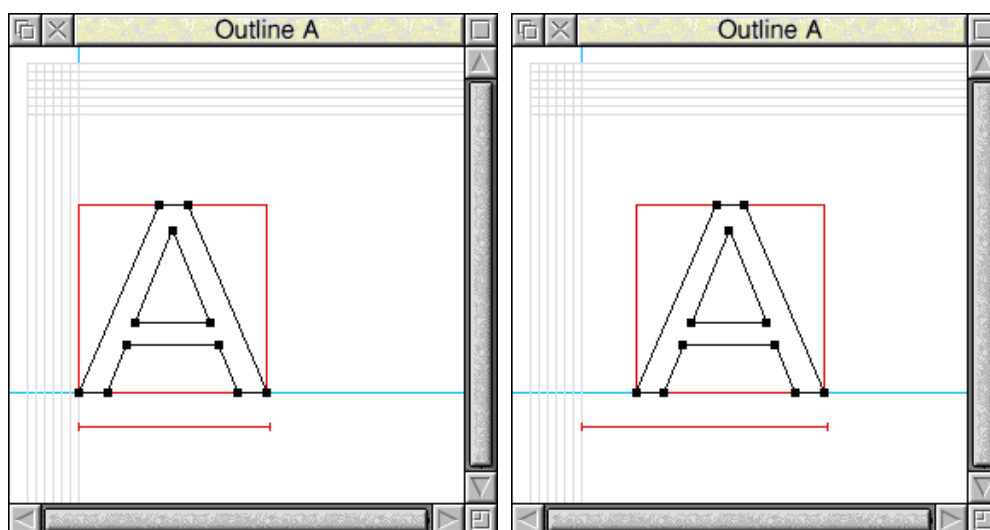


to the start of the character. Also all the widths would be increased by this value. Adding an offset to the left side of the character is called a left side bearing.

Entering a positive value will move all letters towards the right and entering a negative value will move them to the left.

Alter Y Pos

This works in exactly the same way as the Alter X pos dialogue box but moves the characters up and down. A positive value will move the

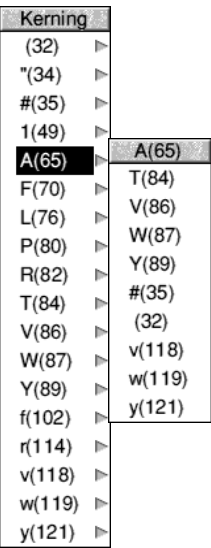


The original character and the result of adding an offset of 200.

characters up and a negative one will move them down.



The kerning editor showing the first kern pair in the Homerton Medium font. This pair is space A.



The kerning menu for Homerton Medium.

The Kerning sub menu

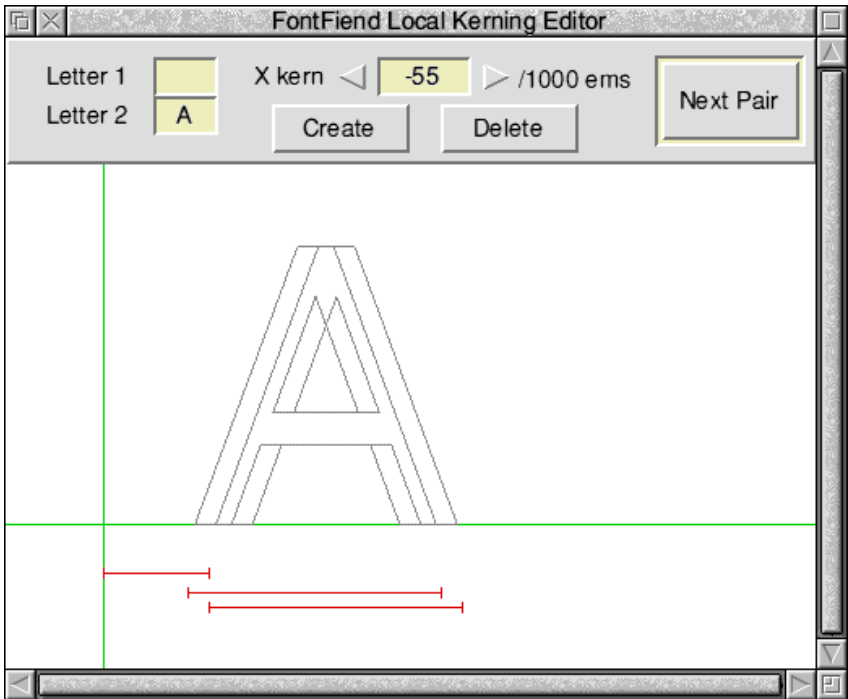
The kerning menu has three options, if the font you have loaded has no kerning then all of these options will be greyed out.

Delete

Clicking on this option will delete all kerning from the font. Normally this option would only be used if the font has very badly corrupted kern pairs or you wish to replace all kern pairs and start again from scratch using the kerning editor.

Edit

Clicking on this option will open the kerning editor for the font.



Using this window you can edit any existing kern pairs in the font, add new kern pairs or delete any kern pairs that are not required. The FontFiend kerning editor is very similar to the **FontKern** application supplied with iSV Products **Font Designers Toolkit**.

The kern pairs menu

Pressing MENU over this window will open a menu showing all the kern pairs in the font. Any kern pair can then be chosen by clicking on the menu. The menu is arranged so that each character that has kerning has an associated submenu. Any kern pairs that a character has will be shown in the submenu. Any kern pair selected from the menu will be displayed in the kerning editor window.

Create

Clicking on this button will create a kern pair from the two characters shown in the writable icons in the top left of the kerning editor window. If the pair already exists then instead of creating a new pair the existing pair will be displayed in the editor.



Delete

This option will delete the kern pair currently shown in the editor.

Next Pair

Clicking on this button will display the next kern pair in the font.

Editing kerning

Kern pairs can be edited easily using either the mouse or keyboard shortcuts. The amount of kerning applied to this character pair is shown in the toolbar at the top of the kerning editor window. Clicking on either the left or right arrows either side of the value will increase or decrease the amount of kerning.

The following keyboard shortcuts can also be used to edit the kerning.

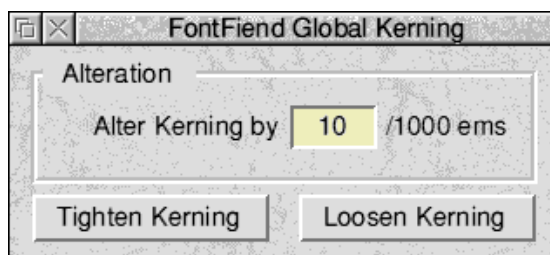
Cursor left	Increase kerning by 1 unit.
Cursor right	Decrease kerning by 1 unit.
Shift cursor left	Increase kerning by 4 units.
Shift cursor right	Decrease kerning by 4 units.
Ctrl cursor left	Increase kerning by 16 units.
Ctrl cursor right	Decrease kerning by 16 units.
Return	Finish editing this pair and display next pair.

Kerning is specified in terms of 1/1000 of an em, for example a kern amount of -50 will alter the gap between a pair of characters by 1/20th of an em when it is rendered

When you change the kerning you will see that the outline versions of the characters in the kerning editor window change, as do the width bars shown underneath them.

Global

Clicking on this option will open the global kerning editor dialogue box. This dialogue can be used to tighten or loosen all kern pairs in a font by a particular amount.

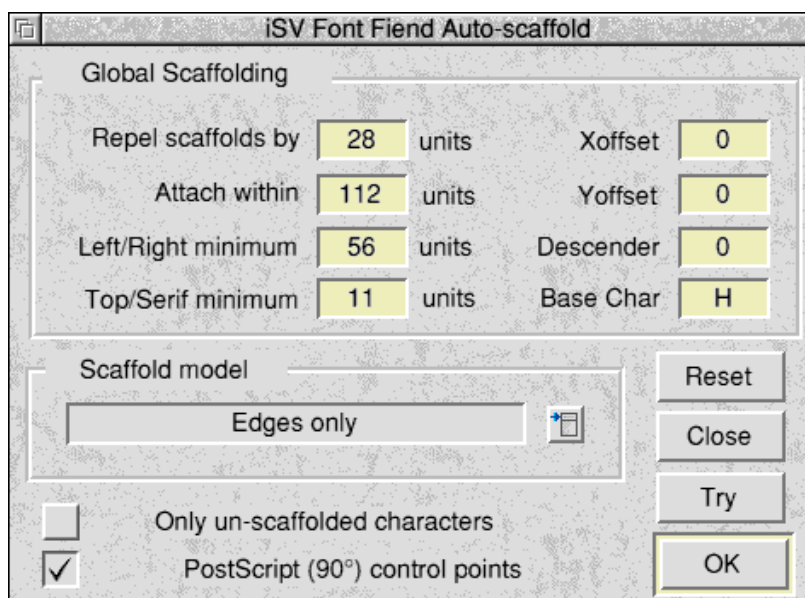
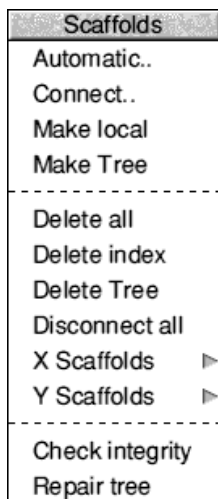


The Scaffolds sub menu

This menu is entirely concerned with scaffolds for the entire font.

Automatic

Clicking on this option will open the global font automatic scaffolder.



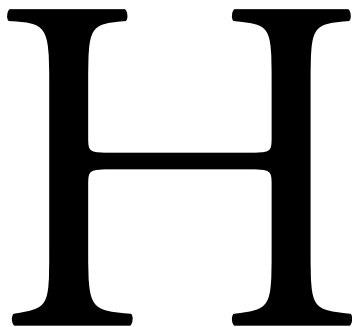
This was covered very briefly in the tutorial section of the manual. This is used to automatically scaffold an entire font.

Scaffold model

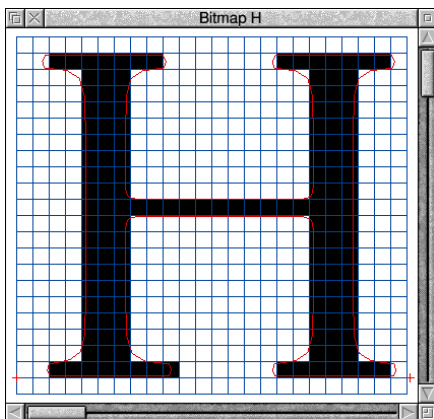
This is the current scaffolding system that will be used. You can change the *scaffold model* by clicking on the menu button. The default models are *Edges only*, *Wide edges*, *Sans-Serif* and *Serif*.

Edges only and *Wide edges* are designed to hint complex decorative fonts that cannot be hinted in the traditional manner. Basically these place scaffolds along each edge of the characters bounding box. They then attach control points in the character to the nearest scaffold. *Edges only* uses just tangent links, *Wide edges* uses wide links.

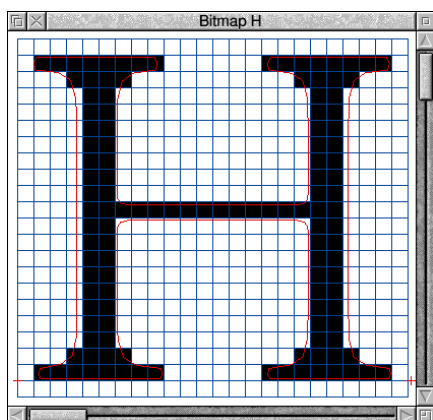
Font purists may be a little reluctant to use these settings since the results may not look correct. You may have read literature from some font suppliers claiming that only “hand-crafted” hinting is good enough for a font. This is not true. Both of the edges type scaffolding models can produce quite remarkable effects, this is because of the way the Acorn font manager works. In many cases adding edges only scaffolding will work as well or better than so called “hand-crafted” hinting.



For example here is a letter H from a commercial font supplied by another company. You can clearly see the curves of the serifs. Note that the cross bar of the letter is roughly half the width of the vertical strokes.



Shown here are two bitmap characters produced by FontED. The one on the left is the original character from the font with “hand crafted” hinting. Note the fact that the curves of the serifs have been squared off. Also note that one of the base serifs is longer on one side than the other. The difference in the stroke widths is also artificially exaggerated.



This example was produced from the same font re-hinted using *Edges only* scaffolding by FontFiend. The curves on the serifs have now been reproduced correctly and all the serifs are the same size. The proportions between the horizontal and vertical strokes are now more consistent with the original letter shape.

Both samples above were produced at the equivalent of 6pt on a 360 dpi printer. This is about as small as font would normally be printed. FontFiend also incorporates sans serif and serif hinting models. These are



designed to reproduce the “hand crafted” style of hinting. These scaffold models can hint a font with over 95% accuracy. Some manual adjustment of the scaffolds may be required in a few cases but this is still far quicker than doing the job totally by hand.

Obviously if your font is sans serif (like Homerton) you could use the sans serif hinting model, if your font has serifs (like Trinity) you could use the serif model.

Global scaffolding settings

These control how the automatic scaffolder works. Whenever a font is loaded into FontFiend it will be scanned and FontFiends best guess for these values will be automatically inserted in the writable icons.

Repel scaffolds - this controls how close together two scaffolds can get. Scaffolds will not be allowed to go closer together than this value.

Attach within - any control point within this distance to a scaffold will be attached to it.

Left/Right minimum - this is the minimum width that a wide link can be if it is at either the left or the right side of a character.

Top/Serif minimum - this is the minimum height of a wide link running across the letter at the top or the bottom.

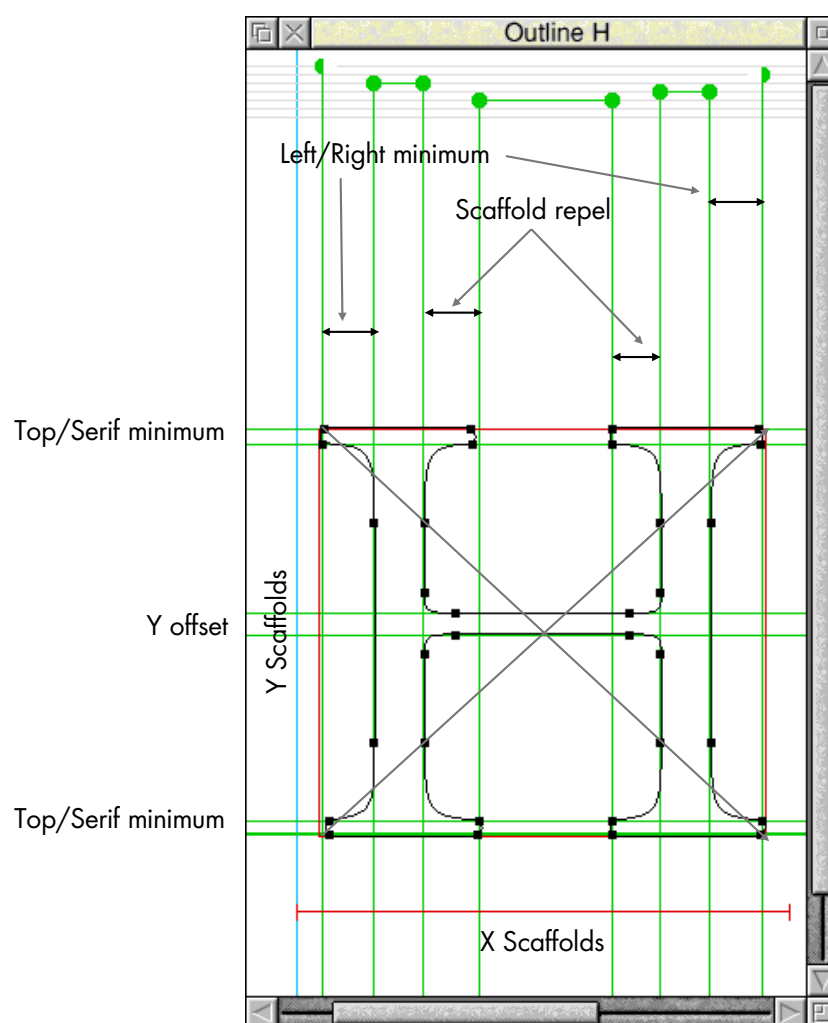
X offset - this is the offset in the X direction to the visual centre of the letter. For most fonts this should be left at zero.

Y offset - this is the offset in the Y direction to the visual centre of the letter.

Descender - this is the amount that the letter can be allowed to descend below the base line before another Y scaffold is added under the baseline. Most rounded letter shapes such as C and O will descend below the base line. If this value is left as zero then FontFiend will automatically place a scaffold for rounded letters and then link this scaffold to the one on the base line.

The illustration on the next page should make these settings clearer.

Base Char - this is not the scaffold tree base character as you might expect



but actually the character from which the values in the Global Scaffold dialogue box were taken. In the tutorial example earlier in this manual you will have tried scaffolding single letters in a font as a test before scaffolding the entire font. This character is the one that was last scaffolded when the *Copy to Global* button was pressed in the local character scaffolder.

Only un-scaffolded characters - if this option is selected then only characters in the font that do not already have scaffolds will be automatically scaffolded.

PostScript control points - click this button on if your font has PostScript control points. If the font has Acorn control points then we recommend you leave the button off. FontFier has in fact been designed from the outset to work with either type.

Reset - Clicking on this button will reset all of the values in the writable icons to the default for this font.



Close - This will close the auto-scaffolder.

Try - this will scaffold the entire font but leave the scaffolder window open afterwards in case the results were not quite correct.

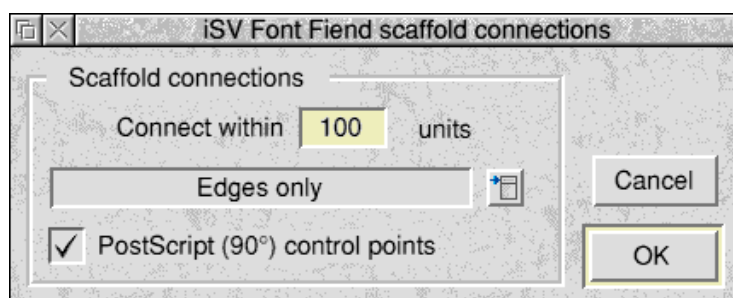
Ok - this is scaffold the font according to the settings given and then close the automatic scaffolder window.

When FontFiend scaffolds a font it also attaches all the control points in both the outlines & skeletons. This means that FontFiend can save even more time by performing another time consuming operation automatically.

Connect

Clicking on this option from the menu will open the automatic scaffold connections dialogue box.

This contains a small subset of the contents of the main automatic scaffolder. You can choose the scaffold model that was used to hint the font and the distance to connect scaffolds within.



Normally you will not need to use this dialogue since FontFiend will automatically connect scaffolds when it hints a font. However supposing some letters had been exported as drawfiles, altered and then re-imported, they would then not be connected to the scaffolds. You can then use this dialogue box to re-connect all scaffolds in the font easily and quickly.

Make Local

This option will convert all global scaffolds in the font to local ones. Obviously this will also remove the scaffold tree as there will no longer be any global scaffolds to make the tree. What will happen to the font is that wherever a character has a global scaffold it will be replaced with an identical local scaffold. All scaffold connections in the outlines and skeletons will stay intact.



Make Tree

This option will scan all the scaffolds in the font and attempt to replace as many local scaffolds as possible with global scaffolds. FontFiend automatically does this when scaffolding a font. However if you have hinted a font by hand this can save a great deal of effort.

Clicking on this option after automatically scaffolding a font, or on some commercial fonts will do nothing as the font will already have a well defined scaffold tree.

Delete all

This will delete all scaffolds in a font, both global and local and will also remove any scaffold tree present. This option is quite useful when you have a font that cannot load into FontED because the scaffolding is corrupted. Corrupted fonts usually report errors such as "Cyclic Scaffold Link".

Delete index

This option will delete the link index between scaffolds. For example on the character E the top and bottom scaffolds may be linked, and the centre scaffold linear linked between the top and bottom. Some fonts cannot be loaded into FontED because they produce the error "Invalid linear link". Loading the font into FontFiend clicking on this option and saving it back out will repair it so that it can then be used.

Disconnect all.

This option will leave all scaffolds in place but disconnect them from the outlines and skeletons. This is useful when scaffolds have been badly defined in a font. For example Y scaffold number 1 should ideally sit exactly on the base line of the character or the font may not work correctly. Some PD fonts have scaffolds arranged in the wrong place, Y scaffold 1 for instance could be at the top, rather than the baseline of the letter. The font may then suffer from strange rendering errors. Clicking on this option will get the font working again.

X & Y scaffolds

These two options on this sub menu lead to another sub menu. These extra sub menus allow you to delete or disconnect either X or Y scaffolds. For example if you took a normal font and turned it into Oblique you would



then probably want to disconnect just the X scaffolds since if not disconnected they could have detrimental effects.

Check Integrity

This option will check the entire scaffold tree for errors. If any characters are found to have faulty scaffolds then FontFiend will report those that require attention. For example if you have accidentally corrupted the scaffold tree in a font by dragging characters around on the main table then FontFiend will be able to tell you which characters are faulty.

Repair Tree

This option will repair any damaged global scaffolds, and hence repair the scaffold tree. Any invalid scaffold references will be removed from the font so that the font will then work correctly. We recommend that if you have a font with faulty scaffolding (e.g. one that makes FontED report a "Cyclic Scaffold Link" error) that you first click on the "Check Integrity" option. Make a note of any characters that are reported as faulty. Now click on "Repair Tree" and the tree will be repaired. Now check each of the characters that were reported faulty by hand to see if any scaffolds are missing. If any are missing then add them by hand.

The skeleton sub menu

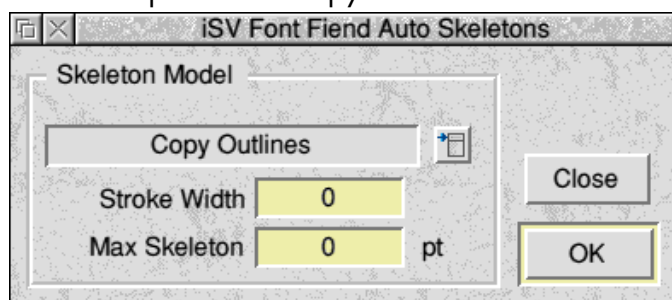
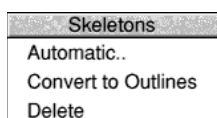
The skeleton sub menu is concerned with skeletons for the entire font.

Automatic

Clicking on this option will open the automatic skeleton dialogue box for the entire font.

This dialogue allows you to apply automatic skeletons to an entire font easily and quickly. The standard version of FontFiend has two built in skeleton models.

Copy outline - this will produce a copy of the outlines of the font as



skeletons. This means that each edge of any character is always rendered. The result is similar to that produced using a Type 3 PostScript font.

NDT V3 - this will copy 1 side of any stroke of the character as a skeleton. This ensures that no part of the character can “drop out” at any size since all strokes are always rendered at least 1 pixel thick.

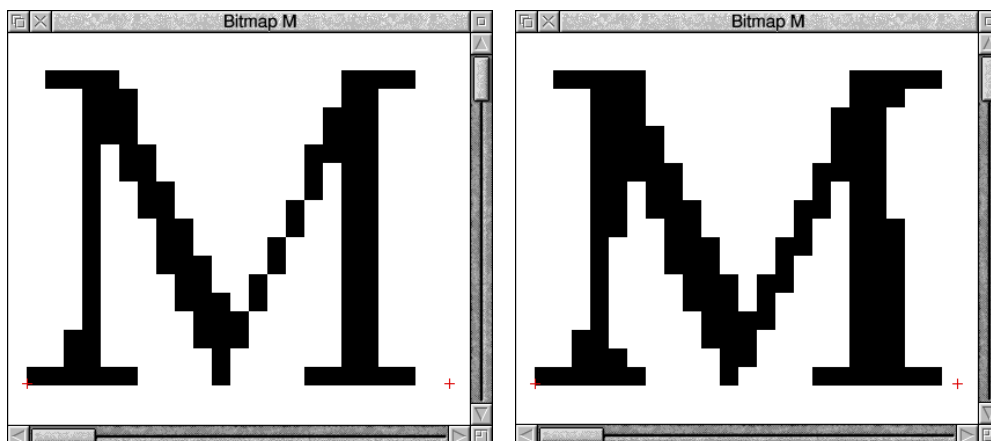


This dialogue also contains two writable icons. Stroke width is currently not used and should be left at 0. Max skeleton is the largest size a font will be rendered with skeletons. If the font is rendered larger than this size then the skeletons will be ignored. If this value is left at 0 then skeletons will always be rendered by the font manager, this is the recommended setting.

To apply skeletons to the entire font click on the *OK* button. To close with window without applying skeletons click on *Close*.

Although automatic skeletons are not quite as good as hand produced ones they still work very well at most sizes. The example below shows two 6pt letters reproduced at 300dpi. The sample on the left is with the original “hand-crafted” skeletons, the one on the right is with FontFiend skeletons.

FontFiend can apply skeletons to a font that has already been scaffolded



by hand and attach the skeletons to the scaffolds with a high degree of accuracy.

Convert to Outlines



This option will replace the outlines in a font with a copy of the skeletons. If you click on this option when you have a font that has no skeletons then all outlines will disappear. If you have a font with a full set of skeletons then these will become the outlines and there will be no skeletons.

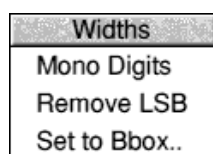
This option is generally only useful in very special circumstances, one of the FontFiend beta testers requested it for a problem they were having with a font that had a complete set of outlines which were defined in the font as skeletons!

Delete

This option will delete all skeletons in the font. This can be used to remove faulty skeletons, for example those that extend beyond the filled character shapes.

The widths sub menu

This menu controls the widths for the entire font.



Mono Digits

Clicking on this option will make FontFiend examine the numbers in the font and attempt to set their widths to be identical. This will involve moving some of the number characters left or right to centre them inside the new widths. For example the widest number in a font may be 8. This would then be used as a base for the other numbers. Character 1 would have to be moved so that it was exactly centered in the width of character 8.

Remove LSB

Clicking on this option will remove the LSB (Left Side Bearing) of the font. Many fonts will have no left side bearing. The left side bearing is the gap between the position 0,0 in the letter space and the start of the character in the letter space.

Removing the left side bearing on a font will also adjust all of the widths appropriately and will normally have the effect of tightening the character spacing.

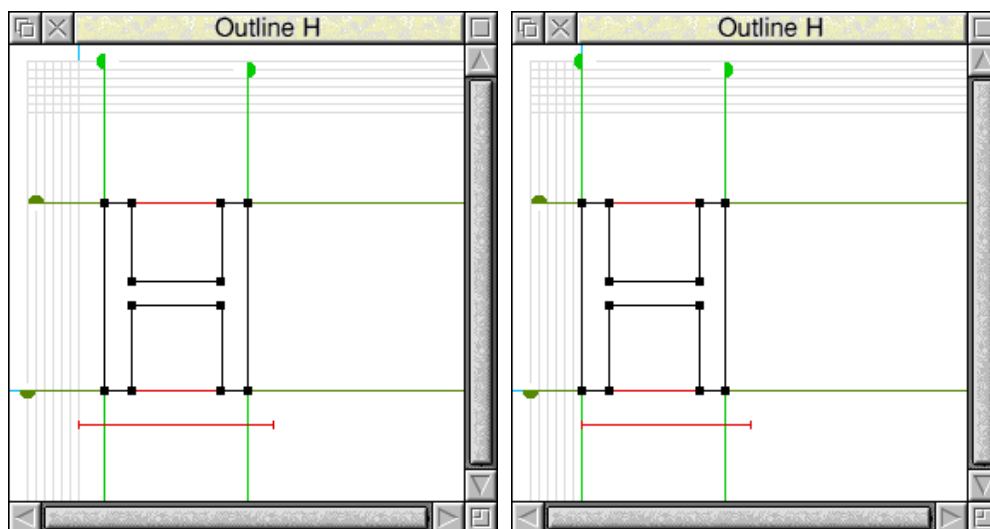
Normally this option will not have to be used on a font. It is provided to help recover fonts that have badly corrupted widths, or no widths at all. FontFiend can sets widths automatically using the *Set to Bbox* option shown on the next page. If the characters in the font have different left side bearings this may not produce sensible results. In this instance the existing LSB can be removed, then a fixed LSB added using the *Alter X Pos*

dialogue box. Now FontFiend can set the widths to sensible values using the *Set to Bbox* option.



Set to Bbox

Clicking on this option will open the FontFiend character widths dialogue

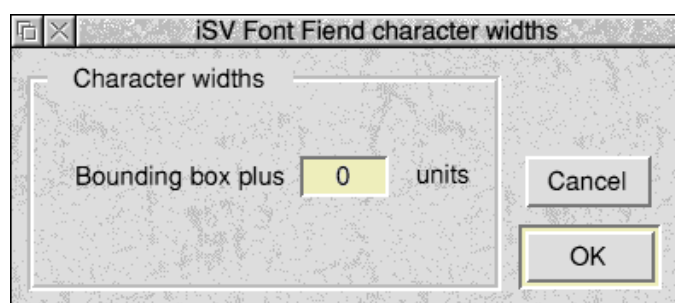


The original character H and the same with the left side bearing removed.

box. This is used to set widths for the entire font. This can be very useful when you have a font with corrupted widths, or indeed no widths at all. Some fonts converted with Acorns Type1 to Font application can suffer from this problem.

As you can see this dialogue lets you set the widths of all characters to their bounding box (the characters size) plus a value. An initially starting value of around 20 to 50 will often produce reasonable results.

Clicking OK will set all the widths in the font. We recommend that having done this you then check the font and make any adjustments that may be



necessary by hand.

This option can also be useful when you are making a font from scratch as you can quickly give the font a set of widths to check it.

Summary

This chapter covers all of the FontFiend main menu. We will now cover the



outline character windows and the single character menus. Similar versions of many of the dialogue boxes covered in this chapter will appear in the single character menus. Generally these dialogues will work in exactly the same way but will apply to a single character rather than to the entire font.

Chapter 6

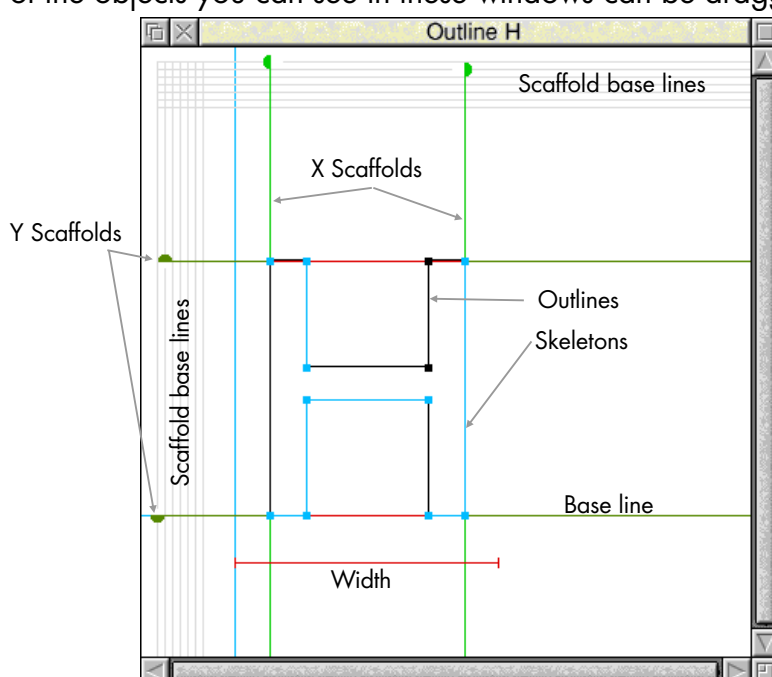
FontFiend outline character windows



FontFiend supports a different character window for each character in the font. These windows show the parts that go to make up the letters. These can include bounding boxes, outlines, skeletons, scaffolds and composites.

Outline character windows are opened by double clicking on the relevant character thumbnails on the main table.

Most of the objects you can see in these windows can be dragged around.



In all cases you should drag with the SELECT button.

If you are dragging an object you can also use the cursor keys for fine alignment. Cursor keys alone will move 1 design unit in the required direction. SHIFT and the cursor keys will move 4 units and CTRL with the cursor keys will move 16 units. When using the cursor keys you will need to keep either the SELECT or the ADJUST button held down on the mouse.

You can also move the entire character using the CURSOR/SHIFT/CTRL combinations, this will move outlines, skeletons, scaffolds and composites. To do this click in the window so that it has input focus and then use the cursor key combination to move the character.

Bounding Boxes

FontFiend will display the character bounding boxes as red rectangles



around the character. Bounding boxes cannot be moved or dragged since they are automatically calculated from the composites, skeletons & outlines.

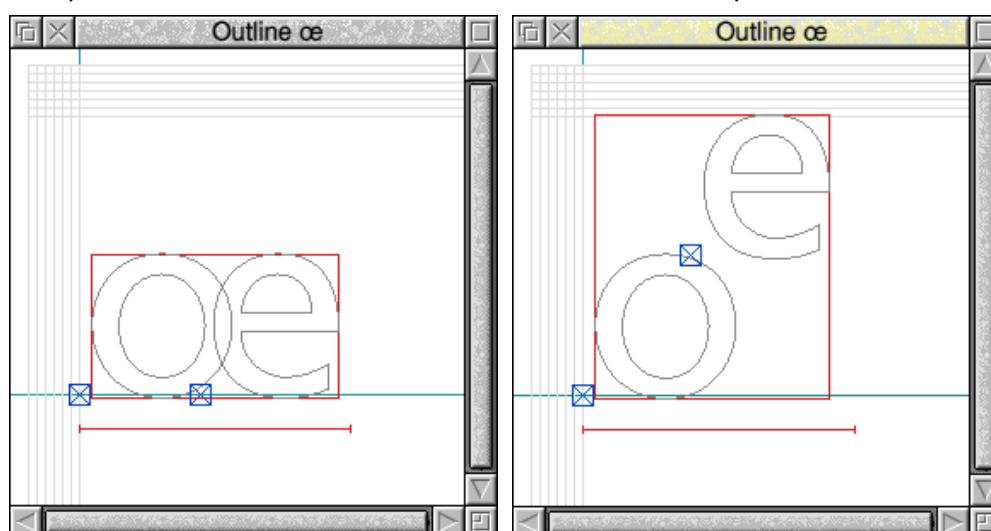
Composites

Composite characters are shown in the outline windows by a grey outline of the composite inclusion with no control points. Composites are moved around by dragging the blue rectangles. These represent the base points of the composites. For example if a composite insert is located at 0,0 then a blue rectangle will be at 0,0.

In the example below the e composite has been dragged to a new location.

When dragging composites it can be difficult to precisely align them. If this is a problem then the composites can be accurately aligned using the Composites List window. This is covered in the next chapter.

A composite character before & after dragging one of the composites around.



Scaffolds

Scaffolds can also be dragged around the local character windows. To drag a scaffold simply hold down the mouse button when the mouse pointer is over the scaffold handle and then drag. The relevant scaffold will then follow the mouse pointer. X scaffolds will only move in the X direction (left & right) and Y scaffolds will only move in the Y direction (up & down). Scaffolds are selected by holding down on the circle or half circles on the scaffold base lines at the top or the left of the character. When you click on a scaffold to either select it to drag it around it will change colour and become red to show it is highlighted.

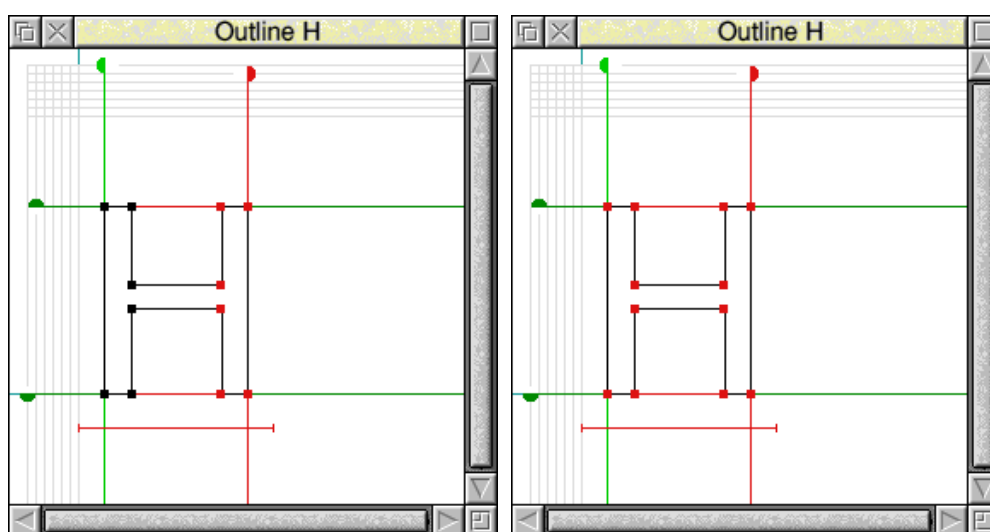
A highlighted scaffold will also “light-up” any control points that are attached to it. Outline control points that are attached will turn red, as

opposed to their normal black colour. Skeleton control points will turn orange as opposed to their normal blue colour.



You can attach further control points to a scaffold when it is highlighted by holding down the CTRL key and dragging over any control points you would like to attach.

Tangent link scaffolds can be moved just by dragging the half circles on the scaffold base lines. Wide scaffolds are a bit different. Dragging the bottom/left circle will move the entire scaffold. Dragging the top/right circle will leave the scaffold in the same place but alter its width.



The example on the left shows a highlighted scaffold and control points. The one on the right shows the result of holding down CTRL & dragging over the entire character.

Outlines

You can drag any characters control point around in the outline window. You can also drag bezier control points around provided that the *Display Outline control points* option is on in the FontFiend rendering preferences window.

If you need to make substantial changes to a character then you can easily OLE it into Draw using a single click with ADJUST in the outline window. The character will then be exported as a draw file and will appear in a draw window. You can now edit it using any of the Draw editing features. When you have finished editing the character simply save it and FontFiend will re-load it.

Some characters, such as g, may extend below the base line and when exported to Draw for editing may not be totally visible in the window. If this happens then you can drag the drawfile version of the character around so that it is visible in the Draw window. When you save the character from Draw FontFiend will reposition it correctly.



For more details on the editing of paths in Draw see the User Guide supplied with your computer (Page 222 in the Risc OS 3.6 user guide).

When you OLE a character into Draw its outlines will appear as a draw object with no fill colour and a thin black line.

Skeletons

Editing of skeletons can only be done by OLE editing a character into Draw. When a character is exported to Draw the skeletons will be shown as an object with cyan lines. This can be edited in the same way as editing outlines.

The currently selected character

This is always the last outline window you were working on, or the most recently opened window. For example you could have two windows open and select items in each one. Whichever one has a currently selected item (and hence a yellow title bar) will be the one with the currently selected character.

For example if you have the character *Special FX* window open along with 2 character outline windows and you click on the *OK* button from this dialogue box then the chosen effect will be applied to the currently selected character.

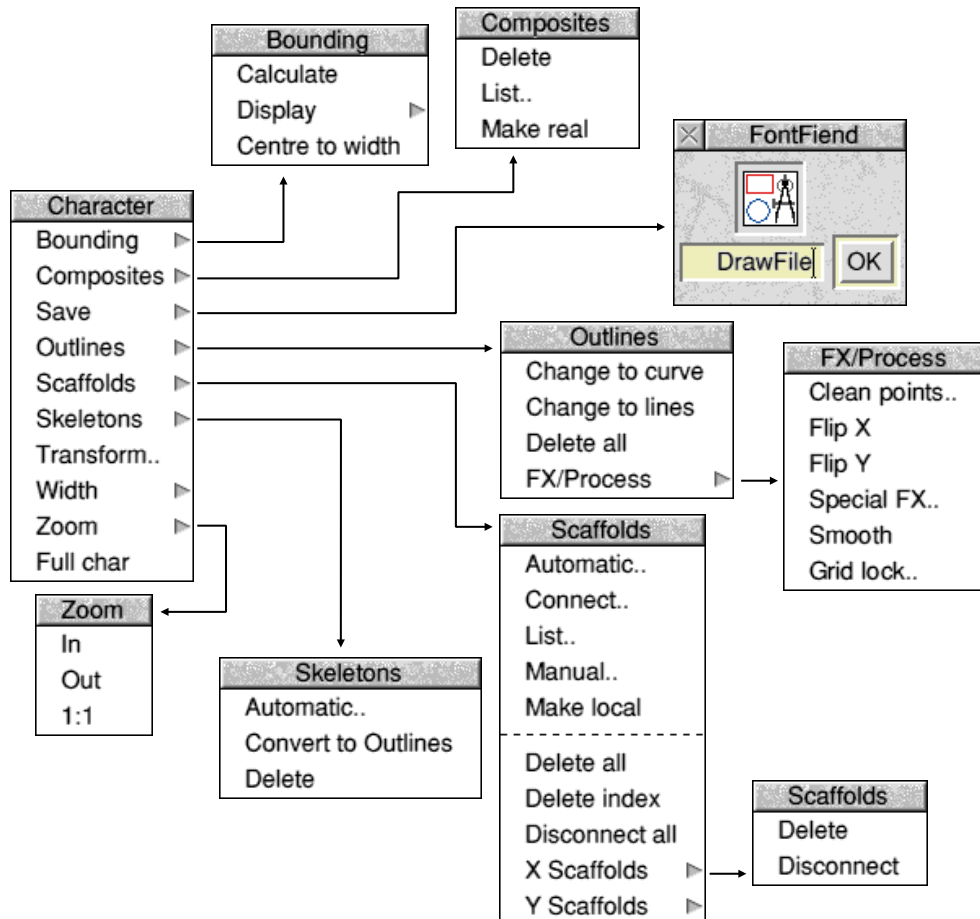
Chapter 7

FontFiend Character Menus



The FontFiend character menus are opened by pressing menu over one of the outline character windows.

The FontFiend character menu structure. This chart does not show some of the dialogue boxes such as Transform & width.



This chapter will cover the character menus. Many of the options on this menu are very similar to those on the FontFiend main menu. The difference being that where as the options on the main menu work for the entire font, these options only work for the currently selected character.

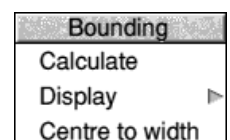
Bounding sub menu

This is concerned with the bounding box for this character.

Calculate

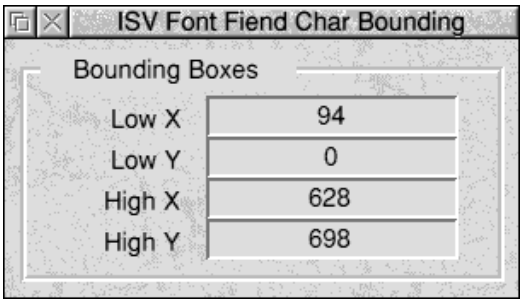
Clicking on this will force FontFiend to recalculate the bounding box for this character. The bounding box is always displayed as a red rectangle around the letter.

Display





Moving past this option on the menu will open a window showing the bounding box for the currently selected character.
The low X value in this window is the same as the left side bearing for this

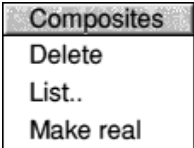
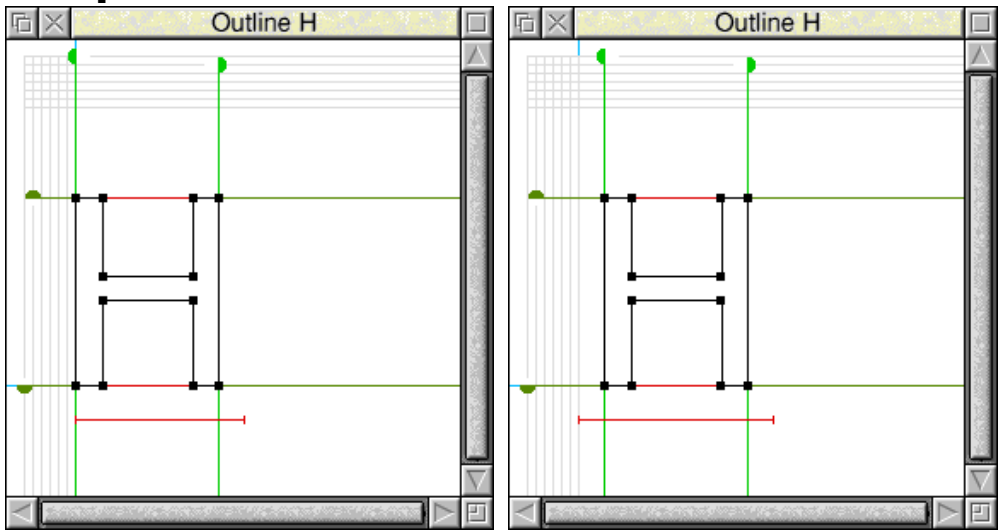


character.

Centre to width

Clicking on this option will move the entire character, including local skeletons, scaffolds and composites so that the bounding box for the character is exactly centered inside the characters width. The illustrations below show a before and after sample.

Composites sub menu



This controls composites in the currently selected character.

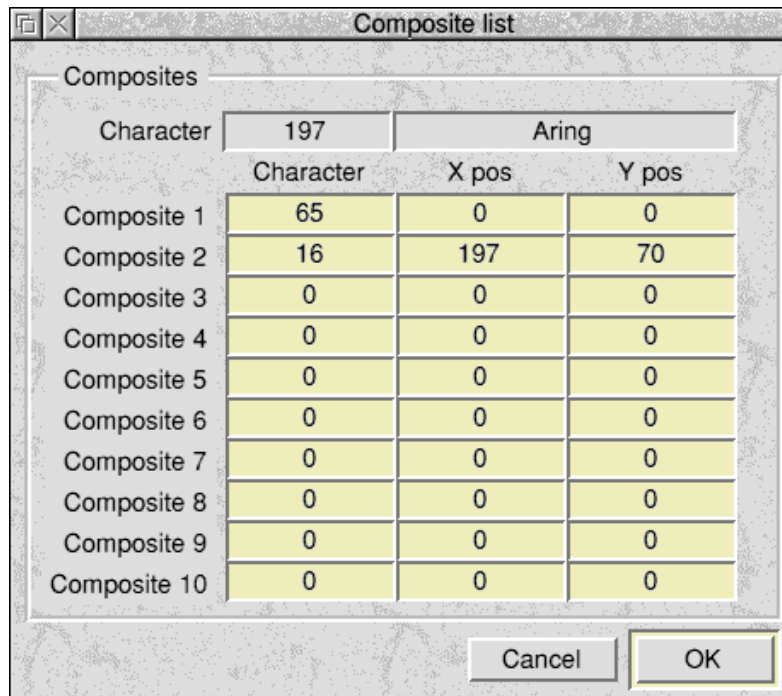
Delete

Clicking on this option will delete all composite character inclusions from the currently selected character. If you only want to delete one composite then you can do this from either the composite list window or by dragging using CTRL and ADJUST on the main character table (see the relevant chapter).

List

Clicking on this option will open a window listing all composites in the currently selected character.

The example above shows the character Aring. This is a composite



character consisting of a composite of A (ASCII 65 at offset 0,0) and a composite of ring (ASCII 16 at offset 167,70).

You can alter the characters used as composites and their X and Y positions. An X/Y position of 0 (zero) means that the composite inclusion will be rendered in the same place as the real character.

Clicking on OK will apply any changes you have made to this window. Clicking Cancel will undo any changes you have made.

Make Real

Clicking on this option in the menu will convert any composite inclusions in the currently selected character to real outlines. Using the example above the two composites (A & ring) would be removed and real outlines of the same characters would be inserted. These outlines will be in exactly the same place as the original composites would have been.

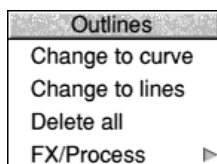
This means that the character will look identical but will no longer be composed of copies of other characters but of real outlines and skeletons. If you do this you may need to re-scaffold this character.

Save



Moving past this option on the menu will open a save dialogue box. From here you can export the outlines of character as a draw file. Even composite characters can be exported as draw files.

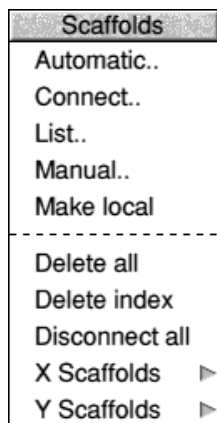
Outlines sub menu



This menu is very similar in structure to the outlines sub menu of the FontFiend main menu.

All of the options on this menu work the same way as their counterparts on the main menu, however they will only apply to the currently selected character. See the contents page in the front of this manual for the relevant page numbers for these options.

Scaffolds sub menu



This menu controls scaffolds for the currently selected character.

Automatic

The automatic scaffolder has already been covered in the tutorial and in Chapter 5. The single character auto-scaffolder is very similar to the global automatic scaffolder.

Connect

Clicking on this will open the local character scaffold connection dialogue box. This is identical to the one covered in the section on the FontFiend main menu.

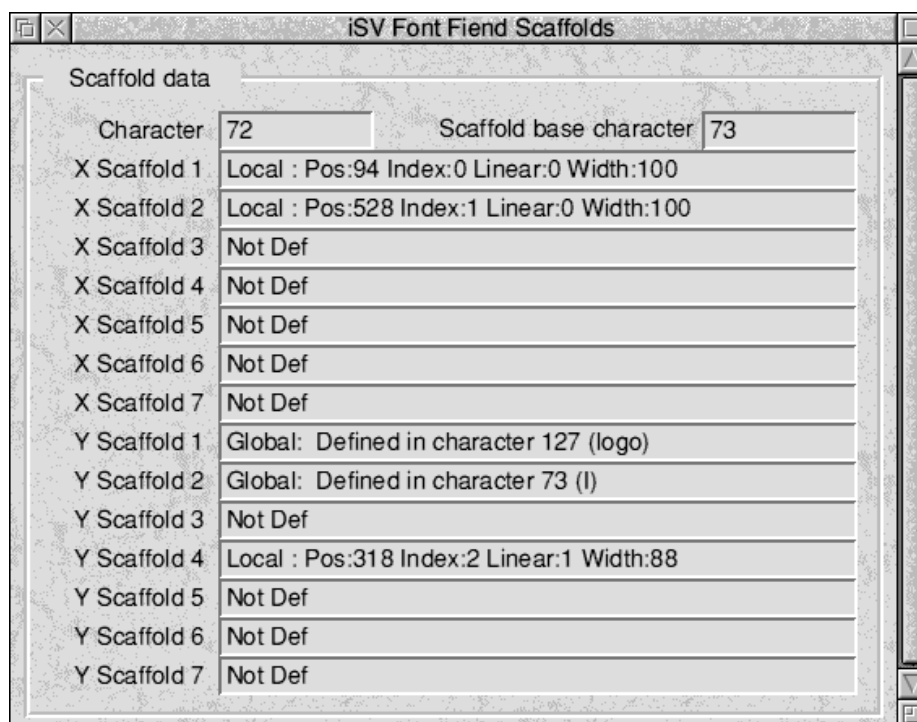
List

Clicking on this option will open a window listing all of the scaffolds in the currently selected character.

You cannot edit the scaffold positions using this window, it is merely designed to display information on the scaffolds.

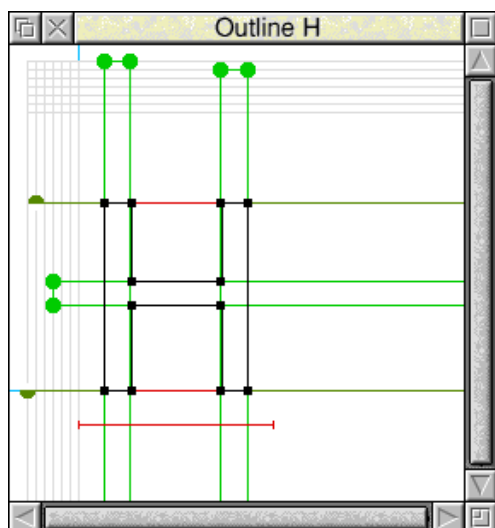
The currently selected character is shown, in the example shown on the next page this is H (ASCII 72). The scaffold base character is also shown. This is the character from which any global scaffolds in the selected character will be taken. In this case the scaffold base character is I (ASCII 73). As you can see two of the Y scaffolds are actually from the scaffold base character, they are shown as global on the list.

There are two local X scaffolds, and one local Y scaffold. An illustration of



The scaffolds list window.

the H character with its scaffolds would probably make this clearer.

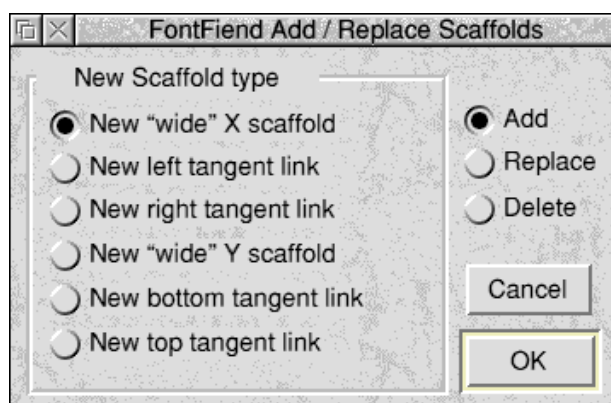


As you can see the character H has two local X scaffolds, one for each side of the character. It has one local Y scaffold, for the crossbar and then two global scaffolds, one at the bottom of the character and one at the top. These global scaffolds have come from character I (ASCII 73) since both H and I have exactly the same height.

Manual

Clicking on this option will open then manual scaffold dialogue box. From here you can add new scaffolds to the currently selected character. You can also delete any unwanted scaffolds as well as replace a scaffold with one of a different sort.

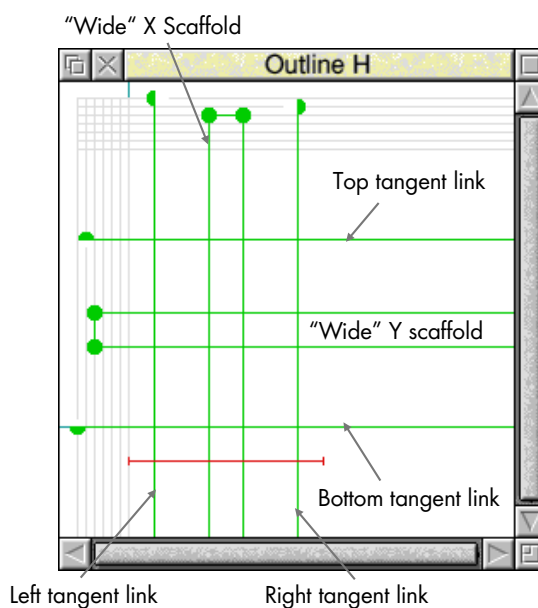
This dialogue contains 6 buttons which allow you to select the type of scaffold and three buttons for add, replace and delete.



If you want to add a scaffold then click on the scaffold type you would like and then select the Add button, finally click on the OK button. The relevant scaffold will then appear in the characters outline window. When adding scaffolds FontFiend tries to place them sensibly.

If you want to replace a scaffold then first select the scaffold you want to

The different types of scaffold available.



replace in the outline window (see Chapter 6). Then click on the type of scaffold you would like to replace it with in the manual scaffold dialogue box. Then click on the replace button and then finally on OK.

To delete a scaffold select it in the outline window. Click on the delete button and then on the OK button.

Make Local

Clicking on this option will convert all global scaffolds to local scaffolds. In the scaffold list example earlier this will remove the two global Y scaffolds and replace them with local versions.



Delete all

This option will delete all scaffolds from the character. This can have rather unexpected side effects if the scaffolds in this character form part of the scaffold tree. If this happens then the scaffold tree will become corrupted and FontFiend will ask you to click on the "Delete Tree" option from the scaffolds part of the main menu. You may then have to re-apply scaffolds to the font.

The rest of the menu

Most of the rest of the outline character scaffold menu behaves in exactly the same way as the scaffold sub menu from the FontFiend main menu. Remember of course that anything you select from this menu will only apply to the currently selected character and not to the whole font.

The skeletons sub menu

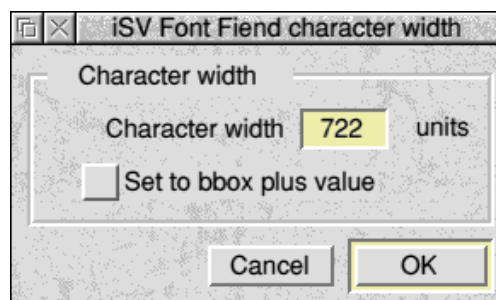
This again is identical to the skeleton section of the FontFiend main menu except the options apply only to the currently selected character. See the Main menu chapter for more details.

Transform

This is almost identical to the *Characters.Transform* option from the FontFiend main menu. The only real difference is that the local character version does not have a button which allows you to save the transformation and of course it only changes one character.

Width

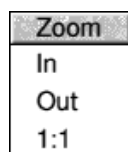
Clicking on this opens the local character widths dialogue box. This shows the width of the currently selected character. The width is a number in





design units. If your font has a design size of 1000 then a width of 722 (as shown on the previous page) means the character has a width of 722/1000 ems.

You can change the characters width by clicking in the writable icon, changing the value and then clicking on OK. You can also set the width to be the characters bounding box plus or minus the value in the writable icon by clicking on the Set to bbox plus value button.



Zoom sub menu

This menu controls the size of the outline character window.

In

Clicking on this will increase the magnification factor of the outline character window. The window will get larger as will your view of its contents. CTRL W can also be used to zoom in to an outlines window.

Out

Clicking on this will decrease the magnification factor of the outline character window. The window will get smaller as will your view of its contents. CTRL Q can also be used to zoom out an outlines window.

1:1

Clicking on this will restore the window to its default size. This is the size the window was when it was first opened.

Full char

Clicking on this option will open the full (filled) character window for the currently selected character. This is the window that shows how the character will look like when rendered.

Summary

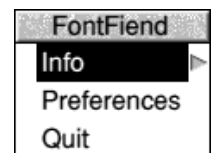
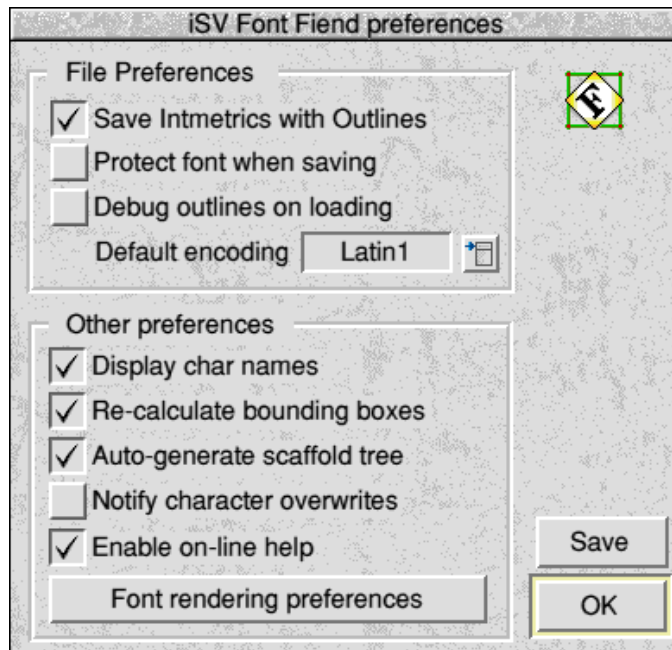
That covers the local character menus. Many of the options on the menus are identical to their counterparts in the main menu. Like any new piece of software it will take a little time to get used to FontFiend. However by making things operate in a consistent manner we hope to make FontFiend as easy as possible to get used to.

The reset of this manual will cover the preferences system, the glossary of terms and appendices.

Chapter 8

The preferences window

The preferences window contains a number of option buttons. It is opened by clicking on the prefs option on the FontFiend icon bar menu. Those under the file heading control how files are saved and loaded.



Save Intmetrics with Outlines

With this button selected FontFiend will automatically save new Intmetrics files with a fonts Outlines file when an outlines file is dragged to a filer window.

Protect font when saving

This will save out a font that will work with the font manager but that cannot be loaded into FontED and examined by a user.

Debug outlines on loading

This option will spool a very large text file to disk when you load a font into FontFiend. This option is not very useful for the casual user, however for those who are highly experienced with font design and the Acorn font format this file can produce a lot of useful information.

If you ever have problems loading a font into FontFiend than try loading the font with this option set. By reading the text file you may be able to work out why the font will not load.



Default encoding

This is the default character encoding that FontFiend will use when it is loaded. If you are not sure what an encoding is then consult the glossary in the back of this manual. Unless you are specifically trying to produce fonts for languages other than English you can leave this set to Latin 1.

Display char names

With this button selected the thumbnails in the FontFiend main window will have their Latin1 ASCII characters displayed in green in the top left of each box.

Re-calculate bounding boxes

With this option selected FontFiend will automatically re-calculate character bounding boxes whenever any change is made to any part of a font. Normally this button should be left on. If it is off then FontFiend will leave all bounding boxes the same as when the font was loaded. Turning this button off can be useful if you wish to load a font that suffers from "Font Cache Full" errors. Remember to turn this button back on if you add/alter any characters in a font.

Auto generate scaffold tree

This option button should be left on under most circumstances. When FontFiend scaffolds a font it can use global scaffolds to reduce the fonts size. If this option is on then fonts will contain global scaffolds. If this option is off then fonts will be scaffolded with each character treated as an individual object. However always remember that if you scaffold a font with this button selected and later on change some of the scaffolds you may corrupt the scaffold tree. If this happens then you will need to delete the scaffold tree and then re-scaffold the font again.

Notify character overwrites

With the button selected FontFiend will confirm any action that could overwrite any character in the font. This button can be useful in preventing accidents, however if you try and use the FontFiend Character Builder with this button selected you will have to confirm every character change (perhaps as many as 400 changes).

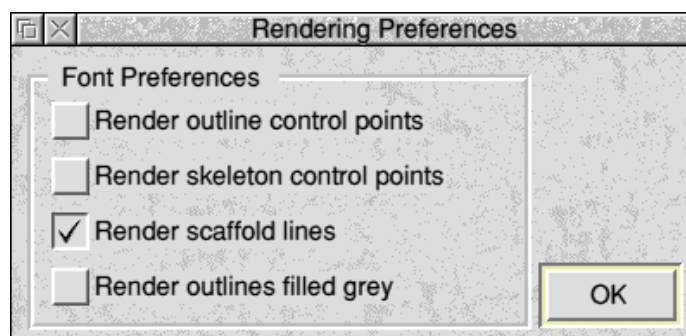
Enable on-line help

FontFiend has its own on line help system which is displayed in the button bar on the main window. You may wish to turn this off and this button will do it for you. If you turn off on-line help then you can still use the Acorn Interactive Help system.



Font Rendering Preferences

Clicking on the Font Rendering Preferences button will open the font rendering preferences window. Another copy of this window can also be opened from the Misc entry at the top of the FontFiend main menu. This window controls how FontFiend displays information in one of the Outline Character windows. It has no effect on filled windows or the main table.



Render outline control points

With this option selected any outline bezier control points in the outlines of a character will be rendered as small rectangles.

Render Skeleton control points

With this option selected any skeleton control points will be rendered as small rectangles.

Render Scaffold lines

If this option is on then all the scaffold lines in a given character will be rendered. If you are just going to use one of the FontFiend automatic scaffolding models and not fiddle with the scaffolding then you could turn this option off.

Render outlines filled grey

If this button is on then the outlines of a font will not just be displayed as black lines they will also be filled a mid grey colour.



Setting preferences

Once you have made any changes to the preferences you can either choose to save these settings to disk for future use, or just use them now.

If you want to save these preferences then click on the Save button in the main preferences window. If you just want to use them now then click on the *OK* button.

Appendix 1 - Glossary of terms



Accent

Defined by Collins as “a mark over a letter to show how it is pronounced.” However note that accents can be either above, below or to one side of a letter depending on their type. For Example Å, È, Ç

ASCII code

American Standard Code for Information Interchange. Each letter in a font has an ASCII number. Very simply, when a key is pressed on your keyboard an ASCII code is sent to the computer, the computer then displays the correct shape of character for that code on the screen.

Ascender

Any part of a lower case letter that extends above the x-height. The upper stroke on a lower case h or b etc. The ascender on a font may be slightly higher than the Cap-Height which is the height of Capital letters such as “A”.

Base 0 (zero)

Base 0 fonts (such as Homerton & Trinity) contain all of the Latin 1/2/3/4 encodings in 1 extra large font. FontFiend can load these fonts and convert them to any encoding. Note that some font suppliers supply fonts in Base 0 format, but with only the Latin1 letters defined. Base 0 fonts can easily be identified by having a 0 on the end of the Outlines file. For example Resources:\$.Fonts.Homerton.Medium.Outlines0.

Base line

An imaginary line that letters sit on. In fact very few letters actually sit on this line. Round characters such as “c” and “o” sit slightly below the base line and above the x-height as round shapes look visually smaller.

Bezier

A bezier is a mathematical description of a curve. Outline fonts use bezier curves to describe letter shapes, in a very similar way to Draw files. Bezier curves consist of a start and end point and two control points that alter the curve that joins the start and end.

Bitmap font

The shapes of the letters in bitmap fonts are actually small pictures made of



pixels (sprites). When enlarged or reduced they distort and the pixels begin to show. When the Outline font manager displays a font it basically draws a sprite of each letter on screen in turn.

Blackletter

A typeface which has the appearance of being created with a thick angled pen nib. Blackletter fonts are often wrongly termed Gothic or Old English. The first typeface set by John Guttenburg, one of the earliest designers of movable metal type, was a Blackletter face.

Body copy

The main part of a page of text, not headings or captions. The "Base Style" in Impression should really be called the Body Style as it affects all of the text that does not have a different style applied.

Bold

Font attribute denoting the thickness of the strokes that make the characters. There is no definitive measure of boldness or lightness, one typeface's bold may actually be lighter than another's Regular. The designer of the typeface decides how bold his bold is. Other similar attributes are extra light, light, book, regular, demi bold, ultra bold, heavy, black etc.

Bullet

Large or bold dot • used for emphasising or separating points in a list. Not to be confused with a period centered · which is a decimal point, although it is often used as a small bullet.

Cap height

The height capital letters extend up to from the base line. Some letters, for example rounded ones or sharply pointed ones, extend above this. Lower case letters such as "h" may extend above the cap height.

Character

A single letter in a font. Not just the normal A-Z and letters but also symbols such as Ð. Some fonts such as Dingbats contain only symbols as characters. Characters contains outlines, skeletons, composites and scaffolds. See chapter 1 of this manual for more information.

Character Set (Encoding)

A character set is the entire list of letters and or symbols available in a font.

Most Acorn fonts have a Latin 1 character set. The character sets are designed by the International Standards Organisation for Acorn machines and by Microsoft for Windows running on an IBM computer.



Condensed

Style attribute of a font: Decker Condensed for example. Condensed characters cannot normally be automatically made by a computer. Condensed typefaces have to be redesigned, otherwise they only appear "squashed". FontFiends *Alter weight* dialogue can solve this problem by altering stroke widths in one direction.

Cursive font

A font with a hand written appearance. This name is also applied to fonts that have no upper case letters, often also referred to as "Celtic" fonts.

Descender

Any part of a letter that extends below the base line: The "tail" on a lower case g or q etc.

Dingbat

Small pictogram used as a bullet or other type of attention grabbing device or indicator. Some fonts contain only symbols and no letters. Examples include ITC Zapfs Dingbats designed by Herman Zapf.

Decorative faces

Typefaces designed to be used only in small amounts at quite large sizes. Typically these will have only capital letters and are quite difficult to read when used for body copy.

Em dash

A dash, not a hyphen, as long as the point size of the typeface. See em

Em or em space(Unit of measurement)

Usually a space or gap in text the length of which is the same as the point size. For example, if you are setting 36pt text and have instructions to indent the first line of a paragraph by 1 em, then the line should be indented by 36points.

En(Unit of measurement)

Half an em. If the typeface is set at 36 point then 1 Em would be 36 points



and an En would be 18 points.

En dash

A dash, not hyphen, half as long as the point size of the typeface. See en

Encoding

An encoding is a text file which lists all the letters that are in a font and what ASCII numbers they have. Most fonts have a Latin 1 encoding, they have all the letters required to type in western European languages. Fonts with a Latin 2 encoding have the letters required for eastern European languages (such as Polish). FontFiend can make fonts with either Latin 1,2,3,4 or custom encodings. For information about which languages are supported by which encoding see the Encoding Tables section in the back of the manual. Also see the character set entry.

Extended

Style attribute of a font. Decker Extended for example. FontFiend can make extended fonts using a combination of the *Transformation* editor and the *Alter weight* dialogue box. (See Condensed)

Fixed pitch font

A font in which every character takes up the same amount of horizontal space like characters made by a typewriter. These fonts are also referred to as mono-spaced since each letter takes up the same horizontal space. Examples include Courier (Corpus).

Font

A Single typeface for example Trinity Italic. Many people say a font when they actually mean a Font Family. The font Trinity.Italic is actually one weight from the Trinity type face family.

Fontitis

An affliction from which inexperienced DTP operators suffer. Symptoms include many different typefaces in lots of sizes and weights on the same page, everything set in upper case, and lots of shadows, auto-expanded, auto-condensed and other generally distorted text.

Gothic

Font attribute meaning the font is sans serif. Usually used where there is a sans serif version of a serif face, Clear Gothic, for example.

Hinting

Extra information coded into all quality fonts which ensures the characters remain symmetrical and correctly shaped when displayed on screen or printed. Hinting also ensures that fine parts of letters do not disappear when their size is reduced so much that these parts become less than one pixel thick. This is very important for resolutions below 1200 dpi.



Italic

Font attribute. The italic version of a font is angled usually by between 5 and 20 degrees off the perpendicular. The characters are also redesigned to appear more cursive. Italic characters cannot be automatically made by any computer or programmer. See Oblique

Leading (Pronounced Ledding)

Extra space added between lines of text. Printers used to do this by placing strips of lead between lines of letters. This is sometimes also known as line spacing but leading is the correct term.

Left side bearing (LSB)

The offset of the left side of each character from the position (0,0). Some fonts have no left side bearing, script fonts often have a left side bearing defined in each character.

Ligatures

Two or more letters designed as a single entity, f and i for example are designed as a single character (fi) in which the dot over the i is often removed. Not to be confused with diphthongs which are actually letters in their own right.

Non-ranging or Oldstyle numerals

Numbers that have x-height, ascenders and descenders, these are sometime called lower case numerals.

Oblique

Font attribute. The Oblique version of a font is angled usually by 5 and 20 degrees off the perpendicular. Unlike an Italic font none of the characters are re-designed.

Outline font

The shapes of the letters in outline fonts are described by bezier curves and



so can be scaled to any size without any loss of shape or accuracy. Most modern computers use Outline fonts since they can be used at any size. Hinting ensures that at small sizes the font is not corrupted when displayed on screen.

Pica

12 points or 1/6th of an inch. This is sometimes called a pica em, and should not be confused with an em or em space.

Pixel

Picture Element. A single dot many of which are used to display a picture. The fonts displayed on any monitor are composed of pixels - even if they are outline fonts because a monitor can only display pixels

Point

1/72nd of an inch. If type is set at 72 point one would imagine that the text would be one inch high, this is not the case. See the entry Point Size for an explanation

Point size

The size of the letters used to set the text. This is complicated by the fact that in the days of metal type the point size referred to the "height" of the piece of metal the shape of the character is stuck out from. Today there is no such piece of metal, so to the casual observer point sizes of typefaces appear to be arbitrary, and differ considerably from face to face. The designer of the typeface decides how much space there should be between lines of text if no extra space, or leading, is added. This overall height of the actual letters plus the designers minimum space adds up to the point size.

Proportional spaced font

Font in which all the letters have different horizontal widths, for example an I will take up less horizontal space than an M.

Sans serif typeface

A typeface in which the letters have no serifs, Helvetica, for example.

Serif

Short terminating lines at the ends of letter strokes. Trinity is a serif font whereas Homerton is a sans-serif font (ie it has no serifs).

Serif typeface

A typeface in which the letters have serifs, Times, for example. The design of serif faces has changed over the years. Originally a serif face would have been closer to hand-writing in style. More modern fonts have an engraved type serif which is vertical. Modern serif faces first appeared in Venice in the late 15th century.



Shilling fraction

A fraction in which the numerator and denominator do not sit directly above and below one another e.g. $\frac{3}{4}$ or $\frac{1}{2}$. See stacked fraction

Small Caps

Capital letters designed to have the same or very slightly larger size than the font's x-height, but which have the same weight. FontFiend can generate small caps automatically. You may need to manually alter the weight of small caps generated by FontFiend using *Alter weight*.

Smart quotes

Single or double quotation marks which have correct opening (or left hand or 66) and closing (or right hand or 99) shapes. Many people incorrectly use the symbols for feet and inches (' and ") instead.

Stacked fraction

A fraction in which the numerator and denominator sit directly above and below one another. See shilling fraction.

Swash letter

A letter with elaborate flowing tails and curves. Swash caps are often used to add decoration.

Typeface or Typeface Family

A group of fonts with the same name: Times, Times Italic, Times Bold, etc.

Weight

The weight of a typeface is its overall appearance of light or black. A bold or heavy font appears darker because the character's strokes are thicker and less of the white page can show through.

x-height

The size of lower case letters without ascenders or descenders, most of the time this is actually the size of the lower case x.



Appendix 2 - Font Problem Solving

The font I have made produces a “Font cache full” error

This font has one or more faulty bounding boxes. Make sure that the Re-calculate bounding boxes option is on in the FontFiend preferences window. Now load the font then save it out again. This will repair any faulty bounding boxes.

The font is fine on screen but doesn't print correctly

The normal reason for this is use of a Computer Concepts Laser Direct or Turbdriver printer driver. Both of these printer drivers have problems rendering many fonts correctly. Turn the halftone text option in the Laser Direct/TurboDriver Halftone options window off. The only draw back with this is that all text will then be printed black, regardless of its actual colour on screen. This is not a fault with FontFiend. All professional font suppliers in the Acorn market have encountered this problem.

When I load a font into FontED I get errors like “Cyclic scaffold tree” and “Invalid linear link”

The font in question has a badly corrupted scaffold tree. What you need to do is load the font into FontFiend and then delete all scaffolds. When you save the font back out from FontFiend the scaffolds will be removed but FontED will then be able to load the font without errors. You may also decide to re-scaffold the font in FontFiend before saving it back out.

When I use the font some characters get filled in

The font has been saved with the wrong winding rule. Load the font into FontFiend and then go to the Misc.Font option in the main menu. Alter the status of the Non zero winding rule button. Now save the font back out again. This will usually fix the problem.

I have made a font using FontFiend, can I distribute it?

Yes, FontFiend is a commercial font editor. The program itself is copyright, however any font you make yourself from scratch using the package belongs to you. FontFiend will place a small message inside any font edited. This message identifies the version of FontFiend used to edit the font in case any user has a problem with the font. FontFiend will also place a date inside the font so that anyone can easily identify when the font was made.

Appendix 3 - Encoding tables



FontFiend generates all characters for a font based upon an encoding file. The standard encodings are:

Latin 1

Danish, Dutch, English, Faroese, French, German, Icelandic, Irish, Italian, Norwegian, Portuguese, Spanish, Swedish & Welsh.

Latin 2

Albanian, Croatian, Czech, English, German, Hungarian, Polish, Rumanian, Slovak, Slovene & Swedish.

Latin 3

Afrikaans, English, Esperanto, French, Gaelician, German, Italian, Maltese & Turkish.

Latin 4

Danish, Dutch, English, Finnish, French, German, Irish, Italian, Norwegian, Portuguese, Spanish, Swedish & Turkish.

Custom Encodings

FontFiend not only supports the encodings listed above (which are built into the application) but any latin custom encoding you drop into the encoding window. As you can see from the list above no single font includes the letters for both Esperanto & Spanish. You could make your own custom encoding which includes both and then generate a font with the required letters. See the section in the manual on importing encoding files.

FontFiend does not support any non latin encoding. This includes Cyrillic (Russian) and Symbolic fonts.



Latin 1 Encoding

%Latin 1	at	Ycircumflex	Acircumflex
iSV_Reserved	A	Wacute	Atilde
.notdef	B	oneproportional	Adieresis
.notdef	C	wcircumflex	Aring
.notdef	D	ycircumflex	AE
.notdef	E	wacute	Ccedilla
.notdef	F	ellipses	Egrave
.notdef	G	eth 1	Eacute
.notdef	H	foursuperior	Ecircumflex
.notdef	I	trademark	Edieresis
.notdef	J	dotlessi	Igrave
.notdef	K	fraction	Iacute
.notdef	L	florin	Icircumflex
.notdef	M	perthousand	Idieresis
dollarbar	N	bullet	Eth
slash_O	O	quoteleft	Ntilde
slash_o	P	quoteright	Ograve
ring	Q	guilsingleft	Oacute
circumflex	R	guilsingright	Ocircumflex
tilde	S	quotedblleft	Otilde
.notdef	T	quotedblright	Odieresis
dotaccent	U	quotedblbase	multiply
breve	V	endash	Oslash
macron	W	emdash	Ugrave
hungarumlaut	X	minus	Uacute
caron	Y	OE	Ucircumflex
.notdef	Z	oe	Udieresis
.notdef	bracketleft	dagger	Yacute
.notdef	backslash	daggerdbl	Thorn
.notdef	bracketright	fi	germandbls
.notdef	asciicircumflex	fl	agrave
.notdef	underscore	space	acute
.notdef	grave	exclamdown	acircumflex
space	a	cent	atilde
exclam	b	sterling	adieresis
quotedbl	c	currency	aring
numbersign	d	yen	ae
dollar	e	brokenbar	ccedilla
percent	f	section	egrave
ampersand	g	dieresis	eacute
quotesingle	h	copyright	ecircumflex
parenleft	i	ordfeminine	edieresis
parenright	j	guillemotleft	igrave
asterisk	k	logicalnot	iacute
plus	l	hyphen2	icircumflex
comma	m	registered	idieresis
hyphen	n	asciimacron	eth
period	o	degree	ntilde
slash	p	plusminus	ograve
zero	q	twosuperior	oacute
one	r	threesuperior	ocircumflex
two	s	acute	otilde
three	t	mu	odieresis
four	u	paragraph	divide
five	v	periodcentered	oslash
six	w	cedilla	ugrave
seven	x	onesuperior	uacute
eight	y	ordmasculine	ucircumflex
nine	z	guillemotright	udieresis
colon	braceleft	onequarter	yacute
semicolon	bar	onehalf	thorn
less	braceright	threequarters	ydieresis
equal	asciitilde	questiondown	
greater	logo	Agrave	
question	Wcircumflex	Aacute	

Latin2 Encoding



%Latin2	at	cent	Acircumflex
iSV_Reserved	A	paragraph	Abreve
.notdef	B	oneproportional	Adieresis
.notdef	C	.notdef	Lacute
.notdef	D	copyright	Cacute
.notdef	E	registered	Ccedilla
.notdef	F	ellipses	Ccaron
.notdef	G	.notdef	Eacute
.notdef	H	.notdef	Eogonek
.notdef	I	trademark	Edieresis
.notdef	J	dotlessi	Ecaron
slashaccent	K	.notdef	lacute
fraction	L	florin	Icircumflex
dotlessi	M	perthousand	Dcaron
dollarbar	N	bullet	Eth
slash_O	O	quotelleft	Nacute
slash_o	P	quoteright	Ncaron
ring	Q	guilsinglleft	Oacute
circumflex	R	guilsinglright	Ocircumflex
tilde	S	quotedblleft	Ohungarumlaut
.notdef	T	quotedblright	Odieresis
dotaccent	U	quotedblbase	multiply
breve	V	endash	Rcaron
macron	W	emdash	Uring
hungarumlaut	X	minus	Uacute
caron	Y	guillemotleft	Uhungarumlaut
.notdef	Z	guillemotright	Udieresis
.notdef	bracketleft	dagger	Yacute
.notdef	backslash	daggerdbl	Tcedilla
.notdef	bracketright	fi	germandbls
.notdef	asciicircumflex	fl	racute
.notdef	underscore	space	aacute
.notdef	grave	Aogonek	acircumflex
space	a	breve	abreve
exclam	b	Lslashaccent	adieresis
quotedbl	c	currency	lacute
numersign	d	Lcaron	cacute
dollar	e	Sacute	ccedilla
percent	f	section	ccaron
ampersand	g	dieresis	eacute
quotesingle	h	Scaron	eogonek
parenleft	i	Scedilla	edieresis
parenright	j	Tcaron	ecaron
asterisk	k	Zacute	iacute
plus	l	hyphen	icircumflex
comma	m	Zcaron	dcaron
hyphen	n	Zdotaccent	eth
period	o	degree	nacute
slash	p	aogonek	ncaron
zero	q	ogonek	oacute
one	r	Lslashaccent	ocircumflex
two	s	acute	ohungarumlaut
three	t	lcaron	odieresis
four	u	sacute	divide
five	v	caron	rcaron
six	w	cedilla	uring
seven	x	scaron	uacute
eight	y	scedilla	uhungarumlaut
nine	z	tcaron	udieresis
colon	braceleft	zacute	yacute
semicolon	bar	hungarumlaut	tcedilla
less	braceright	zcaron	dotaccent
equal	asciitilde	zdotaccent	
greater	logo	Racute	
question	sterling	Aacute	



Latin 3 Encoding

%Latin3	at	cent	Acircumflex
iSV_Reserved	A	paragraph	Atilde
.notdef	B	oneproportional	Adieresis
.notdef	C	.notdef	Cdotaccent
.notdef	D	copyright	Ccircumflex
.notdef	E	registered	Ccedilla
.notdef	F	ellipses	Egrave
.notdef	G	.notdef	Eacute
.notdef	H	foursuperior	Ecircumflex
slash_H	I	trademark	Edieresis
slash_h	J	dotlessj	Igrave
foursuperior	K	.notdef	Iacute
fraction	L	florin	Icircumflex
slash_h	M	perthousand	Idieresis
dollarbar	N	bullet	Eth
slash_O	O	quoteleft	Ntilde
slash_o	P	quoteright	Ograve
ring	Q	guilsinglleft	Oacute
circumflex	R	guilsinglright	Ocircumflex
tilde	S	quotedblleft	Gdotaccent
.notdef	T	quotedblright	Odieresis
dotaccent	U	quotedblbase	multiply
breve	V	endash	Gcircumflex
macron	W	emdash	Ugrave
hungarumlaut	X	minus	Uacute
caron	Y	OE	Ucircumflex
.notdef	Z	oe	Udieresis
.notdef	bracketleft	dagger	Ubreve
.notdef	backslash	daggerdbl	Scircumflex
.notdef	bracketright	fi	germandbls
.notdef	asciicircumflex	fl	agrave
.notdef	underscore	space	aacute
.notdef	grave	Hslash	acircumflex
space	a	breve	atilde
exclam	b	sterling	adieresis
quotedbl	c	currency	cdotaccent
numbersign	d	yen	ccircumflex
dollar	e	Hcircumflex	ccedilla
percent	f	section	egrave
ampersand	g	dieresis	eacute
quotesingle	h	ldotaccent	ecircumflex
parenleft	i	Scedilla	edieresis
parenright	j	Gbreve	igrave
asterisk	k	Jcircumflex	iacute
plus	l	hyphen	icircumflex
comma	m	onesuperior	idieresis
hyphen	n	Zdotaccent	eth
period	o	degree	ntilde
slash	p	hslash	ograve
zero	q	twosuperior	oacute
one	r	threesuperior	ocircumflex
two	s	acute	gdotaccent
three	t	mu	odieresis
four	u	hcircumflex	divide
five	v	periodcentered	gcircumflex
six	w	cedilla	ugrave
seven	x	dotlessi	uacute
eight	y	scedilla	ucircumflex
nine	z	gbreve	udieresis
colon	braceleft	jcircumflex	ubreve
semicolon	bar	onehalf	scircumflex
less	braceright	threequarters	dotaccent
equal	asciitilde	zdotaccent	
greater	logo	Agrave	
question	sterling	Aacute	

Latin 4 Encoding



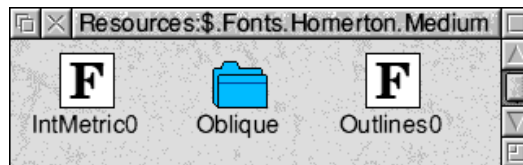
%Latin4	at	cent	Acircumflex
iSV_Reserved	A	paragraph	Atilde
.notdef	B	oneproportional	Adieresis
.notdef	C	.notdef	Aring
.notdef	D	copyright	AE
.notdef	E	registered	logonek
.notdef	F	ellipsis	Ccaron
.notdef	G	.notdef	Eacute
.notdef	H	.notdef	Eogonek
.notdef	I	trademark	Edieresis
.notdef	J	dotlessi	Edotaccent
slash_T	K	.notdef	Iacute
slash_t	L	florin	Icircumflex
dotlessi	M	perthousand	Imacron
dollarbar	N	bullet	Eth
slash_O	O	quoteleft	Ncedilla
slash_o	P	quoteright	Omacron
ring	Q	guilsinglleft	Kcedilla
circumflex	R	guilsinglright	Ocircumflex
tilde	S	quotedblleft	Otilde
.notdef	T	quotedblright	Odieresis
dotaccent	U	quotedblbase	multiply
breve	V	endash	Oslash
macron	W	emdash	Uogonek
hungarumlaut	X	minus	Uacute
caron	Y	OE	Ucircumflex
.notdef	Z	oe	Udieresis
.notdef	bracketleft	dagger	Utilde
.notdef	backslash	daggerdbl	Umacron
.notdef	bracketright	fi	germandbls
.notdef	asciicircumflex	fl	amacron
.notdef	underscore	space	aacute
.notdef	grave	Aogonek	acircumflex
space	a	kra	atilde
exclam	b	Rcedilla	adieresis
quotedbl	c	currency	aring
numeralsign	d	ltilde	ae
dollar	e	Lcedilla	iogonek
percent	f	section	ccaron
ampersand	g	dieresis	eacute
quotesingle	h	Scaron	eogonek
parenleft	i	Emacron	edieresis
parenright	j	Gcedilla	edotaccent
asterisk	k	Tslash	iacute
plus	l	hyphen	icircumflex
comma	m	Zcaron	imacron
hyphen	n	macron	eth
period	o	degree	ncedilla
slash	p	aogonek	omacron
zero	q	ogonek	kcedilla
one	r	rcedilla	ocircumflex
two	s	acute	otilde
three	t	itilde	odieresis
four	u	lcedilla	divide
five	v	caron	oslash
six	w	cedilla	uogonek
seven	x	scaron	uacute
eight	y	emacron	ucircumflex
nine	z	gcedilla	udieresis
colon	braceleft	tslash	utilde
semicolon	bar	.notdef	umacron
less	braceright	zcaron	dotaccent
equal	asciitilde	.notdef	
greater	logo	Amacron	
question	sterling	Aacute	



Homerton Medium is a base 0 font, it has more than 255 characters and a 0 at the end of each file name.

Appendix 4 - Loading Base0 fonts

The Acorn font manager supports a font type known as Base 0. These fonts can easily be identified because they have a 0 at the end of their file name.



Base 0 fonts contain more than the normal 256 characters. FontFiend can load these fonts and convert them to the current encoding. For example you could load Homerton.Medium and convert it to a Latin 2 encoding. Base 0 fonts should contain the full character sets from latin 1,2,3 and 4. i.e. all letters, symbols and accents required to display most latin based languages. Some font suppliers supply fonts in Base 0 format with only the latin 1 characters defined. This means that the font has a lot of blank characters.

The original idea behind Base 0 was that the font manager would be able to select the correct encoding file for the given territory and then display any Base 0 font with the correct character set. However for some reason although this feature is built into Risc OS it doesn't actually work correctly.

When you load a Base 0 font FontFiend will first confirm that you want to convert the font to the current encoding. If the current encoding is the wrong one you can abort loading of the font, change the encoding, and try again.

Some Base 0 fonts, such as Homerton, contain a complex scaffold in which some characters from the tree may not occur in the encoding you are converting to. FontFiend will convert as much of the tree as possible, then remove any defective scaffolds. This ensures that the font will work in the new encoding. However you may find a few characters that have some missing scaffolds.

The font will then load and can now be treated the same as any normal font with 256 characters.

Appendix 5 - How do I....?



This section covers some common questions asked by users.

Make a “proper” extended font?

As we have seen earlier FontFiend can make an extended version of a font using the *Transform* dialogue box. However this will result in the vertical stroke widths of character being bigger than the horizontal stroke width. This can be corrected using the *Outlines.Fx/Process/SpecialFX* dialogue box. Select *Alter weight* as the effect. Choose a small negative *X thickness* (perhaps -20) and a *Y thickness* of 0. Now click on the OK button. FontFiend will now correct the widths of horizontal strokes. The amount of *X thickness* required to correct the font may need to be discovered using trial and error, however between -10 and -20 will normally be correct.

Make a “proper” condensed font?

This can be done in the same was an extended font. However having condensed the font you will need to alter the *X thickness* of the font by a positive amount. When correcting a font after condensing it a value of 15 is a good starting point.

Correct the stroke width of small caps?

When FontFiend makes small caps they are just scaled down versions of upper case letters. The visual weight of these letters may be wrong. The weight of them can be altered using the *Outlines.Fx/Process/SpecialFX* dialogue box available from the Outline character menu. You will need to adjust both the *X thickness* and *Y thickness* by a similar amount. You will need to alter each small capital letter individually, however you can keep the *SpecialFX* dialogue box on screen by clicking ADJUST on the OK button after altering the weight of each character.

Prevent white sections appearing in “made open” letters?

Make Open can have problems with fonts that have multiple close control points. If you find small white sections appearing in characters that have been made open then the following will help. Re load the normal version of the font. Go to the *Outlines.FX/Process.Clean Points* dialogue box. Select a *Clean within* value of around 25 and just select the *Remove double points* option. Now click on OK. Double points will be removed from the font and you can now use *Make Open*.



Convert a font from Latin1 to another encoding?

You can use FontFiend to produce Latin 2,3 or 4 versions of existing fonts.

If the font is a Base 0 font then set the FontFiend encoding to the one you want (2,3 or 4). Then load the font. If any characters are missing you can then use the Character Builder to add them. Turn on the *Overwrite existing accented letters* button and turn off the *Overwrite existing characters* button. Make sure all the *Char Build* buttons are on and click on *OK*.

If the font is a normal font load it into FontED using any encoding. Now change to the correct encoding once the font is loaded. Now go to the Character Builder. Turn on all the *Char Build* buttons. Turn on both *Overwrite existing accented letters* and *Overwrite existing letters* now click on the *OK* button.

You will now have a font of the correct encoding.

Here are just a few of the other products you will be able to
download from RISCOS.com free of charge
(You can make a donation to help support future product releases but this is entirely voluntary)



This Documentation is © 3QD Developments Ltd 2026
Some of the products shown above might not be available immediately.
Not all products will run on all versions of RISC OS, check the product information page for details.

Other Products list last updated 29/2/26